Mapping Implemented Strategies of Bringing Internet-based Software Applications to Market Using Storytelling Research Methods

by

Hugh M Pattinson

A thesis submitted in partial fulfilment of the
requirements for the degree of

Doctor of Philosophy (PhD)

University of Technology, Sydney

2005

CERTIFICATE OF AUTHORSHIP / ORIGINALITY

I, Hugh Matthew Pattinson, certify that this thesis has not previously been submitted for a degree nor has it been submitted as part of the requirements for a degree.

I also certify that the thesis has been written by me. Any help that I have received in my research work and in the preparation of the thesis has been recognised and acknowledged. **In addition, I certify that all information sources and literature used** are indicated in the thesis.

Acknowledgements

There are a number of special people who have provided much support, inspiration and wise counsel throughout this research project.

In the early days of the project, Linden Brown and Nigel Barrett provided strong support and encouragement to start and refine my dissertation topic. Professor Ken Miller has been there from the start of the project and has provided constant encouragement and support throughout the journey. I am particularly indebted to Professor Miller for finding me an outstanding external supervisor and for believing in me and supporting me that I would finally submit the dissertation.

I would like to especially thank Professor Arch Woodside for working with me to finally sort out my topic, and for being there right through to the completion of the project. Professor Woodside provided outstanding turnaround on my work and wise counsel, in all stages of running the project, and particularly with editing the dissertation in the final stages of the project.

I would also like to thank very much the participants in the project who provided me with their time and insights in decision-making, for without them this project could have been completed. They were all generous with their time and following up on my rounds of discussions with them.

There are of course those special people who have been there throughout the journey. I would like to thank Suresh Sood who has been a great encouragement, particularly in challenging the intellectual content and issues in my topic, thus helping me not just to refine the topic but also to greatly improve effective knowledge and practical contribution from the Dissertation.

I want to particularly thank my family who have been living through this journey. To my daughter, Lizzy, who has seen this project from start to finish, and to my son, Mark, for whom this project has taken literally a "lifetime"!

Finally, I want to thank Sue who has been there right through the project, from the early days of many ideas, through to the many days of field work and analysis, and to those last intensive days of completion. The dissertation is really completed and submitted and it is now time to move on to be a real academic!

TABLE OF CONTENTS

Figures, Tables & Exhibits

Abstract

The research offers an advanced hermeneutic framework for the study of B2B decision-making processes for software application development. The proposal builds dynamic sensemaking elements onto existing hermeneutic analysis frameworks used in marketing research (Arnold & Fisher 1994; Thompson Pollio & Locander 1994; Thompson 1997).

The “hermeneutic circle” is extended to a multilevel “hermeneutic system” starting with written accounts of the decision-making associated with the development and delivery of new software applications – these accounts are then revisited (Langley et al. 1995) through multiple rounds of etic and emic interpretation.

An extended form of marketing Decision Systems Analysis (DSA) that complements the advanced hermeneutic analysis framework is also proposed and applied, building on existing B2B Marketing DSA (Hulbert, Farley & Howard 1972; Woodside & Samuel 1981; Woodside 1994; Woodside & Voss 1999) with extended DSA Models, events chronology maps, and cognitive maps. The proposed framework and supporting analysis are applied to six application software houses that created new Internet-based software applications which were either disruptive (Christensen 2003) or supporting disruptive software applications.

Findings from six case studies suggest that the hermeneutic research framework proposals provide a strong systematic platform for analysing and interpreting decision-making with deep prospective, introspective, retrospective, and with imaginatively unbounded current and future perspectives. Individual case-study and cross-case findings were incorporated into a set of new and revised theory propositions for software application developer Structuration at individual, firm (including multi-firm) and industry levels. The propositions provide a base of knowledge for the further development of a cognitive view of a software application development firm.

Chapter 1: Introduction

This research project addresses a number of questions:

- How can we craft accurate, rich and insightful stories on marketing decision-making?
- How can we unearth those nuggets of sensemaking gold that help us to learn more about how marketing decisions occur – or marketing decisions yet to be executed?
- How can we understand marketing decision-making applied to the development of new forms of codified knowledge such as application software, which are the building blocks of a new marketing information environment, and of new marketplaces (or "marketspaces")?
- Are there new issues and challenges that confront decision-makers as they develop applications to extend and enhance the Internet and its multimedia service, the World-Wide Web, or as products to be distributed through this new environment?
- Is this new environment different for business-to-business (B2B) Marketing decision-making from existing B2B manufacturing and high-technology contexts?

The project at its outset was focused on exploring, analysing and "making sense" of stories or accounts of application software houses developing new applications for, and with, the Internet. The rationale for the strategy was that detailed stories augmented by review and revisitation would contribute deeper insights into decision-making processes, more than other research methodologies used in the field of marketing.

Early in the project, there was a realisation that there were few established storytelling methodologies being used for research into marketing decision-making. Furthermore, there was room to develop more advanced, dynamic and interpretive storytelling methodologies that could be better suited to marketing contexts.

There was a base of knowledge in the area of mapping marketing processes, built on flowcharting techniques developed for conceptualising computer and information systems projects, in the 1960s and 1970s. This flowcharting approach, known as decision systems analysis (DSA), progressed slowly through the 1980s and 1990s with incremental improvements in charting (mapping) conventions and symbols, but with increasing application to B2B marketing research challenges.

The project was tuned toward understanding decision-making from a "sensemaking" perspective which included schematic mapping and, hopefully, effective interpretation and discussion of decision-making processes.

As the research project progressed, it became clear that the development of both a framework for dynamic analysis and interpretation, and an appropriate decision-making mapping system (that is, creation of cognitive maps addressing decision-making), were contributions equal in stature to the actual analysis of a new and exciting application software context.

Indeed, by the end of the project while there were several nuggets contributing knowledge scattered throughout the case studies in the project, three contributions stand out:

1. Insights into decision-making processes for the development and delivery of disruptive software applications (or applications contributing to other disruptive technologies, a new context for B2B marketing decision systems analysis (DSA))
2. The development of a new storytelling methodology based on a hermeneutic framework that is well suited to the exploration of B2B marketing accounts
3. The development of a new cognitive mapping system to enhance sensemaking of marketing decisions in a B2B context.

This report is the account of a journey with many twists and turns – a journey where discovering new roads, and how to build them, was just as important as reaching the destination.

1.1 Overview: Sensemaking, Hermeneutic Analysis, Decision Systems Analysis and Its Application to a New Business-to-business Context

Weick (1995) emphasises that all "sensemaking" is retrospective, that is, an individual or organisation crafts an understanding of their lived experiences by reflection. The following question expresses Weick's view of sensemaking, "How do I know what I think until I hear what I say and see what I've done?"

Because sensemaking is the basis for an individual's mental models of reality and the causes of reality, research in B2B decision-making that provides thick descriptions of participants' views of what happened and why (that is, emic sensemaking) complements and extends decision-making researchers' views of decisions – actions – outcomes (that is, etic sensemaking).

From a post-decision and post-action perspective, emic sensemaking reports often indicate more structure and rationality, while etic sensemaking of the same events more often describes anarchical (for example, Cohen, March, & Olsen 1972; Hickson et al. 1986) and iterative sequences (Mintzberg, Raisinghani, & Théorêt 1976) with feedback loops and delays occurring often in organisational decision-making.

Langley et al. (1995) advocate conceptually and empirically "opening up decision-making" to: (1) the ambiguities that surround the relationship between commitment and action; (2) the critical role of insight in transcending the bounds of cerebral rationality—the need to examine an organisation's history, experience, affect, and inspiration (for example, will and vision); (3) dynamic linkages so that isolated traces of single decisions come to be seen as interwoven networks of issues; and (4) using multiple research perspectives and tools—such as zooming in closer to people and processes under study and zooming out to exploring the ramifications of issue networks and the histories of organisations over long time periods (for example, Pettigrew 1985). Regarding their fourth suggestion, Langley et al. (1995) advocate focusing on people and personalities, not just events, and on re-analysing previously analysed decision processes, not just new ones. Compared to Weick's (1995) view, these researchers advocate embracing a more catholic view for research on decision-making:

> As researchers, we may all be acutely aware of the boundedness of cerebral rationality. But that does not justify us in promoting methods that deny the existence of ambiguity, insight, interaction. Decision-making is prospective, introspective, and retrospective, sporadically rational, ultimately affective, and altogether imaginatively unbounded (Langley et al. 1995, p. 277).

The research project advocates and advances the Langley et al. (1995) view of decision-making and research on decision-making.

An advanced hermeneutic framework for the study of B2B decision-making processes was developed for the project that incorporates collecting multiple rounds of emic- and etic- based data of the mental models, decisions, events, and outcomes concerning a specific enterprise issue.

The new framework recognises and greatly extends the research on B2B decision-making from Arnold & Fischer (1994); Thompson, Pollio, & Locander (1994); and Thompson (1997). The principal objective in the project is to describe the conceptual and research tools for achieving deeper sensemaking of what happened and why it happened—including how participants interpret outcomes of what happened and the dynamics of emic and etic sensemaking.

Research on dynamic sensemaking is the study of how emic and/or etic understanding of reality (that is, what happened and why) changes over time. The 21st Century book (*Argument without End: In Search of Answers to the Vietnam Tragedy*) and film (*The Fog of War*) of Robert McNamara's 1993 commentaries of his sensemaking and actions occurring during the U.S. – Vietnam War in the 1960s while he was the U.S. Secretary of Defense is an example of dynamic sensemaking data.

Dynamic sensemaking research includes data collection on more recent sensemaking of both earlier decisions – actions as well as earlier sensemaking. As such, dynamic sensemaking relates to and advances from hermeneutical research—achieving an understanding of a specific situation–problem–decision–action–outcome by reflective analysis of autonomous text including resolving contradictions among and between individual elements (micro-events) and the entire process under study – that is, applying a hermeneutical circle.

Examples of autonomous text include news clips of McNamara's speeches during the 1960s and Harvard Business School cases describing the histories of enterprises along with specific problems–actions–outcomes for these firms. See Arnold and Fischer (1994) and Thompson (1997) for thorough discussions on hermeneutics in consumer research and for deriving marketing insights.

The hermeneutical framework developed for this research project incorporates the use of decision systems analysis (DSA, see Hulbert, Farley, Howard 1972; Woodside 2003), cognitive mapping (see Huff and Huff 2000), computer software-based text analysis (for example, NVivo and TACT (Textual Analysis Computing Tools) 1997), and the long interview method (McCracken 1988) for mapping the mental models of the participants in specific decision-making processes as well as mapping the immediate, feedback and downstream influences of decisions–actions–outcomes.

DSA includes conducting several rounds of interviews with multiple informants to construct accurate flowcharts of information and decision systems within organisations. Several rounds of interviews are necessary to integrate the views of the multiple participants involved in the decision-making process under study and to validate the work of the researchers; thus, the informants examine drafts of the flowcharts and text descriptions of the researchers over several interviews until both the informants and researchers are satisfied as to the accuracy and completeness of the model of the information and decision system (see Capon & Hulbert 1975; Howard, Hulbert, & Farley 1975; Hulbert 2003).

TACT (1997) and NVivo (QSR International 2002) are computer software programs that enable text coding and mapping for displaying complex associations of concepts. Such displays are created to examine both the details and gestalt of the mental models as expressed in the text.

Cognitive mapping includes creating models of the thinking processes of individuals and/or organisations (see Huff 1990). Huff and Huff (2000) present an in-depth case study of how cognitive mapping can provide foundation data for building system dynamic models and running simulations of how organisations respond to environmental changes and engage successfully and unsuccessfully with new product development.

The long interview method includes selecting informants based on their known involvement and participation in the issue under study, using probing open-ended questioning methods, and displaying documents and other exhibits (for example, possibly products) during the interview as topics of discussion; long interviews range from two to four hours.

The long interview aims to acquire a deep understanding and description of the lived experiences, beliefs, attitudes, and intentions in informants' own words (for example, an emic perspective of what happened and why it happened).

While applying the framework is useful in itself for achieving a deep understanding and description of the issue under study; the proposed application of hermeneutic interpretation represents valuable data collection methods for building dynamic models of complex systems (cf. Maani & Maharaj 2004; Sterman 2000, p. 80).

A basic principle in the systems dynamics literature is that feedback loops, delays, hidden demons (that is, positive relationships in feedback loops creating vicious versus virtuous circles of system failure), and unintended (often downstream) consequences go unnoticed by most participants involved in a strategic business unit (for example, see Hall 1976; Maani & Maharaj 2004). Hermeneutical interpretations provide useful insights and tools for explicating these implicit loops, delays, demons, and consequences.

By way of introduction to the whole research project, an overview of the hermeneutical framework for research on understanding decisions, actions, and storytelling by B2B executives regarding a specific issue, or a network of issues, as well as the history of an enterprise, is outlined.

Six case studies were undertaken in the research project and five are fully reported in Chapters 5 to 9 of this document. One case study is reported in the Introduction to the extent of a substantial description of the research application of the hermeneutic framework to the decisions–actions–sensemaking by executives of a computer software start-up enterprise – Red Hat, Inc. Reporting of the complete Red Hat case study extends out of the Introduction chapter into Chapter 4.

1.2 A Hermeneutical Framework for Understanding and Research of Decisions – Actions – Storytelling by B2B Executives

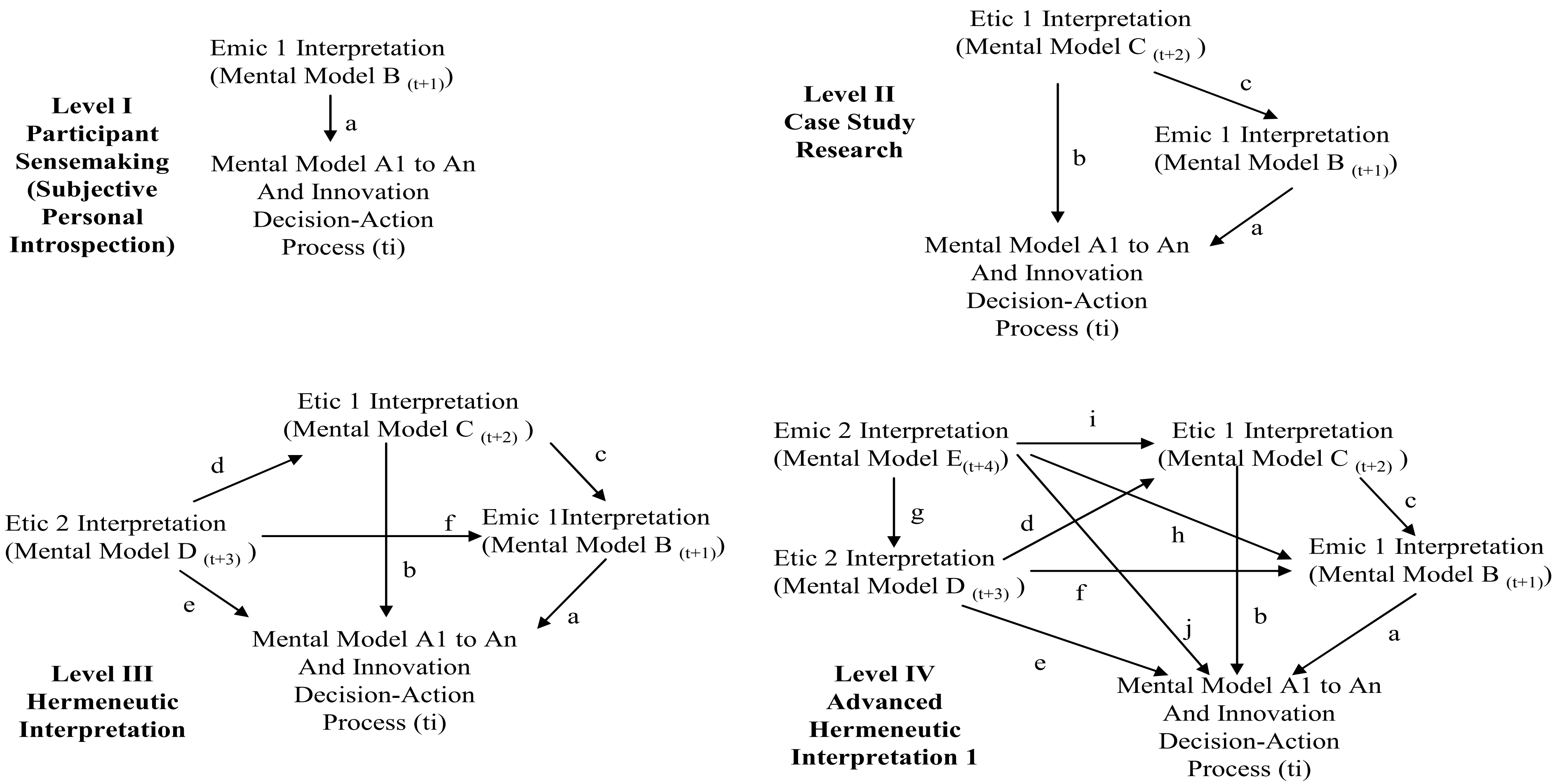
Exhibit 1.1 presents four progressive levels of understanding and research on B2B decision-making. **The first level of analysis, Level I,** is a representation of B2B executives' descriptions and explanations of what happened and why it happened for a focal decision-making issue. For example, a quote or summary description of what an executive reports happened when she met with the president of a customer firm is an example of Level I data.

Note that in Exhibit 1.1 that Level I shows that mental models are crafted and revised during the decision and action under study – at time t. The executive's later (t + 1) interpretation and reporting of what happened represents both a summary and an elaboration of the mental models originating during the decisions–actions. What the executive reports happened to herself and colleagues is subject to self-editing, memory failure, and personal prejudices and biases (Woodside 1993).

Research on the stories (that is, the narrative structure and contents of a participant's sensemaking about what happened and why) support the conclusion that each participant quickly crafts a summary holistic story of what happened, why it happened, and the consequences of the process (see Gergen and Gergen 1986).

If asked, each participant in the process under study is capable of retrieving and discussing some details of people, places, and events. Most case-study research reports are based, in part, on participants' holistic stories and retrieved details of the process and outcomes. Storytelling reflects both plot and sensemaking by participants and case-study researchers. The reported stories are data files for hermeneutical interpretation.

Exhibit 1.1 Case Study Research and Hermeneutic Interpretation of Sensemaking in B2B Innovation Decisions-Action Processes



The current (early 21st Century) dominant logic in B2B research stops at collecting Level I data. Examples include internet or mail survey reports on executives' beliefs, attitudes, and descriptions of decisions concerning a specific issue without the researcher acquiring independent confirmatory data. These survey data are usually analysed using empirical positivistic (for example, analysis of variance and structural equation modelling) in efficacy tests of relationships among proposed antecedent, mediating, and outcome variables. Arrow **a** in Exhibit 1.1 represents a summary of what the participants in the enterprise report about the decision process under study.

The second level of analysis, Level II, recognises that a participant's t+1 interpretation of what happened a week, month, or year ago, and why it happened, is just one view of specific situations, decisions, and outcomes. This participant's emic view is not taken to reflect a complete or a completely accurate account of reality. The researcher provides commentary and often judgments (arrow **c**) on the participant's sensemaking account. The researcher collects (arrow **b**) additional interviews with other participants and/or analyses documents to confirm, deny, and elaborate on the participant's report. Most B2B case-study research reflects the use of Level II research (see Woodside and Wilson 2003).

The third level of analysis, Level III, steps into the realms of real "hermeneutic interpretation", where rigorous analysis of the existing accounts (usually written Level II accounts) is undertaken, and recast into revised outputs.

Level III research reflects Langley et al.'s (1995, p. 277) "Suggestion 5, re-analyse previously analysed decision processes not just new ones." Thus, Level III provides two etic interpretations and involved an additional time period and most likely independent researchers. Note in Exhibit 1.1 that etic 2 interpretations include commentaries of etic 1, emic 1, and mental models and decision processes at the time of the original situation—relationships **d**, **f**, and **e**, respectively.

Level III analysis may include chronologically mapping events of the decision process and outcomes reported by the etic 1 researcher, that is, the etic 2 researcher may conduct decision systems analysis (DSA, see Howard et al. 1975) based on the text of the original case study done by the etic 1 researcher.

Also, Level III analysis may include content analysis assisted by software tools (for example, TACT (1997) and NVivo (QSR International 2002). The research steps in the Red Hat case study reported on below include DSA, TACT and NVivo.

Level IV analysis includes a new round of interviewing one or more of the participants involved in the case study reported by the etic 1 researcher. Level IV research includes asking these participants a series of questions concerning several issues:

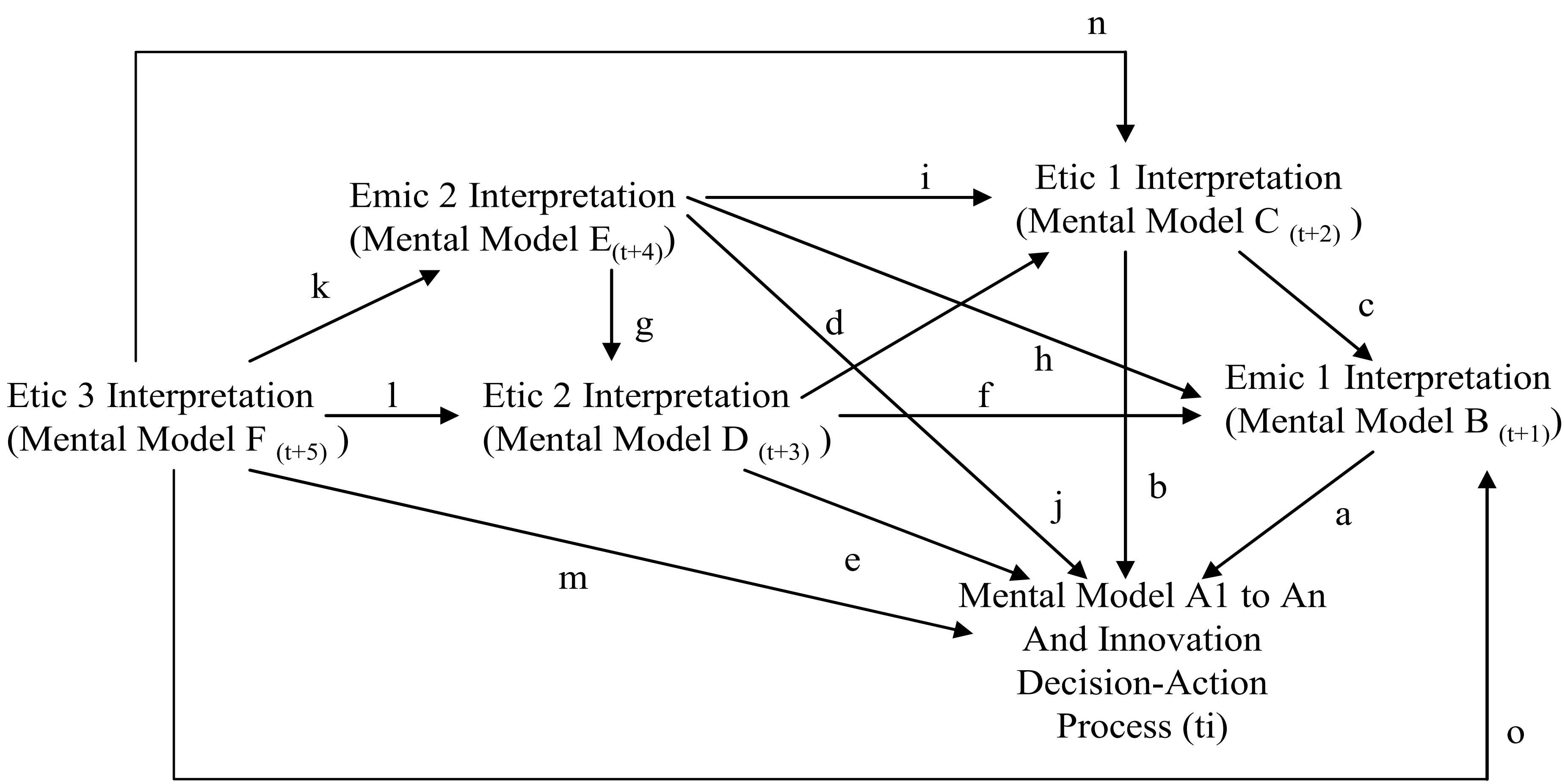
- The accuracy of the whole text and the key individual elements in the text in the etic 1 case-study report
- Possible revisions in the sensemaking expressions by participants that the etic 1 case-study reports
- The accuracy and possible improvements in etic 2 interpretations (arrow **g** in Exhibit 1.1)
- Updating of decisions and events subsequent to the process reported in the original etic 1 case study.

The fourth level of analysis, Level IV, is a further iteration or "advanced hermeneutic interpretation". The analysis includes gaining an emic interpretation of (1) the decision process reported in the original case study, (2) the etic 1 case-study report, and (3) the etic 2 draft sensemaking views of the whole case and details of the process (for example, via DSA, NVivo, and TACT).

Collecting emic data of interpretation of prior etic interpretations is found in several studies in the marketing and contemporary anthropology literatures (for example, see Howard and Morgenroth 1968; Whyte 1943). Howard and Morgenroth (1968) demonstrate collecting data from several rounds of interviews with participants after making revisions to their DSA models following each of these interviews and showing the revisions to the participants for correcting and further elaborations.

The fifth level of analysis, Level V, adds an etic 3 interpretation that focuses on revising sensemaking views of all prior interpretations, based especially on the evaluations and additional updated data from the emic 2 data (see Exhibit 1.2). If ambiguities, paradoxes, and conflicting views still exist and may be potentially resolvable, additional rounds of emic and etic analysis may be conducted. Woodside and Samuel (1981) provide an example of a participant becoming a co-author to a case-study report, following the participant's and researcher's revisions of several rounds of DSA models.

Exhibit 1.2 Level V Advanced Hermeneutic Interpretation 2



The framework proposed for the project was based around one independent researcher conducting distinct multiple rounds of interviews with the decision-makers. The framework could be improvised for more than one independent researcher, and transitions between levels of analysis could be modified. Indeed, in a completely continuous interaction approach, distinct emic and etic stages could become blurred. However, for the purposes of this project, transitions between levels of analysis are delineated through recognised changes between emic and etic information blocks – but those delineations are defined by the researcher. Ultimately the number of levels undertaken in the hermeneutic framework, and, until such time as distinct levels are defined, the transitions between levels, are left to the judgement of the researcher.

1.3 A B2B Application of Advanced Hermeneutic Research

This section offers an application of the proposals that the previous section describes for advancing hermeneutic research in B2B decision-making. The application included all the steps that are in Exhibits 1.1 and 1.2. Thus, the case combines secondary and primary data collection stages that include DSA and chronological mapping of events, cognitive mapping, TACT and NVivo content analyses, and personal interviews with executives participating in the decision process at the time covered by the original case-study report.

The specific case covers the start-up, initial public offering (IPO) of stock, and maturation of a computer software firm, Red Hat, Inc. MacCormack and Herman (2000) serve as the etic 1 report of the case study. The Red Hat case study explores the creation of the Linux operating system as an extension of the "open-source" philosophy (that is, available for sharing and revising among the community of users is encouraged) and organisation, and then packaging by Marc Ewing into Red Hat's version of Linux.

The Red Hat case study provides a conceptualisation of some software houses as "knowledge fabricators." The analogy is to large manufacturers such as General Motors or Boeing or IBM, that is, companies fabricating products out of thousands of parts, where substantial numbers of those parts are acquired from third parties.

The MacCormack and Herman (2000) study chronicles Red Hat founders Marc Ewing's and Rob Young's interest and motivations in Linux from 1993 through to the formation of Red Hat in 1995, and on to the public share float of the company in 1999. Additional primary research extends Red Hat's development philosophies and decision-making process during 2001-02.

Exhibit 1.3 Level I Analysis of Red Hat Case

Emic 1 view:
• "The one unique thing we could do that no else could do was, for the first time, we gave control of the operating system to the user."
• Successful IPO of a company marketing software that supported free products (i.e. the Linux open-source operating system)
• "Not bad for a company that gives its products away for free"
• The key issue: "Would an open-source development approach work for developing applications that could run on Red Hat's Linux operating system?

↓

1993-99 Initial mental model:
•Bundle a set of applications to enable Linux to be easily installed and maintained (1st version released in 1993)
Decisions-actions:
• Start-up of support tools for managing Linux open-source operating; market launch without beta test; RDC of 6 new releases; absorbed into Red Hat in 1995 on 3rd release (became Red Hat Linux Version 1).
• Direction and development of open-source graphic-user interface (GUI – GNOME project 1999); IPO August 1999.

Exhibit 1.3 summarises event milestones and the emic 1 sensemaking views identified in the data in the MacCormack and Herman (2000) case study; subsequently to developing Exhibit 1.3, emic 2 and etic 2 interpretations along with DSA and event maps provide revisions to Exhibit 1.3.

MacCormack & Herman's (2000) coverage of "open-sourcing," and Linux and Red Hat's application developments provide adequate detail for the development of an initial (etic 2) representative DSA model and to undertake further analysis on application development and delivery.

The etic 2 DSA model and maps for this case were updated following extensive questioning of the accuracy and completeness of the original MacCormack and Herman (2000) case. Thus, additional (emic 2) data were collected for etic 3 description and interpretation of the Red Hat decision-making process as reported in the original case study and for a period of three years beyond that reported by MacCormack and Herman. Emic 2 data consists of responses from interviews with the 2001 CEO and one of the founders of Red Hat (see Young 2001).

Exhibit 1.4 presents the etic 1 view of original case researchers regarding the emic 1 views and the major decision and event milestones in the Red Hat case. Thus, Exhibit 4 offers a succinct summary of the key details of the original case report (that is, MacCormack and Herman 2000). The initial DSA, event, and cognitive maps reported below offer details supporting Exhibit 1.4.

Exhibit 1.5 is a brief summary of etic 2 interpretations (that is, a re-analysis of the original case study) of etic 1, emic 1, and key milestones reported in the original case study. Note that etic 2 comments stress the personalities involved between the two founders and the importance of the Internet community in developing both the open source philosophy and the new firm, Red Hat. The initial DSA, event, and cognitive maps reported below support Exhibit 1.5 as well.

Exhibit 1.6 summarises emic 2 interpretations of mental models and events covered in the original case study as well as the work completed for the etic 2 interpretation – including the DSA, event, and cognitive maps developed for the etic 2 interpretation. The completed long interview with Young (2001) provided the data representing the emic 2 interpretation. Young was CEO and co-founder of Red Hat at the time of the original case-study account and subsequently through to 2002. The revised DSA, event, and cognitive maps presented below follow from the emic 2 interpretations and these maps are part of the etic 3 interpretation (see Exhibit 1.7). Data from the long interview with Young also covers 1999 to 2001.

The new data from the emic 2 and etic 3 rounds of interpretation led to the development of the "open-source marketing" concept—enabling a community of interested parties to rapidly develop, extend, and implement a collective intellectual property; the property context includes permission to modify the basic code or any piece of information.

Exhibit 1.4 Level II Analysis for Red Hat

Etic 1 view:
•Account of evolution of open-source software philosophy – creation of Free Software Foundation (late 1980s); GNU Project (late 1980s); General Public Licence; emergence of open-source initiative (OSI – late 1990s)
•Account of Linus Torvalds' RDC of Linux
•Description of emergence of Red Hat
•Description of Red Hat's application development and distribution

Emic 1 view:
•"The one unique thing we could do that no else could do was, for the first time.
We were giving control of the operating system to the user"
•Successful IPO of a company marketing software that supported free products
(i.e. the Linux open-source operating system)
•"Not bad for a company that gives its products away for free"
•Would an open-source development approach work for developing applications
that could run on Red Hat's Linux operating system?

1993-99. Initial mental model:
•Bundle a set of applications to enable Linux to be easily installed and maintained (1st version released in 1993)
Decisions-actions:
•Start-up of support tools for managing Linux open-source operating; market launch without beta test; RDC of 6 new releases; Absorbed into Red Hat in 1995 on 3rd release (became Red Hat Linux Version 1); Direction and Development of open-source graphic-user interface (GUI – GNOME project 1999); IPO August 1999

Exhibit 1.5 Level III Analysis for Red Hat

Etic 1 view:
Account of Evolution of Open-Source Software Philosophy – creation of Free Software Foundation (late 1980s); GNU Project (late 1980s); General Public Licence; emergence of open-source initiative (OSI – late 1990s)
Account of Linus Torvalds' RDC of Linux
Description of emergence of Red Hat
Description of Red Hat's application development and distribution

Etic 2 view:
•New open-source development community and philosophy
•Red Hat created from complementary needs between Ewing &Young
•Importance of Internet for open-source community application development
•Red Hat's approach to symbiotic interaction with open-source development community

Emic 1 view:
•"The one unique thing we could do that no else could do was, for the first time, we were giving control of the operating system to the user"
•Successful IPO of a company marketing software that supported free products (i.e. the Linux open-source operating system)
•"Not bad for a company that gives its products away for free"
•Would an open-source development approach work for developing applications that could run on Red Hat's Linux operating system?

1993-99 Initial mental model:
•Bundle a set of applications to enable Linux to be easily installed and maintained (1st version released in 1993)
Decisions-actions:
•Start-up of support tools for managing Linux open-source operating; market launch without beta test; RDC of 6 new releases; Absorbed into Red Hat in 1995 on 3rd release (became Red Hat Linux Version 1); Direction and Development of open-source graphic-user interface (GUI – GNOME project 1999); IPO August 1999

Exhibit 1.6 Level IV Analysis for Red Hat

Emic 2 view:
•Reaffirmed all of Emic 1 view with extensions for open-source philosophy at Red Hat and its symbiotic relation with the open-source community
• 90 per cent of Red Hat's Revenues for 2001 came from support activities
•Red Hat is like a fabricator such as Boeing – it assembles over 800 applications of which only about 35-40 are actually created by Red Hat
•Red Hat is building up its 3rd party and ISV (independent software vendor) Programs
•Perhaps Red Hat may develop a marketing structure similar to established software houses

Etic 1 view:
•Account of Evolution of Open-Source Software Philosophy – Creation of Free Software Foundation (late 1980s); GNU Project (late 1980s); General Public Licence; Emergence of Open-Source Initiative (OSI – late 1990s)
•Account of Linus Torvalds' RDC of Linux
•Description of Emergence of Red Hat
•Description of Red Hat's application development and distribution

Emic 1 view:
•"The one unique thing we could do that no else could do was, for the first time, we were giving control of the operating system to the user"
•Successful IPO of a company marketing software that supported free products (i.e. the Linux open-source operating system)
•"Not bad for a company that gives its products away for free"
•Would an open-source development approach work for developing applications that could run on Red Hat's Linux operating system?

Etic 2 view:
•New open-source development community and philosophy
•Red Hat created from Complimentary Needs Between Ewing &Young
•Importance of Internet for open-source community application development
•Red Hat's approach to symbiotic interaction with open-source development community

1993-99 Initial mental model:
•Bundle a set of applications to enable Linux to be easily installed and maintained (1st version released in 1993)
1993-2001. Extended mental model:
•Initial mental model with Release dates for Red Linux V3 (Sep-Dec 1995); V4(Oct 1996); V6 (May 1995)
•Shift to focus on providing Red Hat Linux to major computer hardware vendors for its Enterprise/Large Account Customers (e.g. IBM, Dell); revenues mainly from providing technical support to large customers
Decisions-actions:
•Start-up of support tools for managing Linux open-source operating; market launch without beta test; RDC of 6 new releases; Absorbed intoRed Hat in 1995 on 3rd release (became Red Hat Linux Version 1); IPO August 1999; Red Hat V7 (Sep 2000); V8 (Sep 2002)

Exhibit 1.7 Level V Analysis for Red Hat

Emic 2 view:
•Reaffirmed all of Emic 1 view with extensions for open-source philosophy at Red Hat and its symbiotic relation with the open-source community
• 90 per cent of Red Hat's Revenues for 2001 came from support activities
•Red Hat is like a fabricator such as Boeing – it assembles over 800 applications of which only about 35-40 are actually created by Red Hat
•Red Hat is building up its 3rd party and ISV Programs
•Perhaps Red Hat may develop a marketing structure similar to established software houses

Etic 1 view:
•Account of Evolution of Open-Source Software Philosophy – Creation of Free Software Foundation (late 1980s); GNU Project (late 1980s); General Public Licence; Emergence of Open-Source Initiative (OSI – late 1990s)
•Account of Linus Torvalds' RDC of Linux
•Description of Emergence of Red Hat
•Description of Red Hat's application development and distribution

Emic 1 view:
•"The one unique thing we could do that no else could do was, for the first time, we were giving control of the operating system to the user"
•Successful IPO of a company marketing software that supported free products (i.e. the Linux open-source operating system)
•"Not bad for a company that gives its products away for free"
•Would an open-source development approach work for developing applications that could run on Red Hat's Linux operating system?

Etic 2 view:
•New open-source development community and philosophy
•Red Hat created from Complimentary Needs Between Ewing &Young
•Importance of Internet for open-source community application development
•Red Hat's approach to symbiotic interaction with open-source development community

Etic 3 view:
Etic 2 views validated plus:
• Extension of open-source approach to developing advanced software applications
• Open-source approach applied to software-based information products and services
• "Open-source" marketing concept

1993-99 Initial mental model:
•Bundle a set of applications to enable Linux to be easily installed and maintained (1st version released in 1993)
1993-2001. Extended mental model:
•Initial mental model with Release dates for Red Linux V3 (Sep-Dec 1995); V4(Oct 1996); V6 (May 1995)
•Shift to focus on providing Red Hat Linux to major computer hardware vendors for its Enterprise/Large Account Customers (e.g. IBM, Dell); revenues mainly from providing technical support to large customers
Decisions-actions:
•Start-up of support tools for managing Linux open-source operating; market launch without beta test; RDC of 6 new releases; Absorbed into Red Hat in 1995 on 3rd release (became Red Hat Linux Version 1); IPO August 1999; Red Hat V7 (Sep 2000); V8 (Sep 2002)

1.4 DSA, Event, and Cognitive Mapping Before and After Emic 2 Interpretations

This section covers the development and revisions of the DSA, event, and cognitive maps from re-analysing the original case report and following the long interview with the CEO of Red Hat (Young 2001).

Figure 1.1 shows a part of Red Hat's organisational structure and a few of the participants involved in the development of the firm's products. Figure 1.1 and related exhibits were developed as a step in re-analysing the original case report to clarify the roles of specific participants in the enterprise. Figure 1.1 reflects the substantial complexity of human relationships possible within and between departments in the firm.

Figure 1.2 summarises a DSA map showing the planning and event steps along with the principal feedback loops, as developed from re-analyses of the original case. This DSA map becomes data for discussion and elaboration in the long interview with Young (2001).

The emic evaluations of the original DSA map led to the revisions shown in Figure 1.3. Note that Figure 1.3 includes the use of **bold** to direct attention to the revisions and additions to the flow of decisions and events created in the first re-analysis as reported in Figure 1.2.

Figures 1.4 and 1.5 are the original and revised events chronology maps. The events shown in Figure 1.4 are limited to the time period covered in the original case-study report. Figure 1.5 serves as a revision (including corrections and additions) of the events and chronology found in the original case as well as an update on what happened subsequently during the two years following from the original case report. Figure 1.5 shows changes from Figure 1.4 in bold for text and new data representing the etic 3 interpretations and following from emic 2 interpretations.

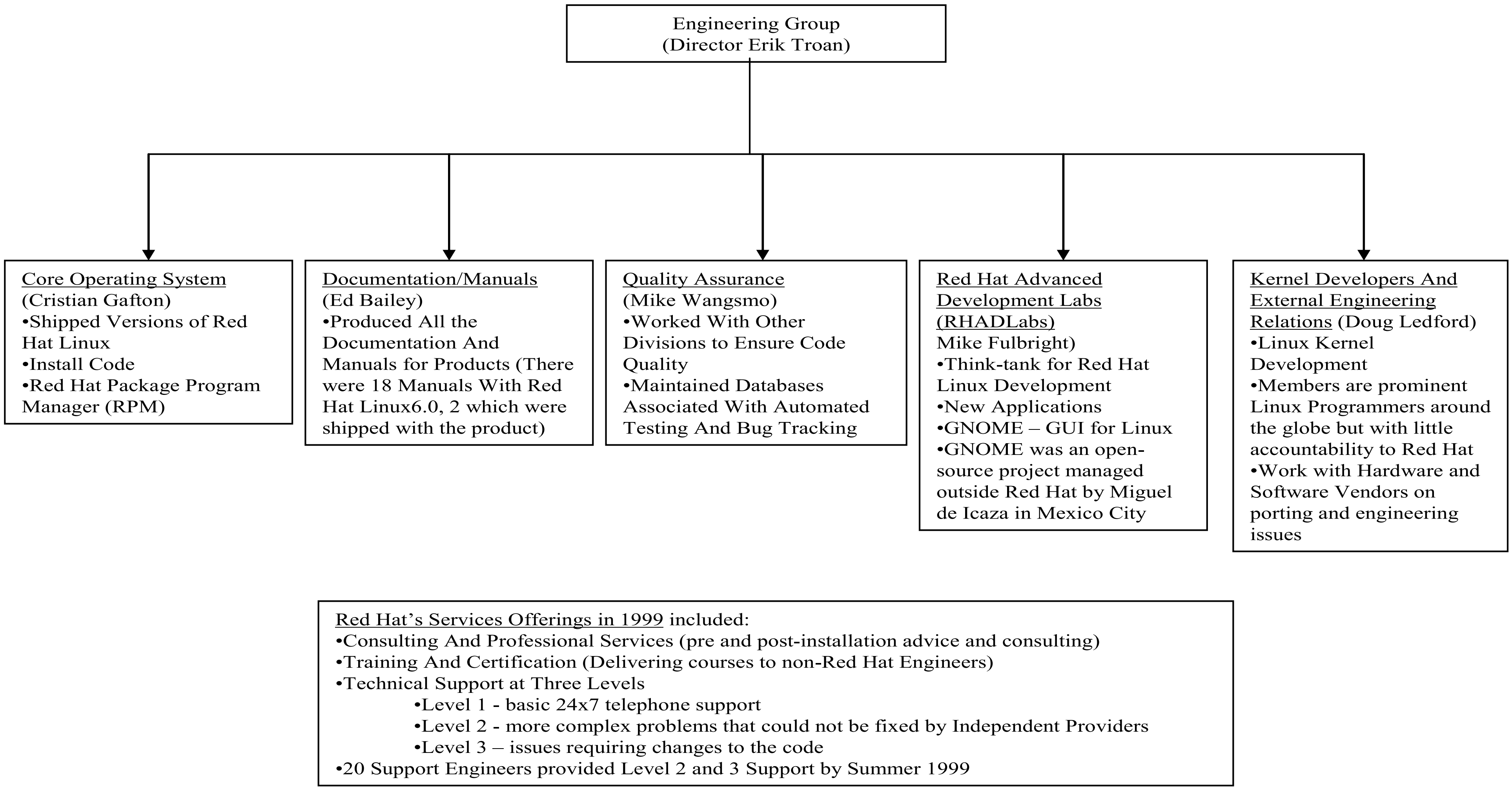


Figure 1.1 Red Hat Engineering Group – Structure And Services (1999)
Created from data in MacCormack & Herman (2000), p. 9.

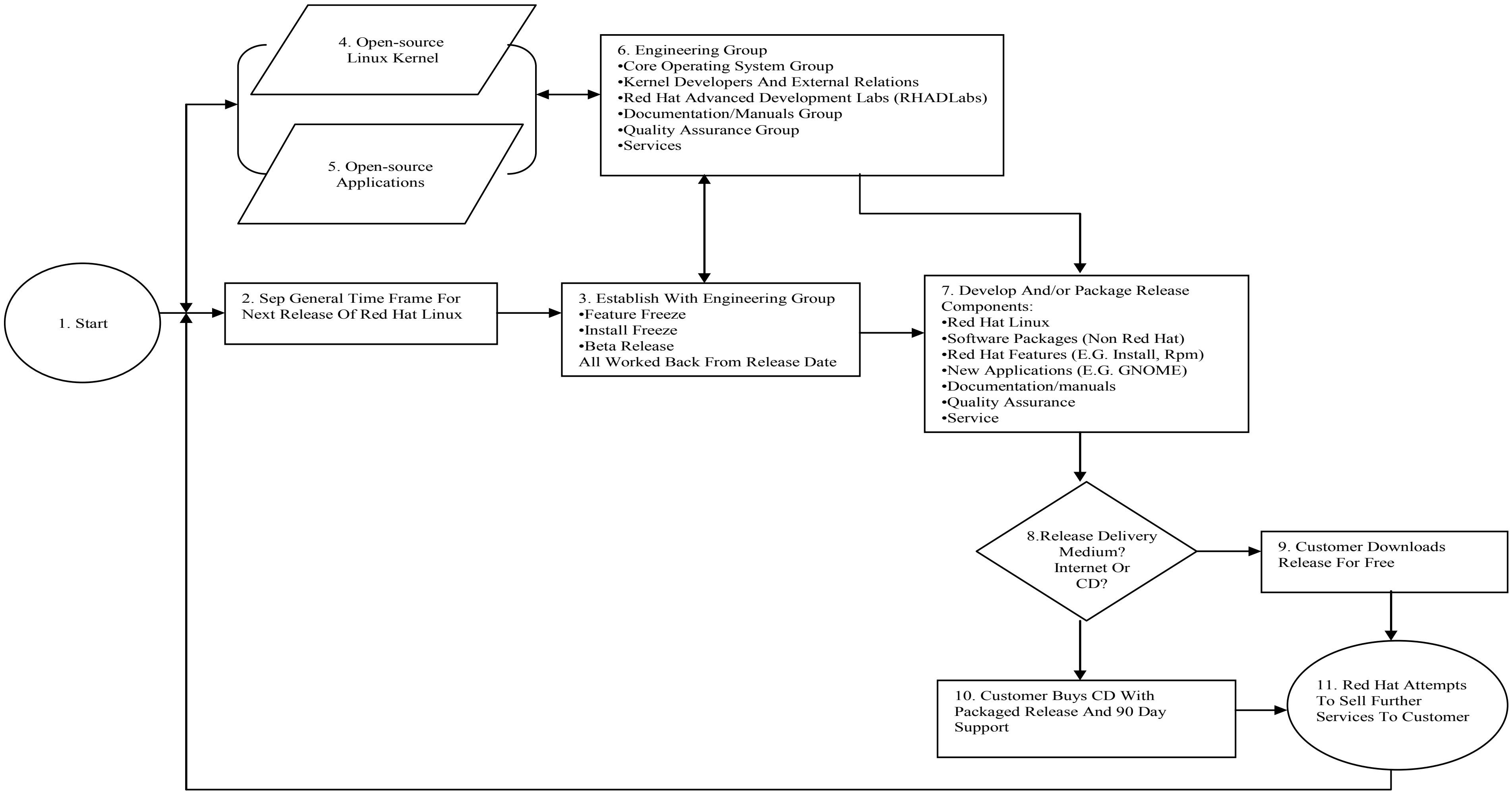


Figure 1.2 Summary DSA Model For Red Hat Application Development (1999)
Created from data in MacCormack & Herman (2000).

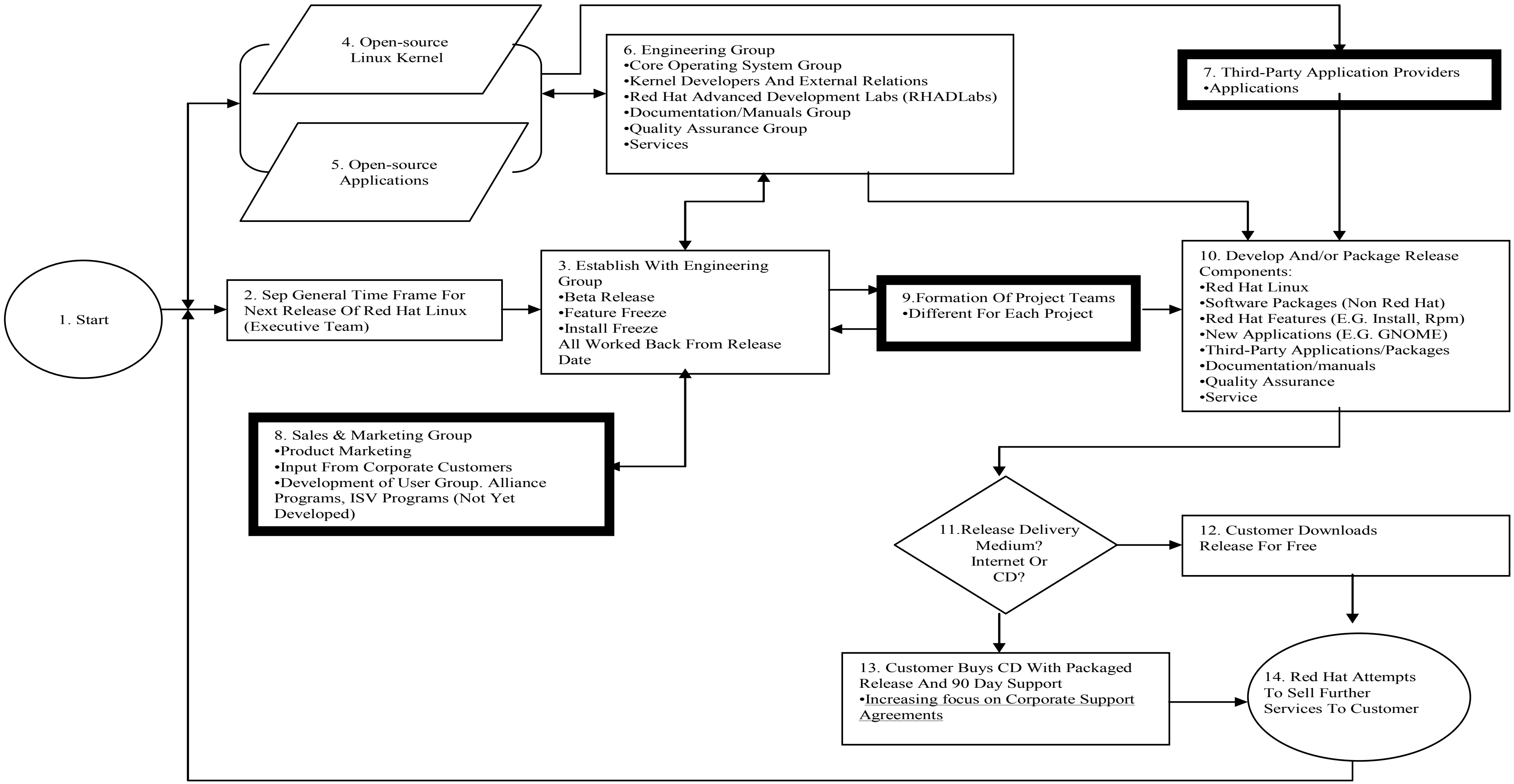


Figure 1.3 Summary DSA Model For Red Hat Application Development 2001
Created from data in MacCormack & Herman (2000) and Young (2001).

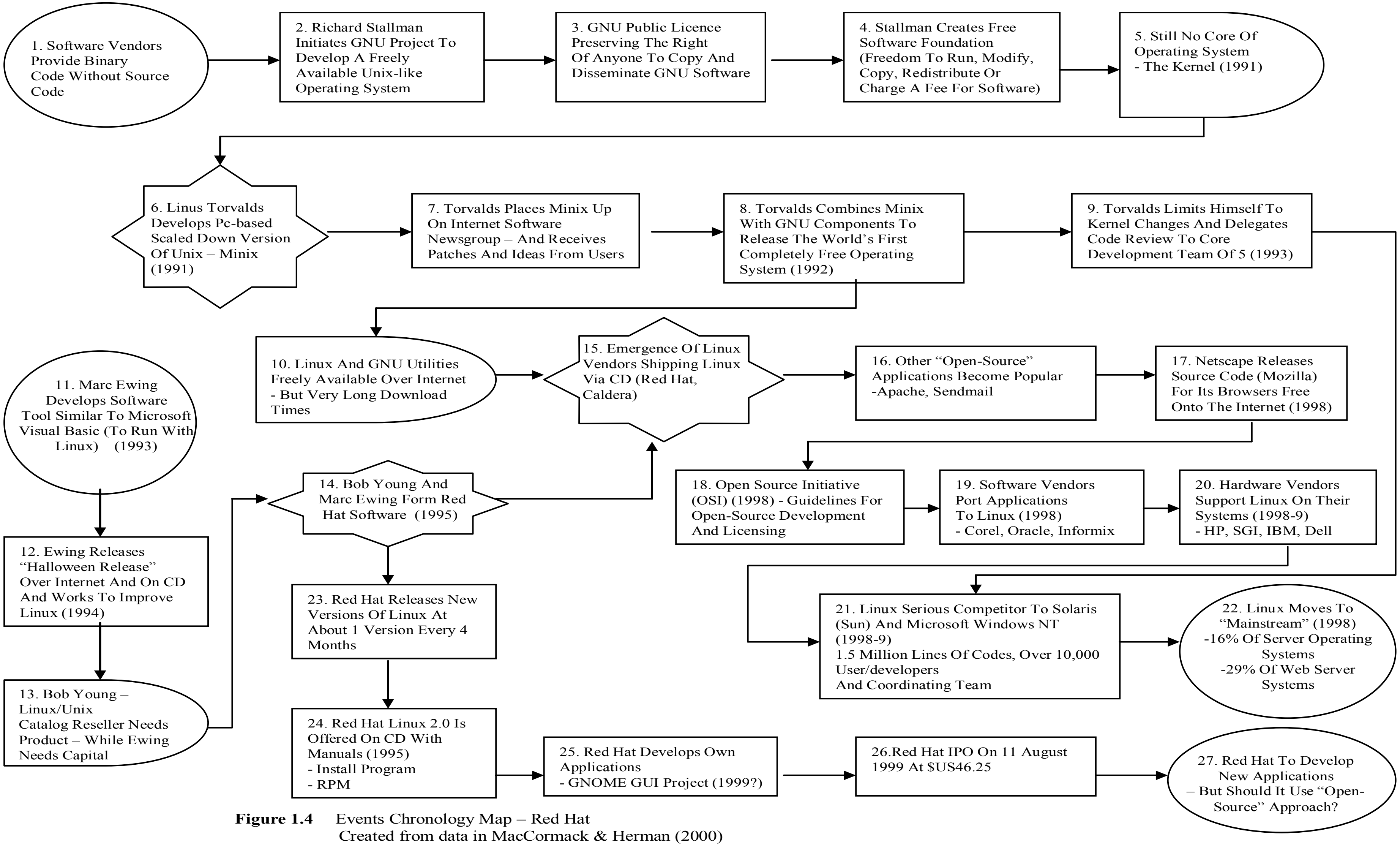


Figure 1.4 Events Chronology Map – Red Hat
Created from data in MacCormack & Herman (2000)

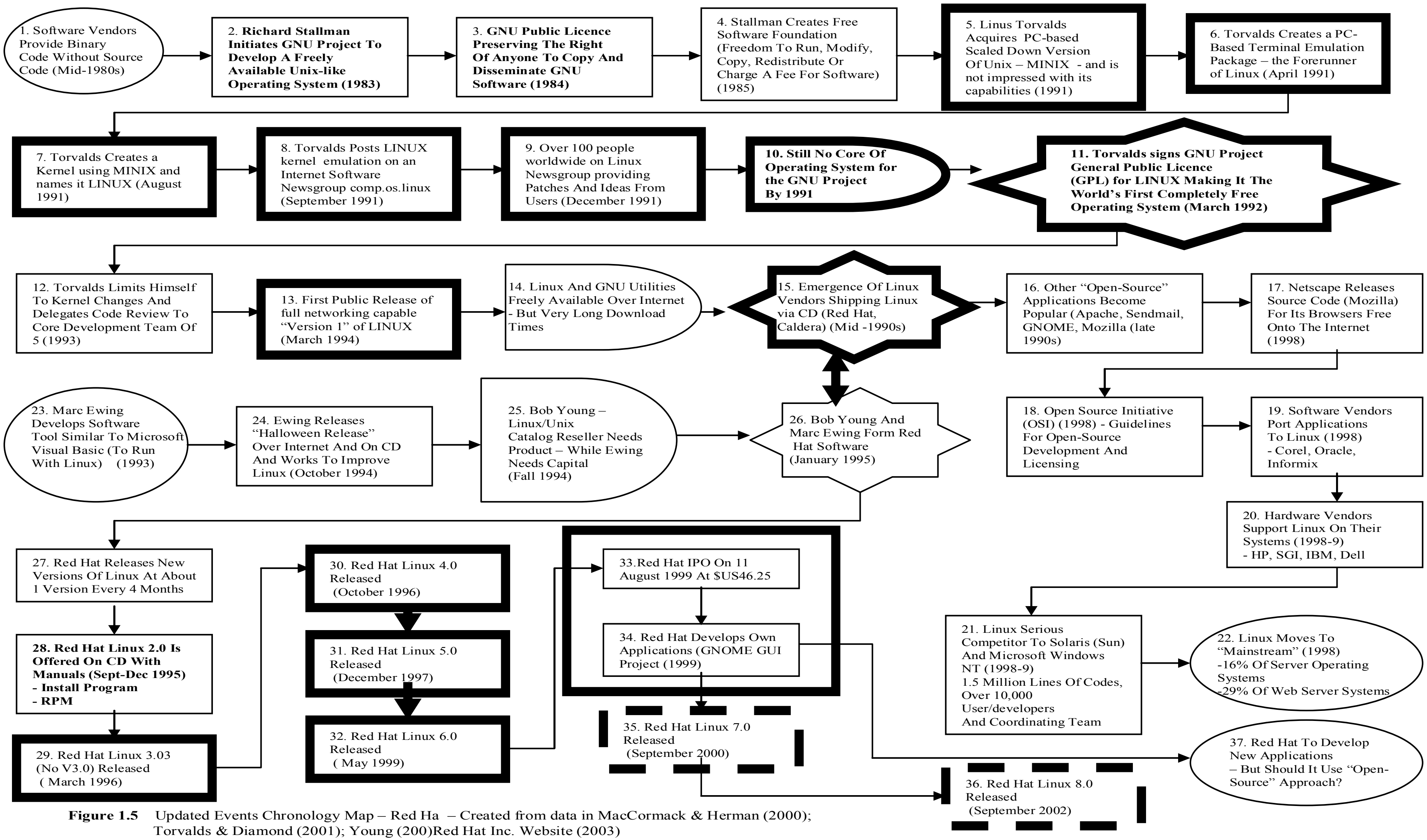


Figure 1.5 Updated Events Chronology Map – Red Ha – Created from data in MacCormack & Herman (2000); Torvalds & Diamond (2001); Young (200)Red Hat Inc. Website (2003)

The remaining exhibit, the table, and the figures illustrate the cognitive mapping steps completed for the hermeneutic interpretation. Exhibit 1.8 is an extract from the original case study by MacCormack and Herman 2000). Following from a TACT analysis, Table 1.1 provides a word frequency account of key concepts appearing in the extract. Exhibit 1.8 is just one extract of several made from the original case study from which TACT and NVivo analyses were applied, with the resulting cognitive maps used during the long interview with Young (2001).

Figure 1.6 presents the cognitive map of micro-events and the development of the mental models of the participants that resulted from an NVivo analysis of the extract shown in Exhibit 1.8. Figure 1.7 represents a revised version of Figure 1.6 that was created for clarifying the cognitive maps and for use in the long interview with Young. Figure 1.8 follows from the emic 2 interpretations by Young; Figure 1.8 summarises the etic 3 interpretation following an analysis of data from the long interview with Young.

Revisions and new information were added into Figure 1.8 compared to Figure 1.7. For example, Bob Young's Linux/UNIX catalogue reseller company is noted as the ACC Corporation incorporated in 1993 (see box 4 in Figure 1.8). Ewing (the other Red Hat co-founder) developed an installer program and a tool to upgrade and track new versions of Linux (RPM – see box 3 in Figure 1.8). The ACC Corporation was renamed Red Hat in January 1995 (see box 6 in Figure 1.8). Intel, DEC and Sun Microsystems were identified as ported hardware platforms for Red Hat Linux V4.0 (see box 9 in Figure 1.8). Figure 1.8 reflects more details and insights than Figure 1.7 based on the emic 2 evaluation and review comments of Figure 1.7 by Young (2001) and the final etic 2 interpretation.

Overall, the updated cognitive map captures the decision-making associated with Ewing creating a version of Linux that was easy to install and maintain, plus Young's need for a competitive Linux product. Ewing and Young's motivations were complementary; with Young's renamed company Red Hat taking on further development of Linux with Ewing's installation and development software tools.

Exhibit 1.8 Text Extract: Red Hat – Early Development (1993-96)

In 1993, Marc Ewing, a young programmer out of Carnegie Mellon decided to leave his contract job at IBM. He wanted to resurrect an earlier project he had been toying with all through school: attempting to build a software development tool similar to Microsoft's Visual Basic program. Given he didn't have the $10,000-$20,000 it cost for a Hewlett-Packard or Sun Workstation necessary to run UNIX, the operating system he needed for his project, Ewing decided to install Linux on his personal computer. But he found the software difficult to install and even more difficult to use-for every hour of work he completed on his project he found himself spending two or three hours fixing and upgrading Linux. "It became obvious that I was doing the wrong thing. The world really needed a better Linux." Ewing decided to switch his attention to just such a project, putting together his first release on a CD by October 1994, which he nicknamed The Halloween Release, and selling it over the internet.

Soon Ewing was looking for better ways to distribute his version of Linux. At the Fall 1994 UNIX Expo, he met up with Bob Young who was at that time running a Linux/UNIX catalogue reseller business. Young had first heard about Linux as part of his effort to cover breaking news for his UNIX usergroup newsletter. He had noticed that while there was no growth in the demand for UNIX-based systems in general, growth and interest in Linux was growing extremely rapidly:
I think of it an awful lot like the car industry. You could buy your tires from Michelin, your airbags from TRW, and so on. But very few of us actually build our own cars. We tend to go to a dealer and buy a FORD or a BMW. Of course, the engineers at Red Hat will tell you that we build the BMW. - Bob Young, CEO Red Hat Inc.

You could see that if this kept going Linux was going to end up in CompUSA, and as just a little reseller of Linux I wasn't going to be in a very tenable position. I needed a product that that I could market to CompUSA once Linux became legit, and some of my customers were telling me that Marc's product was better than Slackware, the dominant version of Linux at the time. Intrigued by his customers' comments, Young ordered 300 copies of Ewing's Halloween Release in the fall of 1994. Given that Ewing needed capital and Young wanted a product, the match seemed perfect. Their relationship was quickly formalised by January of 1995, when Young's company purchased Ewing's copyrights in exchange for shares, renamed the company Red Hat Software, and moved its headquarters to Raleigh, North Carolina. As Young recounted:
We were a little company at that time, it was a little deal. Both of us thought we were taking advantage of the other guy. I had this company that wasn't making any money, and Marc had this product no one wanted.

Exhibit 1.8 Text Extract: Red Hat – Early Development (1993-96) (Cont'd)

The newly-founded Red Hat continued their rapid series of Linux releases, on average keeping to a four month schedule for each new version (see Exhibit 5). This was in keeping with one of the main appeals of the open-source approach: the rapid progress of the main code base. Ewing's initial Halloween Release was followed in early May of 1995 with the Mother's Day Release, and two months later with Mother's Day Plus One. At the end of September 1995, the quirky labelling system was replaced with a more traditional numbering system, and Red Hat 2.0 was released, retailing at $39.95. The 2.0 release marked the first time a manual was provided with the software.

The leaked memos became known as the "Halloween" documents. They outlined Microsoft's assessment of the threat of open-source development methods, creating quite a stir in the business press.
These explanations can be perfectly consistent with the notion that contributors are acting in their own self-interest. As Raymond suggests, "the utility function Linux hackers are maximizing is not classically economic, but is the intangible of their own ego satisfaction and reputation among other hackers."
Red Hat 4.0, released in October 1996, finally marked the company as profitable. The product included the new Linux 2.0 kernel and shipped on a variety of hardware platforms, including alpha, SPARC, and Intel. It was an extremely successful release; along with Infoworld's product of the year award, an industry research group [IDC] tied Red Hat's Linux with Windows NT as "product of the year," bringing Linux into direct comparison with Microsoft in the operating system market.

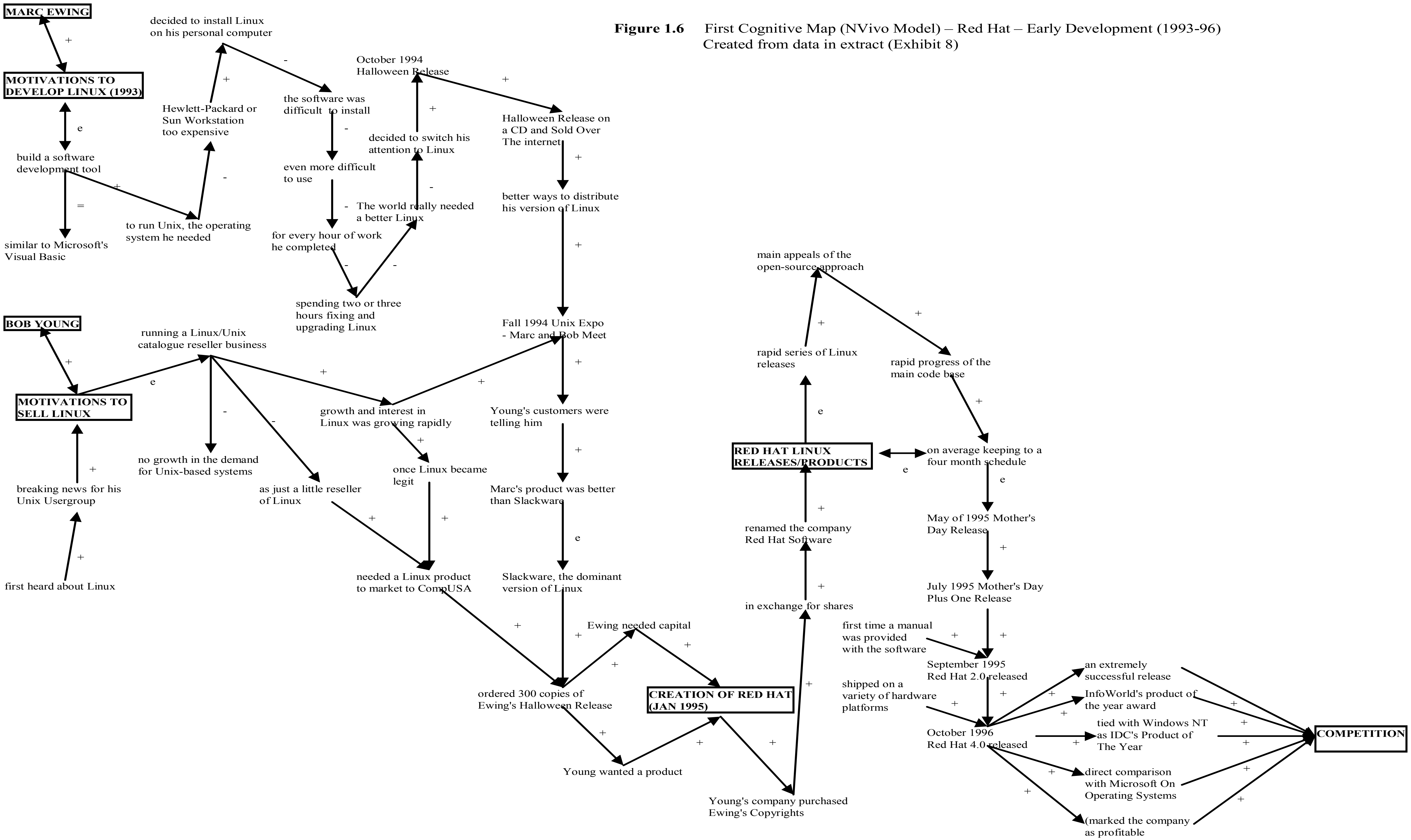
Source: MacCormack & Herman (1998, 6-7).

Table 1.1 Red Hat : Early Development (1993-96) – Ranked Word List

Anchor Concepts	Frequency
Linux	16
release	10
Ewing	8
Red	8
Young	8
Hat	7
product	7
company	5
system	5
Unix	5
time	4
Halloween	4
need	4
project	4
software	4
its	4

Source: data analysis performed using TACT (1997).

Figure 1.6 First Cognitive Map (NVivo Model) – Red Hat – Early Development (1993-96)
Created from data in extract (Exhibit 8)



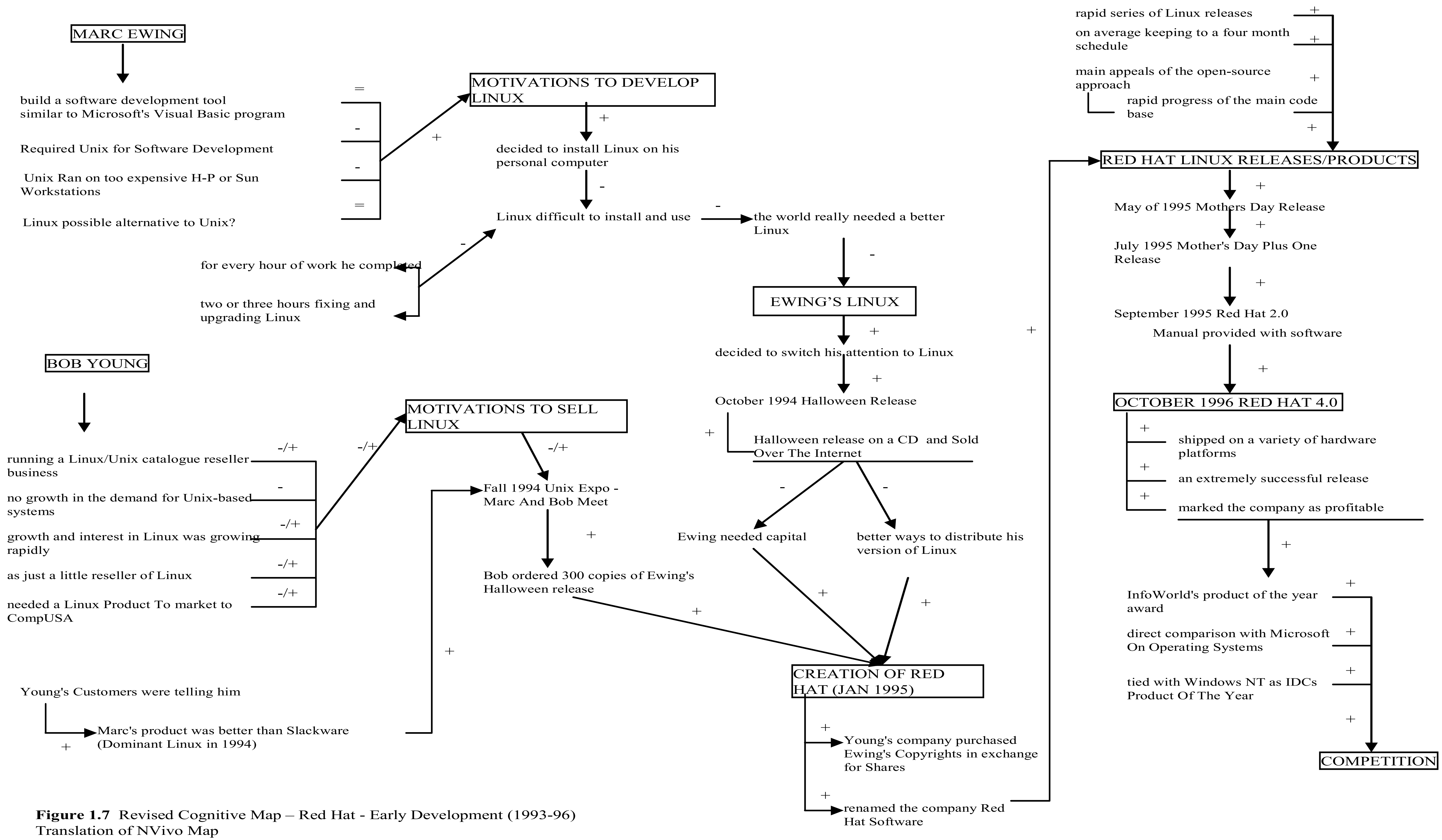


Figure 1.7 Revised Cognitive Map – Red Hat - Early Development (1993-96)
Translation of NVivo Map
Note. This revision was shown to participant in emic 2 stage for his (Young's) interpretation.

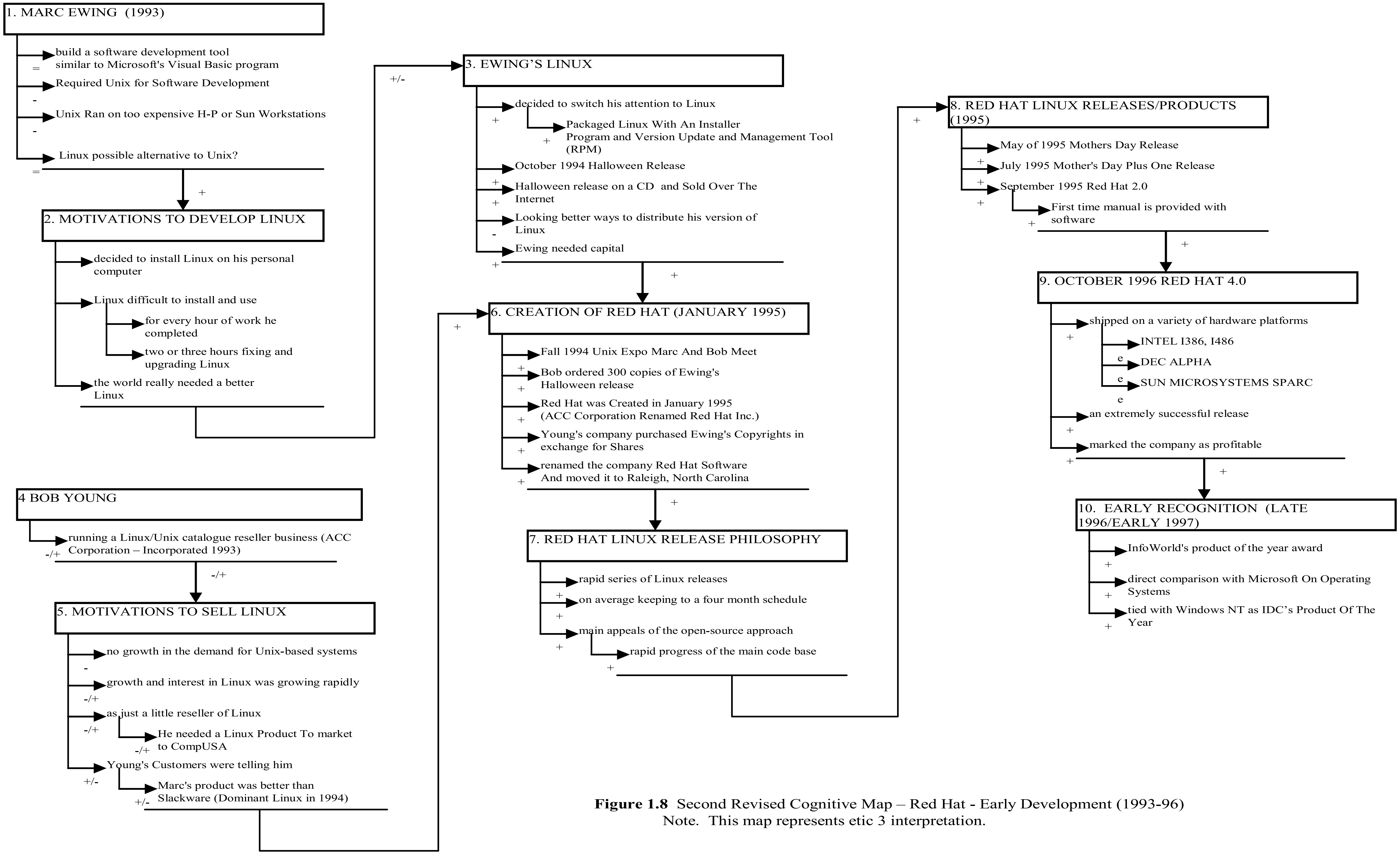


Figure 1.8 Second Revised Cognitive Map – Red Hat - Early Development (1993-96)
Note. This map represents etic 3 interpretation.

1.5 Conclusions for the Hermeneutic Interpretation of the Extended Red Hat Case Study

Decision-making at Red Hat associated with application development supported literally a real-time development and update online environment for Linux. Although packaging Linux with Red Hat's upgraded applications was not so real-time, application development cycles of about two to four months were quite rapid when compared to non-open-source-related application cycles.

The extended Red Hat case study highlights a broad, but variable and complex form of dominant logic. According to Young (2001) the dominant logic within Red Hat varied by project and for each project team. Marc Ewing and Paul Cormier were noted as significant contributors for dominant logic related to technology issues in Red Hat – but even then that was only for the final assembly of all the programs into the application.

Young (2001) views Red Hat as a highly visible organisation within the open-source community, to be heavily influenced by – but also contributing to – that community's collective dominant logic. A single dominant logic is much more difficult to define in relation to the open-source community where collectively tens of thousands of developers are working to create new and improved applications.

Comments from the cognitive map analysis on Torvalds' creation of Linux suggest that, for the open-source community, coordinators and reviewers consolidate and integrate application developments. The actual level that these coordinators actually drive application development direction on, as against creating a collective logic drawn from the community's inputs, would be worthy of further research.

From its inception in 1995, Red Hat has focused mainly on delivering value-added tools to open-source applications. Young (2001) sees an analogy between Red Hat as an applications version of a discrete manufacturer such as General Motors or Boeing assembling many parts from various suppliers into a distinct product.

Over 800 applications are available within the Red Hat Linux application, and Young believes that shared vision is incorporated from all of the suppliers contributing to it. Combining a high level of third-party shared vision with open-source shared vision (which is assumed here to be the same as open-source dominant logic) with the technical fabrication vision within the Red Hat firm produces quite a different type of vision when compared to the etic 1 view provided in the original case study.

1.6 Conclusions, Limitations, Strategic Implications, and Suggestions for Further Research

This chapter illustrates the details of applying hermeneutical interpretations in B2B contexts to deepen the understanding and description of specific decision-making processes. The appeals by Langley et al. (1995) for opening up decision-making inform the method developed and applied for this report—especially the suggestion to re-analyse previously analysed decision processes, not just new ones.

Advancing hermeneutical interpretations is possible, especially in decision-making research through the use of DSA, event and cognitive mapping and collecting additional emic interpretations of original emic reports, as well as emic interpretations of early-on etic interpretations. Uses of such procedures are well known in the systems dynamics literature (for example, Hall 1976; Sterman 2000) though not placed by this literature into a hermeneutic framework.

This introduction represents only a portion of the complete analysis of the extended Red Hat case – which is but one of six cases in this research project. Limitations related to this snapshot of the Red Hat case include not using outside auditors to evaluate etic 3 interpretations, which would be a good addition of an etic 4 level of interpretation (cf. Brinberg and Hirschman 1986).

Further reporting of DSA and findings on the Red Hat case study can be found in Chapter 4. The other five case studies in the project contribute insights from a range of different development approaches for Internet-based software – but are greatly strengthened through using the new advanced hermeneutical framework. The case studies cover applications development and delivery at:

- Red Hat Inc. (Chapter 4)
- Zaplet Inc. (Chapter 5)
- NetDynamics Inc. (Chapter 6)
- Kana Communications Inc. (Chapter 7)
- Intuit Inc. (QuickBooks Chapter 8)
- Trilogy Inc. (Chapter 9)

The Theory Generalisation Development and Conclusions chapter (Chapter 10) discusses cross-case findings and summaries, and develops theory through assessment of a set of Structuration theories, and new propositions. The concluding discussion addresses limitations and recommendations for future research.

A small degree of meta-evaluation of a series advanced hermeneutic B2B research reporting on a specific issue (for example, new product innovation processes) is provided in the Theory Generalisation and Development chapter. However, projects with much larger scale meta-evaluation of a series of advanced hermeneutic B2B research reports would provide an advance for developing a grounded theory of what happened and why it happened. Such a large-scale research effort should enable more rigorous, accurate and useful generalisations of decision-making on a specific issue than is found in literature reviews of models of complex systems.

Chapter 2: Literature Review

2.1 Introduction

This research project draws upon a diverse range of knowledge areas including:

- application software and new product development (including the Internet)
- storytelling methodology with emphasis on hermeneutic analysis
- sensemaking and cognitive theories of the firn
- marketing decision systems analysis (DSA)
- cognitive mapping of strategic thought
- extension issues:
    - dominant logic
    - shared vision
    - leverage points
    - relevant strategic marketing issues

Figure 2.1 presents the structure of this literature review chapter, including the main areas of the review.

2.2 New Product and Software Application Development Research

New product development covers a wide and diverse range of knowledge areas. Krishnan & Ulrich, in their comprehensive literature review on the area, define "Product Development" as "the transformation of a market opportunity into a product for sale" (Krishnan & Ulrich 2001, p. 1).

In Krishnan & Ulrich's study, generally, research related to high-technology products was located in the areas of concept development, product design, product development, organisation, and project management. Relevant high-technology product development papers, noted by Krishnan & Ulrich, and also reviewed for this project included Brooks (1975); Clark (1985); Von Hippel (1986); Dougherty (1989); Clark & Fujimoto (1991); Wheelwright & Clark (1992); Eisenhardt & Tabrizi (1995); and Iansiti (1995a & b – 1995b was developed into a book, that is, Iansiti 1998).

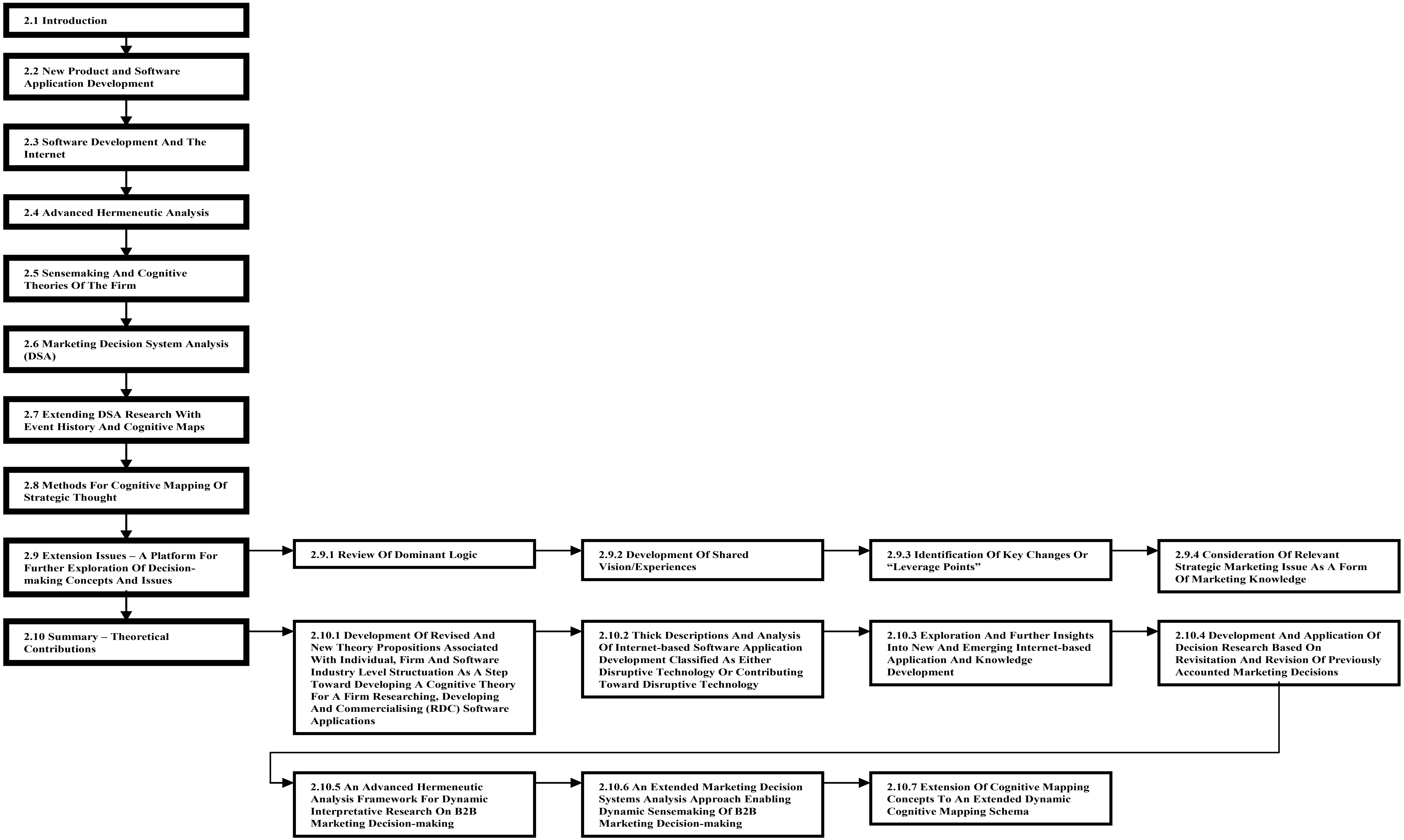


Figure 2.1 Literature Review – Chapter Structure.

Krishnan & Ulrich (2001) classify Iansiti's 1995a research as contributing to product development organisation – but it should also be noted that their definition of product development excludes external participation in product development.

Product development with external participation was regarded by Krishnan & Ulrich as in "innovation" literature. In a separate literature review classification, Shane & Ulrich (2004) review innovation, product development and entrepreneurship – all strong themes in the research project – and Iansiti gains more recognition in the innovation area where product design and implementation is located across a number of businesses.

A major stimulus for undertaking the research project was a strong belief that software applications were not a product, in terms of standardised assembly and manufacturing, but a medium that contains codified knowledge (see Armour 2000). Perhaps software development, viewed as a "medium of knowledge creation", might be different from conventional product development.

Some researchers (for example Iansiti & MacCormack 1998) go a step further, suggesting that rapid software application development (sometimes referred to as product development in "Internet time") is indeed a new and distinct form of new product development. This view points toward the emergence of a new and distinct knowledge creation system. These systems are based on the integration of technical possibilities and customer expectations, expressed as software applications in an Internet-based software development environment (Iansiti & MacCormack 1998).

Iansiti contrasted "traditional" product development as a set of sequential activities where a product concept is clearly defined and frozen before the product is released to the market, against an overlapping or concurrent set of linked development and delivery activities (Iansiti 1998, pp. 185-6). Figure 2.2 presents both the traditional and flexible models of product development.

Iansiti emphasises that his model of flexible development goes further than other concurrent product development models. Models advocated by, for example, Cooper (1983, 1990); Clausing (1994); Clark (1985); and Clark & Fujimoto (1991), highlight joint participation of product development processes. Iansiti highlights that his flexible development approach forces *simultaneous* execution of product conceptualisation and implementation (Iansiti 1999, pp. 186-7).

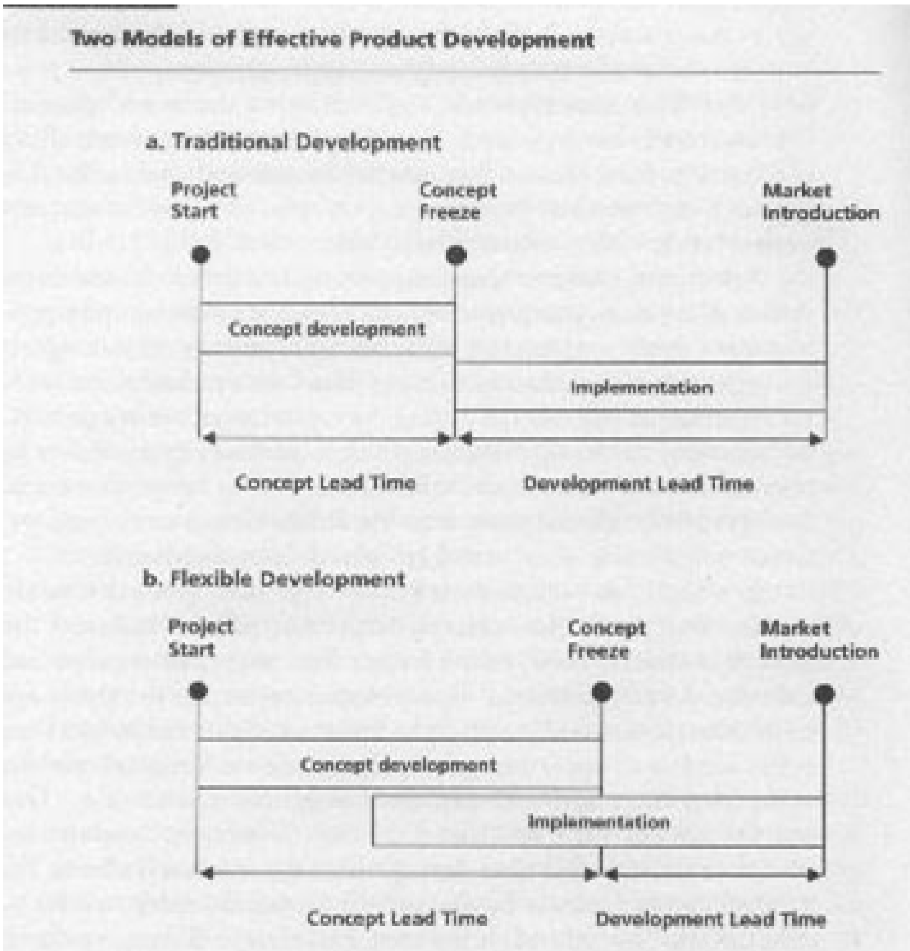


Figure 2.2: Two Models of Effective Product Development
Source: Iansiti (1998, 186).

Iansiti's research on rapid software development at Netscape, Yahoo!, NetDynamics, and Microsoft (see Iansiti & MacCormack 1996), was indeed a very strong inspiration for creating this research project, as it demonstrated distinct differences for Internet-based application development, over both conventional product development, and non-Internet-related application development.

A large field of knowledge covers software application development processes with a strong systems theory heritage. Classification of software development processes into specific methodologies emerged in the late 1960s and into system life cycles in the late 1970s and early 1980s (see Brookes et al. 1982). System life cycles incorporated both hardware and software development processes, but through the late 1980s and early

1990s, system life cycles concentrating on software development came into vogue as software process.

The earliest software process approach was the linear system life cycle (also known as the waterfall system life cycle), in which major phases of software development and delivery are defined into sequences of consecutive phases (Hawryszkiewycz 1998).

Pressures to improve software development productivity have persisted since the early 1980s, and variations of the waterfall approach to address this need have included:

- attempting to parallel phases of software development
- deconconstructing development into subsystems that may be able to be then developed simultaneously
- experimentation and prototyping of applications early in the overall development process.

Most recent software processes (for example, extreme programming) focus on the rapid prototyping of user scenarios, followed by multiple rebuilds of the application with close user/developer interaction, until the application is accepted in a final release form (see Xtreme Programming 2002).

2.3 Software Development and the Internet

This dissertation focuses on Internet-based software applications. The Internet was transformed from a research-oriented academic network in the early 1990s, to a global public network with multimedia software storage and distribution capability.

Development of the Internet's multimedia service, the World-Wide Web, in the early 1990s, heralded a new infrastructure through which software could be created, distributed, changed and stored in different locations, all in real-time (see Berners-Lee (1999) for an account of the development of the World-Wide Web). For software developers, the Internet represented opportunities and challenges in at least two dimensions. One dimension was to rapidly develop applications to enhance the capability of the emerging Internet/World-Wide Web software environment.

The Internet represents a new computer-mediated environment (CME). The Internet can be regarded as a business-oriented CME (see Hoffmann & Novak 1996 for their discussion on marketing in computer-mediated environments with the Internet as an exemplar).

The second dimension is actually using the environment to develop new multimedia-based applications for consumption from the environment. Both dimensions gained the attention of business researchers looking for new product development approaches and new business models. Several Internet-based software vendors have been the subjects of substantive case-studies that have highlighted software application development processes and issues. Examples include Iansiti & MacCormack (1998); Iansiti (1998); Cusumano & Yoffie (2000); MacCormack & Herman (2000).

2.4 Advanced Hermeneutic Analysis

Hermeneutic (or types of "explanatory" and "interpretative") analysis is useful for systematic and extensive interpretation of data, particularly if the data is in the form of a communications discourse.

Holbrook & O'Shaughnessy (1988) describe the processes of hermeneutic analysis including "hermeneutic circles" as a valid approach for interpretative consumer research. Arnold & Fischer (1994); Thompson, Pollio, & Locander (1994); and Thompson (1997) apply hermeneutic circle interpretations also applied in developing and reconstructing consumers' stories and deriving marketing insights. Hackley (2003) notes that advertising agents use a form of hermeneutic analysis, to imaginatively reconstruct consumers, experiences, through a set of interactions with those consumers. These collective interactions over time can be viewed as a "hermeneutic circle" where the final reconstructions have been produced from analysing and reviewing these sets of interactions.

Hermeneutic analysis has also been used for discourse analysis of strategic decision-making to enhance sensemaking (see Hendry 2000, Phillips & Brown 1993). Hendry (2000) highlights three perspectives of strategic decision-making – the "traditional" rational perspective, plus more recent action, and interpretive perspectives (see Table 2.1). Hendry (2000) brackets hermeneutic discourse analysis with rhetoric analysis (Mohrmanm et al. 1973), metaphorical analysis (Lakoff & Johnson 1980), conversational analysis and interaction analysis, as interpretative research resources for the analysis of strategic decision-making discourse.

An advanced hermeneutic analysis for the study of B2B decision-making processes was developed for the project. The framework adds multiple rounds of interpretative discourse – not just for an observer running multi-round interpretations and challenges against their data – but with direct involvement and revisitation from the actual decision-makers.

Table 2.1: Three Perspectives on Strategic Decision-making

Rational perspective	
Conceptualisation of strategy	Rational – instrumental (strategy realised as enacted plan)
Conceptualisation of decision-making	Rational intentional choice
Key assumptions	Objective view of reality. Intentional decisions are made, exist, precede and provide primary context for decisive acts of implementation
Focus	Intentional decisions, as identified retrospectively by actors and researchers
Action perspective	
Conceptualisation of strategy	Emergent (strategy realised as pattern of behaviour)
Conceptualisation of decision-making	Decisions are decisive acts, intentions are irrelevant
Key assumptions	Objective view of reality. Reject rationalism of traditional perspective but without offering anything in its place
Focus	Decisive acts (for example, commitments of resources)
Interpretative perspective	
Conceptualisation of strategy	Emergent (strategy realised as pattern of behaviour) or cognitive (strategy as shared cognitive schemas)
Conceptualisation of decision-making	Interpretive
Key assumptions	Social constructionist view of reality. Apparent Intentional decisions are retrospective rationalisations of prior actions, necessary for individual and collective sensemaking and/or legitimation
Focus	Decision statements as outcomes of cognitive processes

Source: Hendry (2000, p. 958).

2.5 Sensemaking and Cognitive Theories of the Firm

The advanced hermeneutic analysis framework proposed for application to B2B decision-making focuses on highlighting dynamic sensemaking at an individual level that symbiotically relates to a cognitive perspective of behaviour within firms.

Since the mid-1980s, researchers have provided frameworks for understanding and undertaking cognitive research within firms. Mintzberg (1990) and Lyles (1990) apply cognitive perspectives to business strategy, and Minztberg, Ahlstrand & Lampel (1998) note increased applications of cognitive perspectives to other areas of business activities during the 1990s.

Various researchers have attempted to encapsulate cognitive perspectives into a framework, mainly using schematic representations. Figure 2.3 highlights a broad organising framework for cognitive research presented by Huff, Huff & Barr (2000) (also adapted from Walsh 1995).

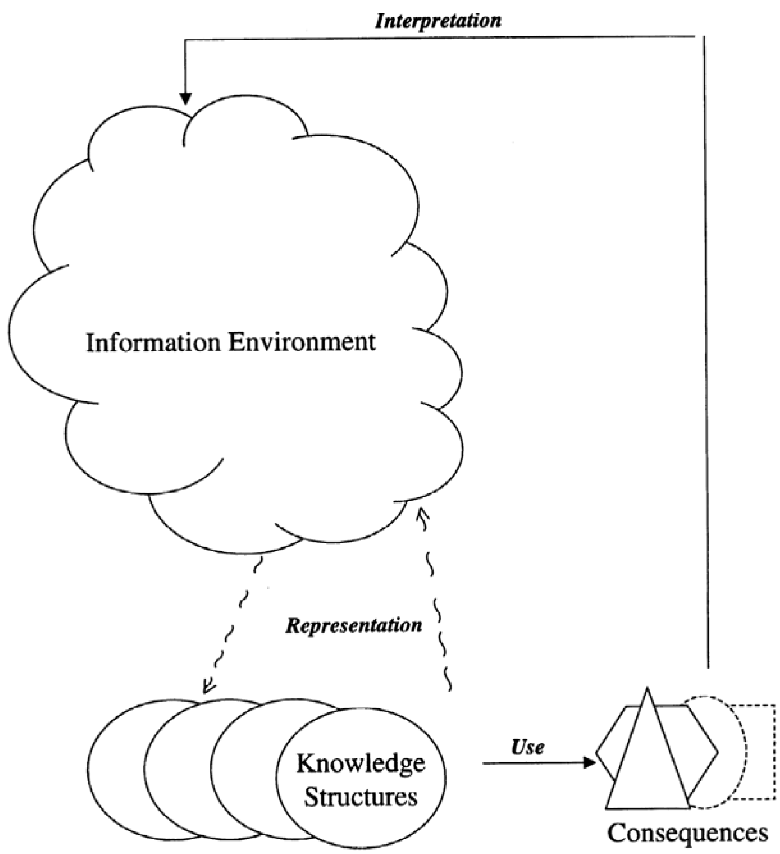


Figure 2.3: An Organising Framework for Cognitive Research
Source: Huff, Huff & Barr (2000, p. 16).
Modified from Walsh (1995).

For some researchers, the development of a range of schematic representations of firm-based cognitive activity is really the mapping of thought within the firm. Huff (1990) focuses on mapping strategic thought, with Weick (1990) providing justification of cartographic (or mapping) representations of "theories of action".

As a commentator and encapsulator of emerging trends on mapping strategic thought from a cognitive perspective, Huff (1990) effectively chronicles transitions from mapping of cognitive perspectives in Huff (1990). Huff articulates potentially distinct cognitive theories of the firm (Huff, Huff & Barr 2000). Huff and Jenkins (2002) update mapping techniques and put forward a cognitive model for individual strategic decision-making.

Huff, Huff & Barr (2000) suggest that no specific cognitive theory of firm has been formally developed but they describe five candidates for consideration (see Table 2.2).

The present research project draws ideas and concepts from at least three of the potential cognitive theory areas of knowledge acquisition and use, sensemaking, and change. The main theory area applied to the project is a sensemaking cognitive theory of the firm.

Weick (1995) not only discussed mapping "theories of action" (Weick 1990) but he also interpreted models around an organisation's sensemaking of its interactions with itself and its environment. Figure 2.4 represents Weick's (1995) stimulus–response (S–R) model of such interactions. This dissertation explores executives and founders of a selected set of Internet-based software houses and their sensemaking of decision-making related to the development and delivery of applications. Sensemaking is captured initially through mapping existing retrospective accounts of decision-making processes, and gathering new retrospective accounts of both the original decision-making processes as well as the original retrospective accounts, via additional emic data collection from the founders and executives.

This dissertation probes the ability of the dynamic data collection method to gain deep insight into and a rich understanding of decision-making. The rationale that is relevant here is that revisions of sensemaking by DM participants can result in unearthing or clarifying insights that were not necessarily clear or obvious at the time of the original set of actions, or at the time of the original retrospective account.

Table 2.2: Potential Cognitive Theories of The Firm

Theory	Orientation	Process of Discovery	Central Research Problem	Theoretic Conceptualization of the Firm	Manager's Motives/Behavior	Goals of Firm	Nature of Environment
Decision making and choice	Primarily positive/ some interpretive	Inductive/ deductive	What are the characteristics of rational analysis and problem solving? What procedures can overcome bias in decision-making processes?	A boundedly rational actor	Establish reliable decision-making procedures/focus on representing the task environment, specify-ing goals, searching for alternative actions, choosing the best alternative to fit goals	Survival, profit	A setting that establishes temporal decision points and offers alternatives for action
Culture	Positive/ interpretive	Inductive/ deductive	Understanding the assumptions and basic vocabulary that create cohesion	A social system re-sponding to a larger culture, creating its own culture and subcultures	Use symbols to mobi-lize support/subject to cultural symbols and assumptions	Grounded in and limited by culture	Source of and transmitter of symbols
Knowledge acquisition and use	Positive/ interpretive	Deductive/ inductive	What needs to be known in order to act? How can firms effectively acquire, store, update, and use knowledge?	A collection of knowledge assets and a means of extending individual learning capabilities and knowledge	Create purpose, broker information, challenge status quo/hire and train knowledge workers	Respond to changing circum-stances with structures that promote learning and knowledge application	A field that can be manipulated and changed; a source of stimuli and feedback
Sensemaking	Interpretive	Deductive/ inductive	How do individuals and groups recognize and inter-pret stimuli? How do they generate what they interpret?	Net of activities	Understand interrup-tions in the ongoing flow of activity/ "Author" coherent interpretations and actions	Often not central to sensemaking	Source of physical stimuli. Social set-ting dominated by talk, symbols, promises, lies, agreements, threats, expectations, etc.
Change	Positive/ interpretive	Deductive/ inductive	When, why, how much, and in what direction will firms Change strategy?	A socio-political set-ting that generates outcomes satisficing diverse stakeholders	Establish "theories of action" to guide be-havior in new situa-tions/both conserva-tive and excitable	Adapt to maintain satisficing routines; invent new routines if necessary	A setting of varia--tion and structure influencing but enacted by human activity

Source: Huff, Huff & Barr (2000, pp. 20-21)

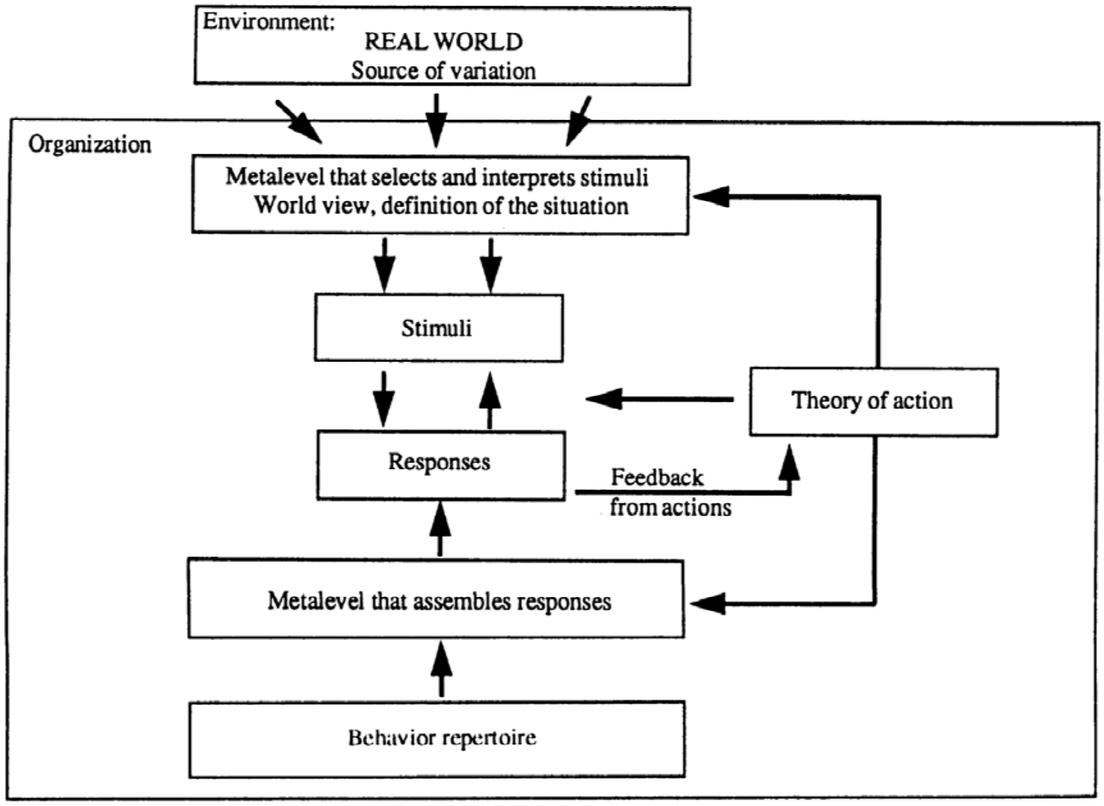


Figure 2.4: An S-R Model of How an Organisation Interacts With Its Environment
Source: Weick (1995, p. 122).

Langley et al. (1995) provide sets of descriptors and metaphors for researching decision-making. The dissertation assumes that decision-making with developing and delivering Internet-based software is “insightful” (Langley et al. 1995, p. 269); the decision-makers are “creators” (Langley et al. 1995, pp. 267-8); and the type of decision-making research is to “re-analyse previously analysed decision processes not just new ones” (Langley et al. 1995, p. 277).

Given the rapidly changing context of Internet-based software houses (see more discussion later in the chapter on software development), there was a need to track at least some aspects of sensemaking for decision-making over time – plus to incorporate some cognitive perspectives on "Structuration". Huff, Huff, Barr & Stimpert (2000) suggest using Structuration theory to extend a cognitive theory of a firm and strategic change for the firm. They restate Gidden's (1984) view that "Structuration theory asserts that individuals are active agents with the capacity to transform their setting through action". Figure 2.5 is a schematic depiction of Structuration theory.

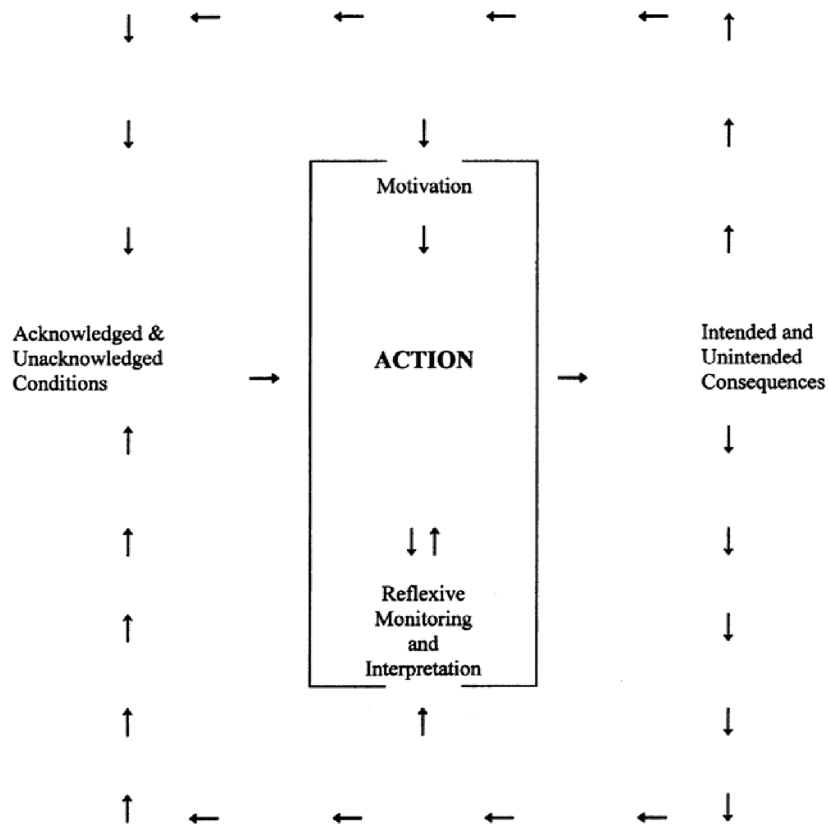


Figure 2.5: Context of Knowledgeable Action
Source: Huff, Huff, Barr & Stimpert (2000, p. 207).

Although Huff, Barr & Stimpert (2000) discuss a comprehensive multilevel Structuration model for strategic change with commentary at each level (industry, firm, group, individual), the research project is clearly focused on individual accounts of sensemaking over time. Nevertheless, accounts of individuals' sensemaking for decision-making over time do cover group and firm level activities – and especially as most software houses in the project will be still classified as “start-ups” and being less than ten years old.

A further aspect to individual sensemaking to be explored in the project is the substantial accounts of increasing decision-making with third parties. When viewed from a perspective of multi-firm interactions for software application development activity, there is also some focus at the firm and the software application industry level.

Huff, Huff, Barr & Stimpert (2000) offer ten propositions relating to cognitive perspectives on strategic change, mainly at an industry level, with some propositions going through to firm, group and individual level. Exhibit 2.1 offers the ten propositions.

Four propositions were dropped, as they were deemed to be beyond the scope of the dissertation. The revised set of Structuration propositions is presented in Exhibit 2.2.

Using the extended case method as outlined by Burawoy et al. (1991) and applied by Danneels (2002), accounts of individual sensemaking were collected, revisited and updated and cognitive issues developed for this dissertation..

Revised Structuration propositions from Huff, Huff, Barr & Stimpert (2000) are reviewed in the theory development section of the project (Chapter 10) against the subject software houses and collected accounts. Theory development includes new propositions. Cross-case analysis and theory development will be undertaken using a late-stage high-level running exchange approach as described by Danneels (2002, p. 1101) between case-study findings and the propositions.

Supporting the broad cognitive perspectives of exploring sensemaking associated with decision-making over time, an extended version of marketing decision analysis (DSA) is applied in the dissertation.

2.6 Marketing Decision Systems Analysis (DSA)

A well-established body of research on describing real-life marketing decision-making is available (Cyert, Simon & Trow 1956; Morgenroth 1964; Hulbert, Farley & Howard 1972; Howard, Hulbert & Farley 1975 and revisited by Hulbert in Hulbert 2003; Mintzberg, Rasinghani & Theoret 1976). Early DSA models were developed using computer flowcharting techniques.

Exhibit 2.1 An Anchored Cognitive Theory for Strategic Change – Structuration Propositions

---

Proposition 1: As knowledge actors within the industry draw on past success to respond to new conditions, they draw primarily on familiar rules and resources. Because firms with the industry have knowledge gained from past experience in the industry, even when they are entrepreneurial, knowledgeable incumbents tend to reconstitute industry structures that are quite similar to previously existing industry structures.

Proposition 2: The motivation for entry increases as outsiders perceive opportunities in actual or contemplated activities that are ignored or perceived as trivial by industry insiders.

Proposition 3: New entrants respond to system opportunity in novel ways by drawing on rules and resources that are unfamiliar to most industry participants.

Proposition 4: New activities by entrants, amplified by alliances with mavericks within the industry and/or actors not previously involved in the industry, modify the underlying structure (rules and available resources) of the industry.

Proposition 5: Active monitoring and interpretation by entrants tends to quickly increase their relative knowledgeability and power. Strong niche performance motivates successful new entrants to expand their activities and invites further entry by imitators.

Proposition 6: Shared schema-based interpretations significantly decrease the likelihood that incumbent firms will quickly notice opportunities demonstrated by new entrants or clearly understand the threat of entrant activity.

Proposition 7: Structures of control and regulation limit the incumbents' power to respond to the circumstances that attract new entrants or to respond to subsequent opportunities associated with entry itself.

Proposition 8: Withdrawal to apparently "safer" or "stronger" strategies is often unsuccessful because new actors and activities are reconstituting industry structure, decreasing the utility of past knowledge and increasing opportunities for entrepreneurial activity.

Proposition 9: Over time, new entrants "train" an increasing number of actors within the industry to notice and understand their insights and behaviours, and therefore becomes easier for incumbents to monitor and replicate the activities of new entrants.

Proposition 10: Efforts to institutionalise their success begin in structure entrant activities, reduce their capacity to respond to subsequent changes in the industry, and thereby increase the opportunities for new entrepreneurial activity by others.

The propositions reproduced above were extracted from Huff, Huff, Barr & Simpert (2000, pp. 212-9).

Exhibit 2.2 An Anchored Cognitive Theory for Strategic Change – Revised Structuration Propositions

---

Proposition 1: As knowledge actors within the industry draw on past success to respond to new conditions, they draw primarily on familiar rules and resources. Because firms with the industry have knowledge gained from past experience in the industry, even when they are entrepreneurial, knowledgeable incumbents tend to reconstitute industry structures that are quite similar to previously existing industry structures.

Proposition 2: The motivation for entry increases as outsiders perceive opportunities in actual or contemplated activities that are ignored or perceived as trivial by industry insiders.

Proposition 3: New entrants respond to system opportunity in novel ways by drawing on rules and resources that are unfamiliar to most industry participants.

Proposition 4: New activities by entrants, amplified by alliances with mavericks within the industry and/or actors not previously involved in the industry, modify the underlying structure (rules and available resources) of the industry.

Proposition 5: Active monitoring and interpretation by entrants tends to quickly increase their relative knowledgeability and power. Strong niche performance motivates successful new entrants to expand their activities and invites further entry by imitators.

Proposition 6: Efforts to institutionalise their success begin in structure entrant activities, reduce their capacity to respond to subsequent changes in the industry, and thereby increase the opportunities for new entrepreneurial activity by others.

The propositions reproduced above were extracted from Huff, Huff, Barr & Simpert (2000, pp. 212-9).

A developing body of decision-making research is available for business marketing, particularly for manufactured goods (Capon & Hulbert 1975; Woodside & Samuel 1981; Vyas & Woodside 1984; Stubbart & Ramaprasad 1988; Woodside 1994; Woodside & Wilson 2000). Figure 2.6 shows a DSA model from an industrial marketing context developed by Capon & Hulbert (1975). Figure 2.7, developed by Woodside & Wilson (2000), covers DSA for buying solvents, also in an industrial marketing context.

Most of the DSA research in the context of business-to-business or industrial marketing contexts is used in developing "thick" descriptions of a buying or product development process. The DSA models provide clarity and further insights and, as such, represent powerful supporting analytical tools for the development of more rich thick descriptions (for discussion on "thick" descriptions, see Geertz (1973).

Very little DSA-based research is available with respect to marketing decision-making for software application development. This dissertation extends DSA research for marketing decision-making into software application development, with particular emphasis on recent Internet-based software applications.

2.7 Extending DSA Research with Event Chronology and Cognitive Maps

Most of the case studies noted in the previous section provide varying levels of description of a software process that is able to be replicated for new versions of an application or new applications. However, while such software processes can be discerned from the cases, almost all cases have a detailed chronology of the genesis and evolution of an application, intertwined with a history of the software house, plus description of one or more focal issues related to the development and delivery of the application.

Therefore, extending DSA models with events chronology maps, and translating insights and nuances from the decision-makers into mental or cognitive maps (see Senge 1990, Weick 1990, Huff 1990a) is desirable. While the events chronology map identifies key events that could be used as the starting point for an event history analysis, this particular methodology is not within the scope of this project. Rather, an events chronology map assists in both further refining the DSA model for relevant timeframes and issues, plus it is a vital input for the development of cognitive mental maps.

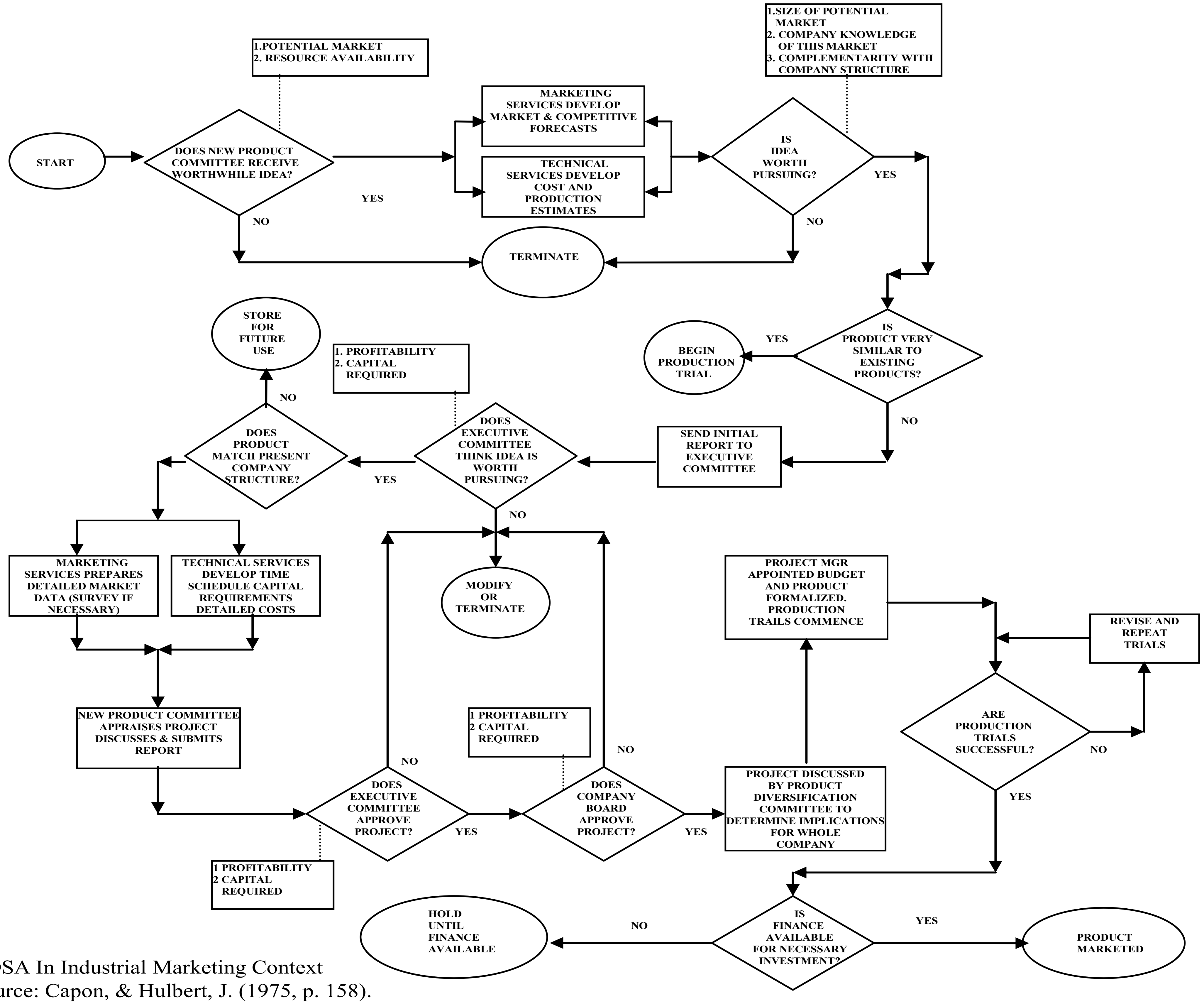


Figure 2.6: DSA In Industrial Marketing Context
Source: Capon, & Hulbert, J. (1975, p. 158).

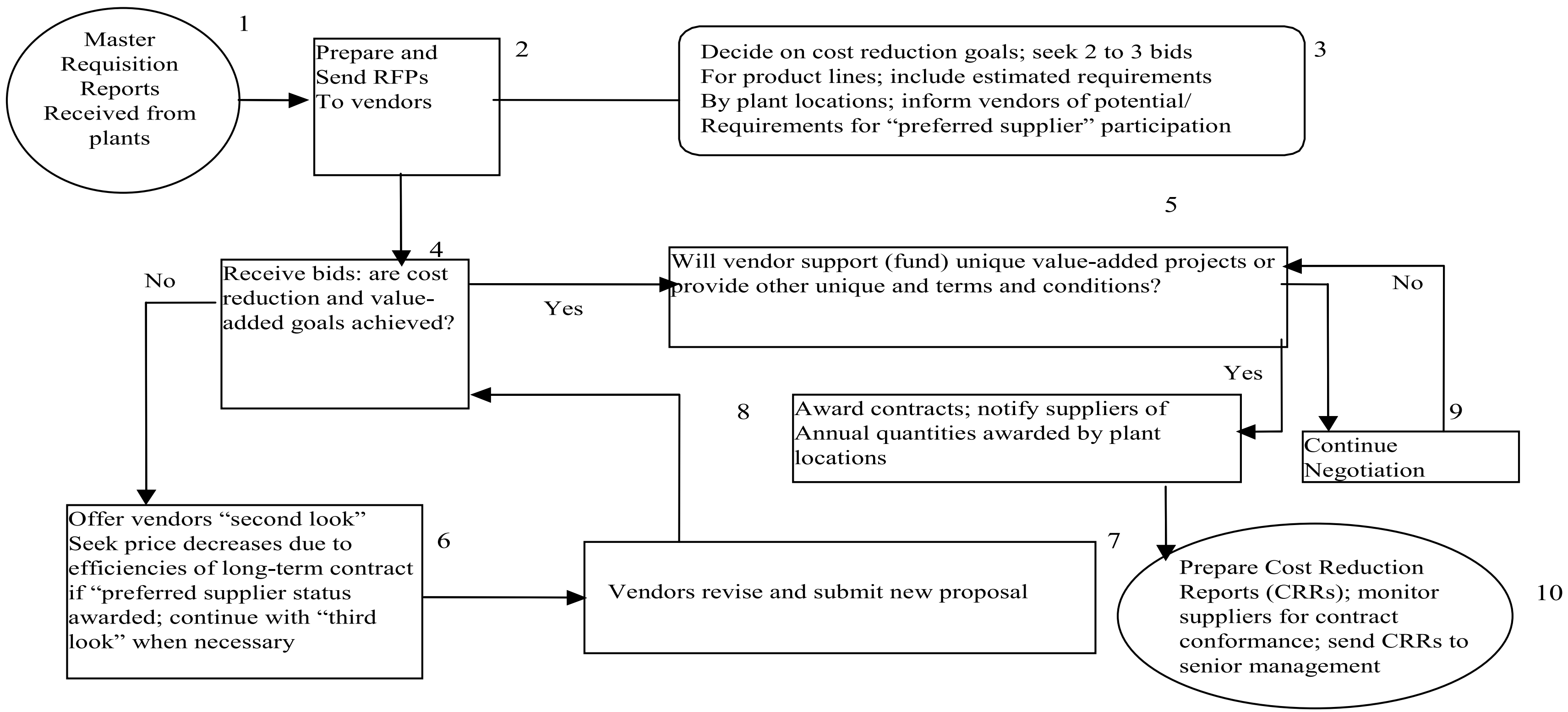


Figure 2.7: DSA In Business-To-Business Markets: Decision Systems Analysis for Buying Solvents at Smith Metal Works Finishing Plant
Source: Woodside & Wilson, E. (2000, p. 360).

2.8 Methods for Cognitive Mapping of Strategic Thought

An exploration of actual philosophies and methodologies for mapping strategic thought was undertaken for the dissertation. Huff (1990b) and Weick (1990) emphasise that strategic value might be obtained from taking a cartographic approach to framing strategic thought. They also describe various techniques for developing mental maps depicting these strategic thoughts. Huff, in particular, has provided considerable discussion of issues associated with developing mental maps for business contexts.

Several methodologies are available for the development of cognitive mental maps. Selection took some time, with exploration of and experimentation with five mapping methodologies – with each methodology capable of encapsulating insights and nuances associated with software development and delivery.

Storytelling methodology, as Schank (1998) describes it, was evaluated and rated as appropriate for identifying and indexing insights and key thoughts associated with decision-making. Storytelling research complements previous well-known case studies covering the development of high-technology products in the form of a story – for example Kidder's (1981) Pulitzer Prize winning novel, *The Soul of a New Machine*.

However, significant challenges occur with using output for coding and developing cognitive mental maps. A Venn diagram mapping technique was used to experiment with mapping indexed outputs, and while this approach provided useful insights into some issues, it was difficult to develop consistent codification and map design for multiple mapping.

Influence diagrams based on Clemen & Reilly's (2001) methodology were developed, covering a hypothetical and broad scenario of application development. This approach rated highly for being able to develop insights into links of influence between various decision-makers and events.

Software was available to model various links in the influence diagrams in more quantitative detail. However, the approach tended to produce static maps and did not necessarily unearth enough detail on some of the key issues confronting the relevant decision-makers.

Strategic argument mapping (SAM), as Fletcher & Huff (1990a and 1990b) outline it, includes a coding and mapping process enabling the detailed breakdown of statements into claims, grounds, qualifiers, warrants, elaborations, subclaims, and reiterations.

The SAM approach provides detailed insights into very specific issues, such as the justification for a specific development decision or release date, for instance. However, it was not dynamic and very difficult to map for multiple timeframes and issues.

Causal mapping, described by Huff, Narapeddy & Fletcher (1990), offers a relatively simple but thorough approach to analysing and coding statements or passages or sections of an account. Moreover, there was at least one good example of the effective development of a set of cognitive maps in a business context.

Nath & Newell's (1998) case study of Pepsi Canada's strategic success in a hypercompetitive environment provided a detailed discussion on coding and developing cognitive maps, using a cut-down version of Huff, Narapeddy & Fletcher's (1990) causal coding taxonomy. The cognitive maps provide detailed insights into the strategic thinking of executives at Pepsi Canada for three different time periods (1990, 1993 and 1994) and two of these maps (1993 and 1994) are presented in Figures 2.8 and 2.9.

Ambrosini & Bowman (2002) report on cognitive mapping organisational routines; James (2000) reports on mapping industry-level relationships and strategic alliances; Rughase (2002) reports on customer cognitive mapping; Eden & Ackermann (2002) map organisational strategy. For mapping individual insights related to specific strategic issues, see Hodgkinson & Maule (2002).

The causal mapping (cognitive mapping) approach was chosen ahead of the other four evaluated methodologies. In retrospect, this decision not only enabled the development of clear but detailed and consistent maps, but it also opened up a great opportunity to use new qualitative research software to electronically deconstruct accounts into words or phrases or sentences or short passages, then to dynamically model causal links through several iterations to produce powerful cognitive maps.

All the cognitive map examples reported in Huff & Jenkins (2002) include discussion on evolving the mapping of strategic thought toward the mapping of strategic knowledge, with a very strong leaning toward using cognitive mapping techniques. Huff & Jenkins (2002), in their introduction, highlight the use of software applications to "elicit primary information" from workshop discussions on organisational routines (Huff & Jenkins 2002, 7: Ambrosini & Bowman 2002) and for developing customer maps (Huff & Jenkins 2002, 7; Rughase 2002).

Cognitive Map: Pepsi Canada Strategic Plan 1993

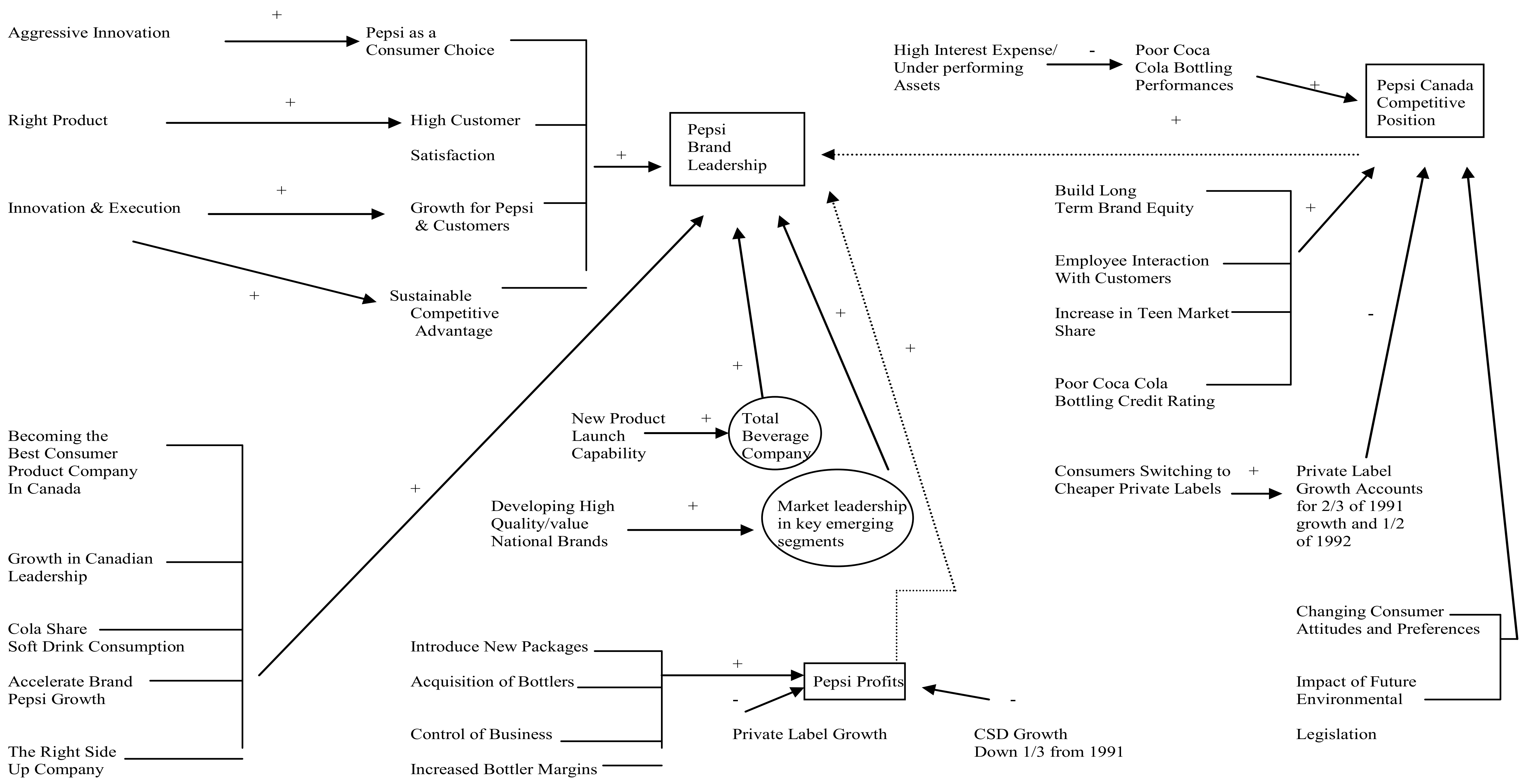


Figure 2.8: Cognitive Map: Pepsi Canada Strategic Plan 1993.
Source: Nath & Newell (1998, p.46).

Cognitive Map: Pepsi Strategic Plan 1994

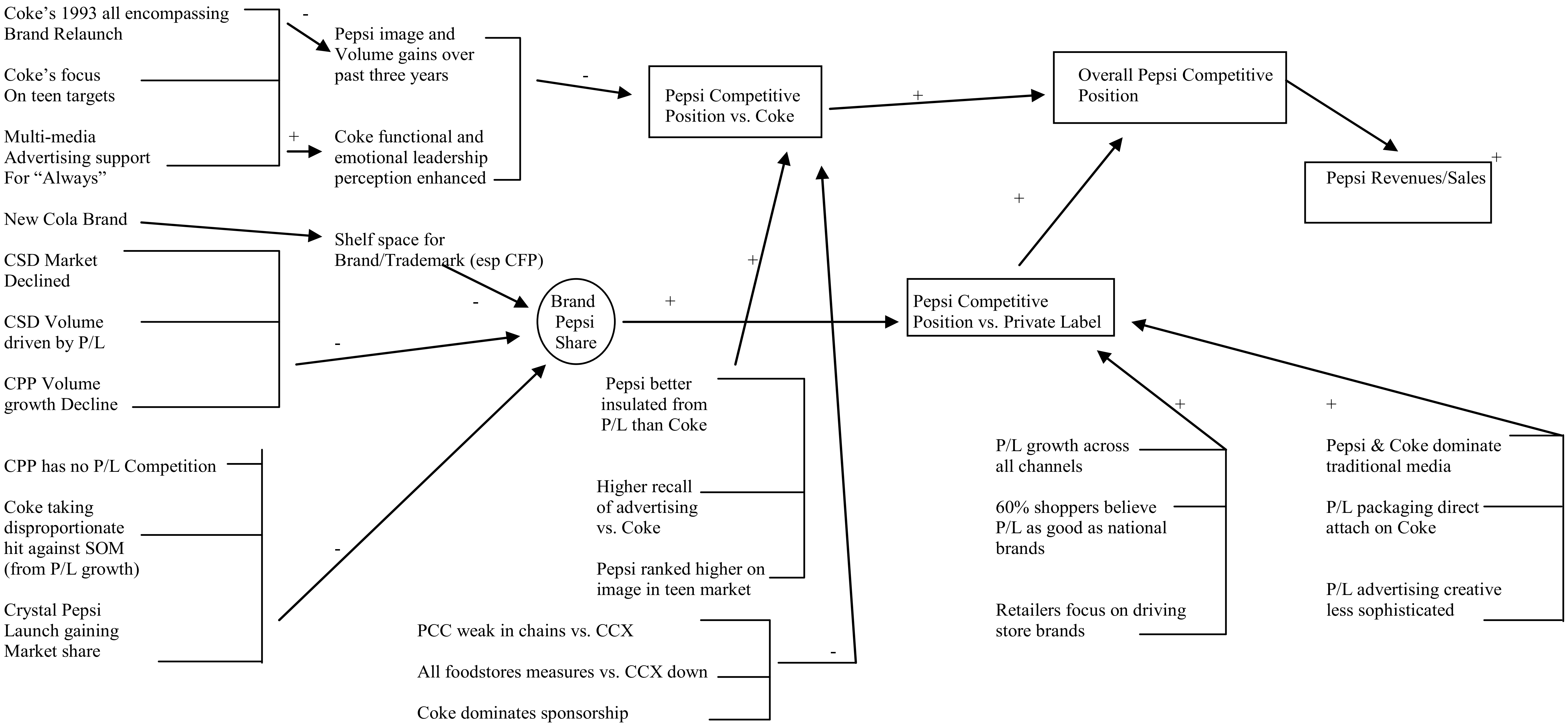


Figure 2.9: Cognitive Map: Pepsi Canada Strategic Plan 1994.
Source: Nath & Newell (1998, p.47).

Jasinski & Huff (2002) also describe a knowledge-based system for mapping strategic options that provide similar mapping outputs to the software applications proposed for the research project. There is further discussion on relevant software applications in Chapter 3, on Research Methodology.

When the research project was conceived in 2000-01, there was no reported research literature on using software applications for cognitively mapping cognitive strategy.

However, that situation has changed during the life of the project and, while it is now not likely that this project represents the first contribution on the cognitive mapping of aspects of strategy, it will be one of the first projects to report to the cognitive mapping for marketing decision-making associated with the development and delivery of software applications in new business-to-business environments. The actual final formatting of the cognitive maps in the research project is a new and distinct contribution in terms of schematic representation for cognitive maps.

Any number of cognitive maps could be chosen to support the DSA models and event history maps. Nath & Newell (1998) developed three cognitive maps covering perspectives for three different time periods. For the research project, initial cognitive mapping was set at two maps focusing on strategic issues captured in different timeframes, and one map addressing in more detail a key decision-making issue associated from the original sensemaking account of developing and delivering the Internet-based application.

2.9 Extension Issues – a Platform for Further Exploration of Decision-making Concepts and Issues

The combination of models and maps described thus far provides an analytical platform for adequately describing and cognitively mapping decision-making processes associated with the development and delivery of Internet-based software. This platform could be extended to research a wide range of concepts and issues.

For the research project, understanding the development and delivery of Internet-based software applications as a knowledge-creation system could lead to the exploration of several extension issues. Any number of issues could have been canvassed. However, the following set of issues was regarded as significant for the decision-making associated with the development of Internet-based software applications:

2.9.1 Review of Dominant Logic.

2.9.2 Development of Shared Vision/Experiences.

2.9.3 Identification of Key Changes or "Leverage Points".

2.9.4 Consideration of Relevant Strategic Marketing Issues as a Form of Marketing Knowledge.

2.9.1 Review of Dominant Logic.

Evidence exists of managers dominating decision-making with their mental models or logic on how products should be developed (see Prahalad & Bettis 1986; Bettis & Prahalad 1995) in manufacturing and B2B cases. Several Internet-based case studies point (at least informally) toward an individual or a group of founding members with dominant logic in terms of the types of applications and how they should be developed.

Based on a cursory reading of selected case studies and literature on Internet-based software houses, some examples of apparent dominant logic may include founders of Zaplet in terms of application use (DeLacey & Leonard 2001); the founder of NetDynamics in terms of the platforms chosen for application development (Iansiti & MacCormack 1996); and the founder of Trilogy for the type and standards of software development (Austin 1998). Extension research on dominant logic may identify those individuals or groups who establish and disseminate it through an organisation and its partners.

2.9.2 Development of Shared Vision/Experiences.

Previous DSA research in manufacturing/B2B contexts has highlighted the sharing of knowledge within and across organisations (Weick 1990; Woodside 1994; Woodside & Voss 1999). The vision for the development of products has been located within production, sales and organisations in manufacturing. Some researchers have suggested these groups working together as a knowledge creation system (for example, Dougherty 1992). Such vision may be formulated with third parties, such as suppliers of key components, subassemblies and manufacturing consumables (see Biemans 1991; Biemans 2003).

Extension of shared vision, not only throughout a software development house but also to relevant third parties such as independent software vendors (ISVs), system integrators and other developers, is explored through extension questions in the dissertation.

2.9.3 Identification of Key Changes or "Leverage Points".

There are events or points in decision-making processes that either lead to or signify a new or emergent state. This state could be the conversion of an idea to a product, conversion of prototype to product, a major change in the business or development focus, or a change in the logic or vision of the founders of the company.

Klein's (1999) definition of "leverage points" focuses on "a small change that can turn a situation around" but he provides examples of actions that led to large changes, some of which could be seen as emergent states. A variation on the leverage point is Gladwell's (2002) "tipping point" which attempts to explore actions that led to large changes, except that it focuses on the collective convergence of a range of actions, rather than attempting to track single actions and their possible emergent consequences. For the research project, key changes, rather than the specific small actions, are identified following Klein's (1999) "leverage point" perspective.

2.9.4 Consideration of Relevant Strategic Principles as a Form of Marketing Knowledge

Rossiter (2001) proposes four forms of marketing knowledge, that is, marketing concepts (building blocks of marketing knowledge), structural frameworks (non-causal models), strategic principles (hypothesised causal models that relate one concept to another in a functional "if do" form), and research principles (hypothesised causal models pertaining specifically to the appropriate use of particular research techniques). Rossiter (2001) classifies "cause maps," "argument maps," and "reason maps" as strategic principles.

Some clarification is required here, as these maps would have to be developed within a marketing perspective, context or focal idea to be considered as strategic principles that could be construed as a form of marketing knowledge. The cognitive maps developed in this project could in themselves be seen as strategic principles. Indeed, all the DSA models and event chronology maps and cognitive maps are classifiable as strategic principles and a form of marketing knowledge.

For the research project, decision-makers will be probed to gain their insights into the key strategic marketing issues associated with the development and delivery of Internet-based software applications. While these insights on their own do not add up to strategic principles, they may assist to further orient the cognitive maps toward strategic principles as a form of marketing knowledge.

The complete package of DSA models, event history maps, cognitive maps and extension research outputs should provide a comprehensive exploration of decision-making associated with the development and delivery of Internet-based software applications, positioned as strategic principles, that is, a form of marketing knowledge.

2.10 Summary – Theoretical Contributions

This chapter built up a structured analysis of the literature on a diverse range of knowledge areas commencing with rationale and inspiration for the project – that is, contrasting conventional product development with rapid and flexible software development where software is viewed as a medium of codified knowledge.

The review explored storytelling methods for analysing the decision-making associated with rapid software application development, of which hermeneutic analysis was chosen as the preferred methodology.

Supporting the actual storytelling method, the review analysed theories addressing the expression of sensemaking into effective outputs and further analysis. Sensemaking theory and its association with developing cognitive theories of the firms were reviewed, and propositions on the Structuration of firms were presented as propositions to be evaluated during the cross-case analysis stage of the research project.

Decision systems analysis (DSA) was nominated as a framework for evaluating and mapping marketing decision-making processes, because of the relatively strong body of marketing research using DSA. However, there was considerable scope to expand existing DSA to incorporate more sensemaking information and to map that information.

The specific expression of sensemaking of decisions and decision-making through mapping, with a view to adding it to existing DSA, was evaluated, with a variety of mapping techniques being subjected to rigorous assessment for their effectiveness to schematically represent decision-making.

An advanced hermeneutic framework using an extended of marketing decision systems analysis including cognitive mapping was developed out of the literature review. The framework was also set up as a form to explore issues deemed significant to the development of Internet-based software. These issues included: dominant logic, shared vision, leverage points, and relevant strategic marketing issues (as viewed by the decision-makers themselves).

To sum up – in fact an advanced hermeneutic analysis "metasystem" – between the author of the research project and the literature for the project – transpired, with several levels of dynamic analysis and interpretation that resulted in proposing the following theoretical contributions for the project:

2.10.1 Development of Revised and New Theory Propositions Associated with Individual, Firm and Software Industry Level Structuration, as a Step toward Developing a Cognitive Theory for a Firm Researching, Developing and Commercialising (RDC) Software Applications

2.10.2 Thick Descriptions and Analysis of Internet-based Software Application Development Classified as Either Disruptive Technology or Contributing toward Disruptive Technology

2.10.3 Exploration and Further Insights into New and Emerging Internet-based Application and Knowledge Development

2.10.4 Development and Application of Decision Research, Based on Revisitation and Revision of Previously Accounted Marketing Decisions

2.10.5 An Advanced Hermeneutic Analysis Framework for Dynamic Interpretative Research on B2B Marketing Decision-making

2.10.6 An Extended Marketing Decision Systems Analysis Approach, Enabling Dynamic Sensemaking of B2B Marketing Decision-Making

2.10.7 Extension of Cognitive Mapping Concepts to an Extended Dynamic Cognitive Mapping Schema

A comprehensive extended case-study research methodology was developed to support the theoretical analysis in the project, and is outlined in the next chapter.

Chapter 3: Research Methodology

3.1 Introduction

The research project used the extended case method (Burawoy et al. 1991), drawing upon initial accounts of decision-making associated with application development from a selection of existing Harvard Business School Case-studies on Internet-based software companies.

The accounts were analysed and mapped, with the outputs reviewed with individuals involved in the software development process described in the original case-studies. During the review process, the individuals validated or revised their beliefs as to what decisions were made and implemented, events occurring in the process, and rationales used to justify actions at the time of the original time period. The outputs were updated accordingly.

Steps in the project included the development of a set of DSA models, event history, and cognitive maps based on text analysis of the selected cases, followed by review of the models and maps through interviews. The complete extended DSA package was then incorporated into a multilevel advanced hermeneutic analysis framework and represented as an advanced hermeneutic “system”.

Rigorous cross-case study analysis was undertaken, using new semantic web-based analytical tools, producing a composite DSA Model, and several cross-case findings. Both individual case study and cross-case findings were then further analysed and incorporated into revised and new theory propositions.

The Research Methodology chapter structure is presented in Figure 3.1. Figure 3.2 outlines the overall research project procedure.

3.2 Research Project Methodology Development

Initially, the research strategy proposed for the project was to examine about five or six Australian-based Internet software applications as separate cases through producing a written account of their application development and delivery processes. These written accounts were to be analysed and developed into a set of decision-making maps, which would be further validated by the relevant decision-makers.

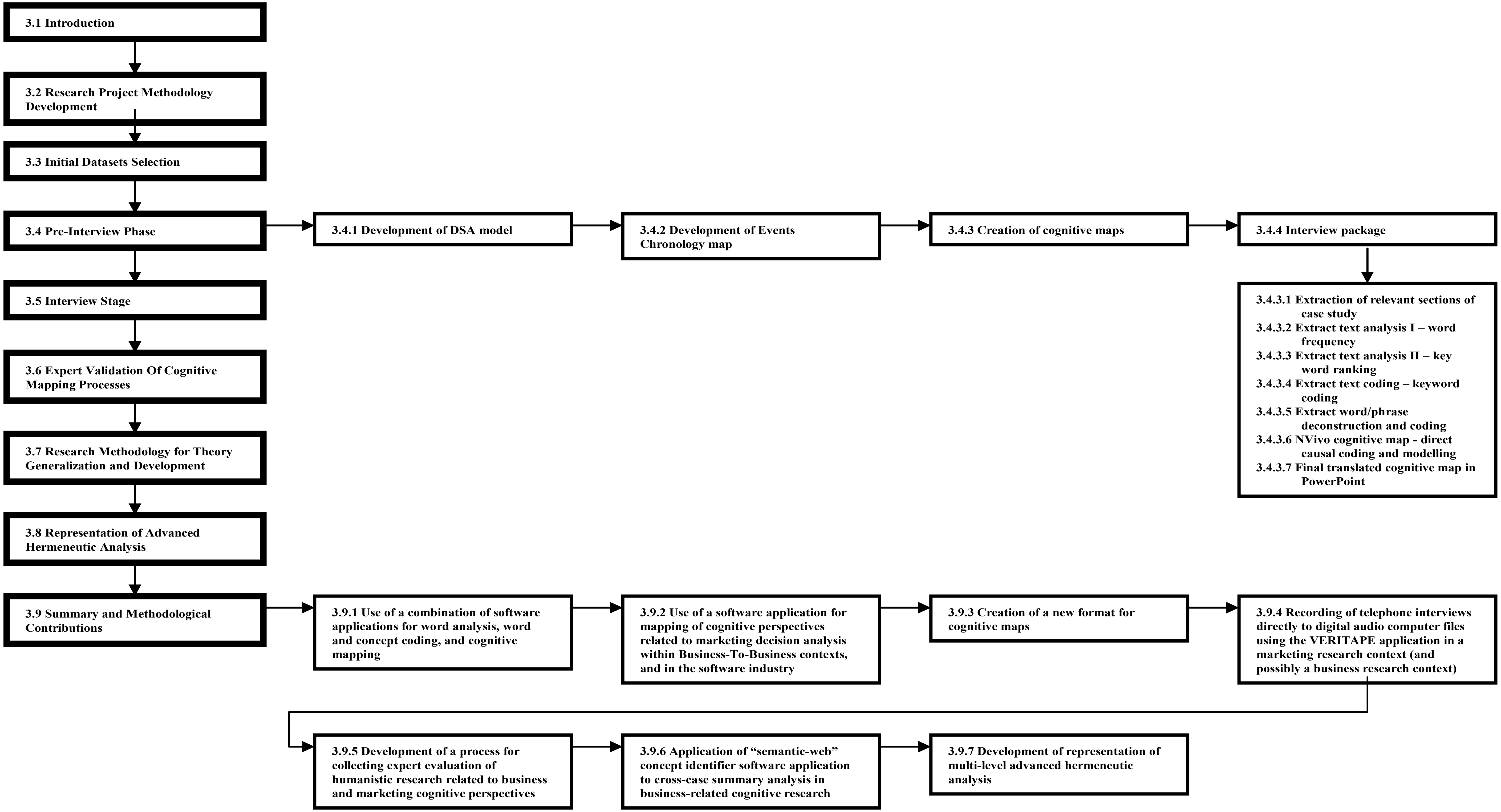


Figure 3.1 Research Methodology – Chapter Structure

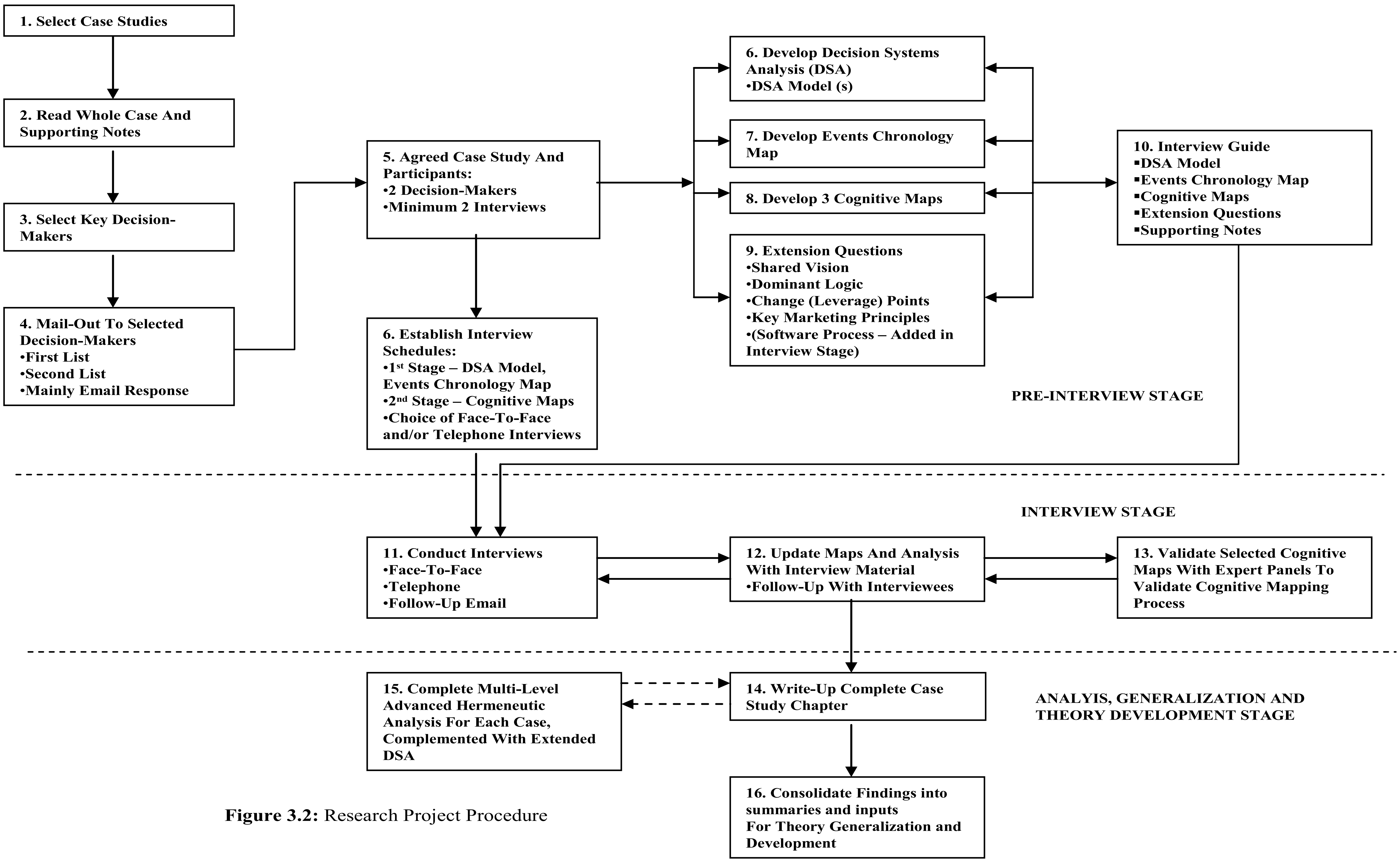


Figure 3.2: Research Project Procedure

However, the vast majority of major Internet-based software application developers during the period 1994-2001 were located in the United States, and a decision was reached to research a subset of these developers rather than focusing on smaller developers in Australia.

Research on the development of businesses both contributing to the development of the Internet and to businesses using the Internet from a business management perspective has grown rapidly since 1997. A significant subset of this area is research on Internet-based application developers.

A review late in 2001 of over 300 Harvard Business School (HBS) case-studies on software developers revealed a rapid increase in the number of Internet-based software case-studies from 1997 through to 2001. By the end of 2001 over 100 Harvard Business School case-studies covered several aspects of Internet-based software development.

3.3 Initial Datasets Selection

Further evaluation of the Harvard Business School Case-studies related to Internet-based application software development indicated that about fifty cases contained sufficient content to develop thick descriptions of decision-making associated with software application development and delivery.

However, enhanced marketing decision-making analysis description could be developed with new supporting DSA models, event history maps and cognitive maps, through further research with participating decision-makers. Furthermore, research with participating decision-makers extends the project beyond a direct review of the case study itself and the quality of the author's account of software development processes.

Initial case study selection was based on the following criteria:

- Software has to be "beta"-tested and commercially introduced
- Case-studies were written by well-known authors
- The application development process studied must be over a timeframe of more than 12 months
- Names of case study companies and participants can be disguised if requested
- Applications must be Internet-based

Harvard Business School (HBS) Case-studies (or equivalent as some cases available through HBS were actually undertaken by Stanford University or the Ivey Business School in Canada) on Internet-based software applications that met the criteria were allocated into a 2x2 project case study selection matrix based on business performance (success/failure), and size (small/large). A target was set of two cases to be selected from each quadrant, giving eight case-studies for the project.

Allocation of case-studies by either “success or failure” and by size was more difficult than anticipated. Size was defined by the annual revenues at the time of the case study publication, as indicated by HBS in the website entry for a case study under consideration for the project. Large was defined as having annual revenues of over $US1 billion per annum.

Through 1999 to 2001, several application developers that were case study subjects were either acquired by larger companies (for example, Firefly Networks acquired by Microsoft; Insite Marketing, acquired by Silknet, which was in turn acquired by Kana Communications, which recently merged with Broadbase).

Several other developers had experienced dramatic falls in revenues in line with the “dot.com” crash and technology stock corrections of this period. Allocation into the Project Case study Selection Matrix was retained, based on original HBS definitions for annual revenues at the time of Case study publication.

Three case study project matrices were created in August, September and November 2001, with the final list reduced to 23 cases allocated thus:

- Success/small – 8 cases
- Failure/small – 6 cases
- Success/large – 7 cases
- Failure/large – 2 cases

(See Table 3.1).

Table 3.1: Case Selection Matrix

SUCCESS/SIZE	**SMALL**	**LARGE**
YES	**First List:** •Zaplet Inc. **Second List:** •Trilogy (evolved to MEDIUM) •Kana Communications •Verge Software •Alibris •Motive Communications •WebSpective Software •First Virtual **TOTAL: 8**	**First List:** •Red Hat/LINUX •YAHOO! **Second List:** •SUN Microsystems •Adobe Systems Inc. •Intuit QuickBooks •SIEBEL Systems •i2 Technologies Inc. **TOTAL: 7**
NO	Second List: •VerticalNet •Ventro •Insite Marketing •NetDynamics •Extricity Inc. •Ninth House TOTAL: 6	**First List:** •Netscape Communications **Second List:** •Inktomi **TOTAL: 2**

All 23 case-studies were reviewed and key decision-makers were identified for each case. A total of 93 decision-makers were identified, ranging from two to six per case study. An extensive Web-search was undertaken to locate all decision-makers, as several had moved on to other ventures and organisations. Two contact lists were developed from the Web-search.

The first contact list contained 17 decision-makers from four case study application development organisations, with a target of reaching sufficient respondents to complete two case-studies for the project. A mail-out based on the first contact list with telephone or e-mail response inviting participation in the project was initiated in November 2001.

For the first list (and actually also for the second list) e-mail response and dialogue was the fastest and most effective communication means for finalising involvement in the project. Two case-studies were engaged from the first list with telephone interviews commencing in December 2001.

The second contact list contained 54 decision-makers from 19 case study application development organisations, with a target of reaching four to five case-studies (and a total of six case-studies for the project).

The mail-out for the second list, with increased emphasis on e-mail response was initiated in December 2001. Four case-studies were organised from the second list, with two more organisations classified as "reserves".

The final set of companies completed in the research project is presented in Table 3.2. The table was extended to incorporate comment on whether the applications in the project were examples of disruptive technology.

The assessment was added because Christensen & Raynor (2003) extended Christensen's (1997) work on disruptive technologies to classifying over 75 products and companies as "disruptive technologies". At least four applications within the final project case study selection covered applications that were either directly "disruptive" (Linux – and open-source, and QuickBooks being a new means for small businesses to manage their financial affairs), or were key components of disruptive technologies (Zaplet and Kana as key components of e-mail).

While NetDynamics and Trilogy were not in Christensen's (2003) disruptive technologies list, they should be, as both companies and their applications strongly met Christensen's (2003) criteria for disruptive technology. Trilogy can be classified as a direct disruptive technology because its configuration software revolutionised pricing and quoting on computer systems and software, plus it assisted in slashing general sales and marketing costs for enterprises. NetDynamics could be classified as both direct and indirect disruptive technology. NetDynamics was a key component of browser technology which, with Sun Microsystems' Java and Netscape Communications Navigator browser, could be seen as a disruptive technology.

Furthermore, NetDynamics' application became part of Sun's iPlanet enterprise application suite, which is now part of the Sun One Web Services environment – a very likely candidate for disruptive technology in terms of current and future application development and roll-out.

The final list of cases for the research project represents new and disruptive software applications, mainly from new and emerging software houses.

Table 3.2: Final Company Case Study List

Software House	Case Selection Criteria	Disruptive Technology Assessments based on Christensen (2003, 56-65)	Harvard Business School Case
Red Hat Inc.	Size: Large Success: Yes	Yes, Linux	"Red Hat And The Linux Revolution", MacCormack & Herman (2000)
Zaplet Inc.	Size: Small Success: Yes	Yes, direct enhancements to email	"Zaplet Inc. (A)", DeLacey & Leonard (2001)
NetDynamics Inc.	Size: Small Success: Became Part of SUN's iPlanet Group	No, linked to Java, Web Browsers and Ultimately Web Services	"Living On Internet Time: Product Development at Netscape, Yahoo!, Net Dynamics and Microsoft", Iansiti & MacCormack (1996)
Kana Communications Inc.	Size: Small Success: Yes (through at least two substantial mergers With Silknet and Broadbase)	Yes, related to email and online communication management	"Kana Communications", Sarvary (2000)
Intuit QuickBooks Inc.	Size: Large Success: Yes	Yes, Quickbooks	"Intuit QuickBooks", Kochikar & Lal (2000)
Trilogy Inc.	Size: Small/Medium Success: Yes	No, Application Cut Sales & Marketing Costs and revolutionized quoting systems	"Trilogy (A)", Austin (1998); "Trilogy (B)", Mandel & Austin (2000)

3.4 Pre-Interview Stage

A significant circular challenge emerged in the Project, whereby the development of a full package including a DSA model, an events chronology map and three cognitive maps could not be commenced until there was sufficient response from an organisation to proceed.

Development of a full package for each case ranged from two to four weeks. However, as will be outlined later in this chapter, in practice, the package was split such that multiple interviews were conducted with the DSA model and events chronology mapcovered in the first interview, and the cognitive maps covered in the second interview.

The pre-interview sequence for each selected case study was:

3.4.1 Development of a DSA Model

3.4.2 Development of an Events Chronology Map

3.4.3 Creation of Cognitive Maps

3.4.4 Interview Package.

3.4.1 Development of a DSA Model

The main objective of the DSA model is to visually depict decision-making processes associated with the software process for development and delivery of an application. The model should reflect a set of processes that were not “one-off” and specific to an application, but could be replicated for either new applications or new versions of an existing application.

After a case study selection was finalised for interviewing, the relevant case study and supporting notes were thoroughly analysed and a flowchart DSA model was developed.

The DSA model uses standard flowchart conventions and symbols similar to, for example, Hulbert, Farley & Howard (1972); and Howard, Hulbert & Farley (1975). The DSA models were developed as complete flowcharts with minimal or no supporting notes.

Identification of the key decision-makers and groups associated with the software process was critical. One major advantage of researching Internet-based application developers was that all applications were developed in the 1990s and that either at least one company founder or original application developer for each case study participated in the project.

Overall, the young age of the selected organisations meant that at least for the early versions of an application, the relevant groups and decision-makers could be readily identified. Understanding where and when their decisions impacted on the software application development process often required judgement beyond the written case study.

Every effort was made to depict the development process as a cycle with the use of feedback loops in the DSA models.

3.4.2 Development of an Events Chronology Map

All selected case study documents were written with application development processes and case issues cast in chronological order. However, determining commencement and end dates for an events chronology mapwas a significant challenge.

For some cases the decision-making processes associated with application development were satisfactorily encapsulated within the written case study timeframe; for some cases the interviewees had moved on and preferred to focus on either the timeframe of the written case study or a portion of it.

For some case-studies, important events in the year or so after the written account (usually in 2001 and 2002) were of sufficient gravity to be incorporated into the events chronology map(and for some cases to be added in the form of new cognitive maps and DSA models). However, these extensions were only considered emerging out of the interview stage of the project.

A simple sequential process chart approach was used for development of event history maps, and most of the conventions and symbols for the events chronology mapwere based on Woodside & Wilson (2000), with the addition of two symbols related to decision-making challenges, and solutions to those challenges.

3.4.3 Creation of Cognitive Maps

The DSA model and events chronology mapfor each case study was further analysed to determine three distinctive themes for cognitive mapping.

Typically the three themes were:

- Snapshot of case study organisation at an early timeframe in the event history map
- Snapshot of case study organisation at a late timeframe in the event history map
- Perspectives of a key or focal issue related to the application development process

This list of themes was not exhaustive, as in all case-studies there were other issues and timeframes that could have qualified for cognitive mapping.

Creation of the three cognitive maps for each case study was in itself a substantial project. For each map, selected extracts from the case study were subjected to rigorous text analysis, text coding, causal linking, and modelling into the final map form.

As noted in the Literature Review chapter, there have been substantial developments in software applications for cognitive mapping since the research project was conceived in 2000-01.

At the time of the commencement of the research, a rigorous search was undertaken for software applications that would provide ranked word frequencies, capabilities to code words and concepts, and enablement of dynamic mapping concepts and associations into cognitive maps.

The combination of applications selected for the project, TACT and NVivo, met these requirements – although there was room for a lot more automation and further analytical tools beyond this combination.

Huff & Jenkins (2002) and contributors to "Mapping Strategic Thought" provide plenty of discussion and examples of approaches and software applications for developing cognitive maps.

Ambrosini & Bowman (2002) and Eden & Ackerman (2002) used Decision Explorer to display and store data. For more information on Decision Explore from Banxia Software see Banxia Software (2002). The Model Explorer Module in NVivo is in fact part of the Decision Explorer application licensed to QSR International for the NVivo application.

Rughase (2002) used SENSOR – an application that can store stories from in-depth interviews and produce cognitive maps. The application appears to produce effective maps, although the maps are generated by the application rather than created by the user.

Jasinski & Huff (2002) described the ATLAS/ti Software application as a knowledge-based system for coding data and producing maps. ATLAS/ti is not a new application but using it for the mapping of cognitive perspectives represents a new area of use for it.

ATLAS/ti can produce maps similar to NVivo/Decision Explorer (although not with as much dynamic rendering capability) and it does offer more freedom for coding links and associations than SENSOR and other proprietary or "home-grown" applications.

There is still a lot of potential for the development of new and more advanced applications for coding, storing and mapping cognitive perspectives, but for the present, Decision Explorer (and its incorporation into NVivo) is a well-justified choice.

Application of TACT/NVivo to support reporting and cognitive mapping for the marketing decision-making associated with the development and delivery of software applications in new business-to-business environments, still represents a new methodological contribution in the area of marketing, business-to-business activities, and for the software industry.

The seven analytical steps to produce a cognitive map were:

3.4.3.1 Extraction of Relevant Sections of Case study

3.4.3.2 Extract Text Analysis I – Word Frequency

3.4.3.3 Extract Text Analysis II – Keyword Ranking

3.4.3.4 Extract Text Coding – Keyword Coding

3.4.3.5 Extract Word/Phrase Deconstruction and Coding

3.4.3.6 NVivo Cognitive Map – Direct Causal Coding and Modelling

3.4.3.7 Final Translated Cognitive Map in PowerPoint.

3.4.3.1 Extraction of Relevant Sections of Case study

Each theme for cognitive mapping usually encompassed at least one subsection of a selected Harvard Business School case study, ranging from about ½ to 1½ pages. Initially, short subsections were selected, which led to the production of smaller but clear and detailed cognitive maps. Longer subsections in some case-studies encapsulated the key issues effectively but also resulted in very large and very detailed cognitive maps.

These maps could have been broken down further into micro-maps for further clarity, but in order to effectively manage interview times, larger holistic "all-in-one-page" cognitive maps were developed and discussed with interviewees.

The selected subsections were extracted and saved as TEXT files for input into Text Analysis applications.

3.4.3.2 Extract Text Analysis I – Word Frequency

The first analytical step was to compile a complete word frequency for an extract. While there are a plethora of applications that can return specific word searches, very few applications can produce a complete document word frequency.

Most of these applications are still running on PCs using MS-Dos or early versions of Windows (3.x and 95). Some of these packages have been or are currently being ported to Windows 98 or NT or 2000/XP operating systems in 2001/2 (see Klein 2003).

Comprehensive text analysis packages should include complete document Word frequencies in future releases and this recommendation has been forwarded to the developers of NVivo (see QSR International 2002a and 2002b) – the application chosen for most of the text analysis and mapping for this package.

During 2002 and 2003 there has been a significant increase in new applications for text analysis, coding and mapping. Some of the newer applications in this area, such as ResearchWare, Megaputer and Copernic are currently under evaluation and review by the author of the research project. These new packages are showing great promise for new projects related to storytelling methods, new cognitive perspectives on strategy, and for emergent marketing and sales strategies. For a list and description of a wide range of text analysis applications see Klein (2003).

The Textual Analysis Computing Tools (TACT) Package from the University of Toronto running under MS-Dos was utilised to create full Word Frequencies (see University of Toronto 1993 and TACT 1997). While TACT was running on old technology it was available as a free FTP file from the University of Toronto and was easy to use.

Text (.txt) Files of the selected extract were input into TACT, which created one file with a count for every word in the extract file.

3.4.3.3 Extract Text Analysis II – Key Word Ranking

As of late 2001 there was no single application that could take a full document word frequency and rank the output. The Word Frequency output files from TACT were thoroughly reviewed, and words with frequencies of three or more were marked, and then input to MS-Excel (see Microsoft Corporation 2002a and 2003a) for ranking. Common words such as “the”, “and”, “what” and “of” were not input for ranking.

Depending on the size of the extract the Ranked List ranged from about 25 to 60 words. Setting a threshold frequency of usually three or four words further filtered these lists. The final key word lists ranged from about 17 to 25 words.

3.4.3.4 Extract Text Coding – Key Word Coding

The NVivo software application from QSR International Pty Ltd (see QSR International 2002a & 2002b) is designed for qualitative analysis and contains a strong array of tools for analysing and coding document extracts, plus it has provision for the storage of derived analytical sets and modelling of selected aspects of these sets (see Bazeley & Richards 2000).

The same Extract text (.txt) files used in the TACT application were input to NVivo and converted to **Project Documents** within the NVivo environment. Microsoft Word Rich Text Files (.rtf) may also be input to NVivo.

Ranked Keyword Lists from MS-Excel for each extract from the TACT analysis were then keyed into the NVivo Search Tool. The Search Tool explores the relevant extract Project Document, returns the same Word frequency as in the TACT analysis, and automatically codes the word occurrences in the Project Document as a separate piece or "Node" which can be utilised for further analytical activities.

Occasionally, in order to score similar word frequency counts between TACT and NVivo, the search tool in NVivo had to be adjusted from a "regular" word search to an "approximate" word search. NVivo's Document Browser enables viewing of coded key words as a "coloured stripe" system to the right of the document text lines. The coded key words can serve as "anchors" or a starting point for more detailed Project Document word and section coding.

3.4.3.5 Extract Word/Phrase Deconstruction and Coding

The selected text extract can be viewed in NVivo with distinct colours for the words that have already been coded. From there, intersections of several frequent words and themes can be coded directly into further nodes directly from the extract.

The InVivo system within NVivo allows direct coding of strings, words, phrases or sentences from the Document Browser to nodes. The coding limit is 36 characters (although this limit can be exceeded through editing nodes when using the Model Explorer tool for mapping in the net step of cognitive map creation). This coding process is the equivalent of the cause mapping coding process as described by Huff, Narapeddy & Fletcher (1990), in which words or phrases are written (or typed) in columns.

3.4.3.6 NVivo Cognitive Map – Direct Causal Coding and Modelling

Causal linking words or phrases under Huff, Narapeddy & Fletcher's (1990) coding procedure involves placing a word or phrase in one column and then linking it with another word or phrase in a second column. A judgment is made in selecting a causal type of association between the linked words and/or phrases.

Huff, Narapeddy & Fletcher's (1990) coding schema for the association of concepts uses eight association codes devised by Axelrod (1976), plus two of their own codes (see Table 3.3).

Nath & Newell (1998) developed causal links in their research, referring to Huff, Narapeddy & Fletcher's (1990) coding schema. However, closer inspection of Nath & Newell's causal links within the cognitive maps presented in their research suggests that they appear to have only used or chosen to show associations with positive "+" or negative "-" associations in their cognitive maps.

For the current research project, the conventions used by Nath & Newell have been followed with "+/-", "=" and "e" used in addition to "+" and "-" associations.

Huff, Narapeddy & Fletcher (1990) and Nath & Newell (1998) appear to have manually drawn cognitive maps from their causal linking documents. The NVivo Model Explorer Tool enables immediate and dynamic visual linking of nodes – and the labelling of those links with appropriate causal codes plus for single or multiple directions of causality.

Coded nodes can be quickly added as text boxes in the Model Explorer Tool and linked with connector lines which can be edited to include single – or – multiple direction arrowheads, plus labelled with an appropriate causal association label.

Mapping commences with first two node links, and builds up as additional nodes are linked. Node boxes can be quickly edited, for example, to combine words and phrases (or to combine nodes), or deleted if necessary. Links between nodes can be quickly deleted or redirected to other nodes, plus the causal link labels can be changed if required.

Additional item text boxes for new words or concepts that may be added at the modelling stage can be introduced to the overall map. Clusters of linked nodes may be re-oriented or linked to additional item boxes if required.

Several nodes can quickly be linked up as a causal map and the map can be rapidly re-rendered for additional nodes and/or different links between nodes. Re-ordering of nodes for sequencing can be easily achieved using the Model Explorer Tool.

The final version of the NVivo cognitive map is saved as an NVivo model.

3.4.3.7 Final Translated Cognitive Map in PowerPoint

The cognitive map as produced by the NVivo Model Explorer Tool would in itself be a satisfactory final cognitive map. However, there are some limitations in NVivo that prompted further processing for this project. The NVivo Model Explorer offers very limited printing and filing formats, but at least the model or map can be transferred to a Clipboard and pasted into or MS-Word or PowerPoint. An example of a completed NVivo cognitive transferred to PowerPoint can be found in Figure 3.3.

Table 3.3: The Association of Concepts
- Instructions to Coders

/+/	A	positively affects Facilitates Advances Increases makes better Helps Promotes Expedites makes possible is necessary for	B
/-/	A	negatively affects makes difficult Hinders Hurts Impedes Prevents Inhibits changes for the worse	B
/⊕/	A	won't positively affect won't help won't promote is of no benefit to (construct negatives of /+/ above)	B
/⊖/	A	won't affect negatively won't hurt (construct negatives of /-/ list above)	B
/m/	A	affects in some nonzero way somehow affects in some way affects	B
/0/	A	has no effect on has no relation to doesn't matter for	B
/=/	A	is equivalent to is the same as is defined as	B
/≠/	A	is not the same as	B
/∈/	A	is a member of is an example of belongs to set	B
/∉/	A	is not a member of	B

CODING PROCESSES

- Finding the location of a relationship in a text
- Placing components in a sequence
- Identifying the nature of the relationship and assigning a linkage code
- Selecting the portion of actual text to enter one the coding sheet

Source: Huff, Narapeddy and Fletcher (1990, 315)

The transferred cognitive map usually needs reformatting for text and box size in PowerPoint (see Microsoft Corporation (2002b and 2003b). For this project, a methodological contribution is a format and schema for presenting cognitive maps. The new format maps are produced through further translation of the NVivo maps in PowerPoint, incorporating various changes:

- Nesting of nodes under distinct sections

- Flow wherever possible in columns and forward flow rather than backward or circular loops.
- Nesting of nodes denoted "e" ("examples of" or "sets of") rather than having multiple links
- Where possible, clear end sections of points in the map

An example of a cognitive map in a new final format is presented in Figure 3.4. The new format cognitive maps are capable of representing much more content than the "standard" format cognitive maps and they have the potential to accommodate more details, insights, and possible nested relationships. However, there are tradeoffs with complexity of content and detail in the new maps.

3.4.4 Interview Package

A complete **Case study Interview Package** was developed and the package template is presented in Appendix I.

The package was then adapted for face-to-face and telephone interviews. The face-to-face interview package contained extra pages and sections for compiling interview notes, in a full document format. The face-to-face (in-depth) interview package is presented in Appendix II. The telephone interview package was edited to contain slides with instructions and DSA models and Maps in a PowerPoint package, but without notes pages.

All participants, whether for face-to-face or for telephone interviews, were e-mailed a PowerPoint package of DSA models and Maps prior to their scheduled interviews. All but two interviews were recorded (with permission from participants prior to the interview) and transcribed into MS-Word documents.

Initially each package contained the complete set of Models and Maps plus extension questions covering:

- Shared Vision
- Dominant Logic
- Key Change or "Leverage Points"
- Key Strategic Marketing Issues

Packages were later edited and adapted for multi-staged interviews – typically for the DSA model and event chronology map in a face-to-face interview package, and the cognitive maps were covered in a follow-up telephone interview.

Packages were also edited for interviewees whose expertise or preference was relevant for only a selection of the models and maps.

3.5 Interview Stage

Telephone interviews with the participants from the first two case-studies were undertaken in December 2001. Face-to-face interviews with participants from four case-studies were completed in February 2002, with all interviews tape-recorded with permission from participants.

Experience gained from the December 2001 and February 2002 interviews indicated that in almost all cases, only the DSA model and events chronology mapwould most likely be covered in a 1 ½ - 2 hour interview.

Subsequently the interview process was split, with the first round of interviews covering the DSA model and events chronology map, and the second round of follow-up interviews covering the cognitive maps. Extension questions were asked during the follow-up interviews after covering the cognitive maps.

An additional extension question for the second round covered the identification of the type of software process or lifecycle used in the application development process. However, the responses to this question were folded back into commentary on development cycles and DSA models, rather than being treated as a stand-alone section in the case study chapter write-ups.

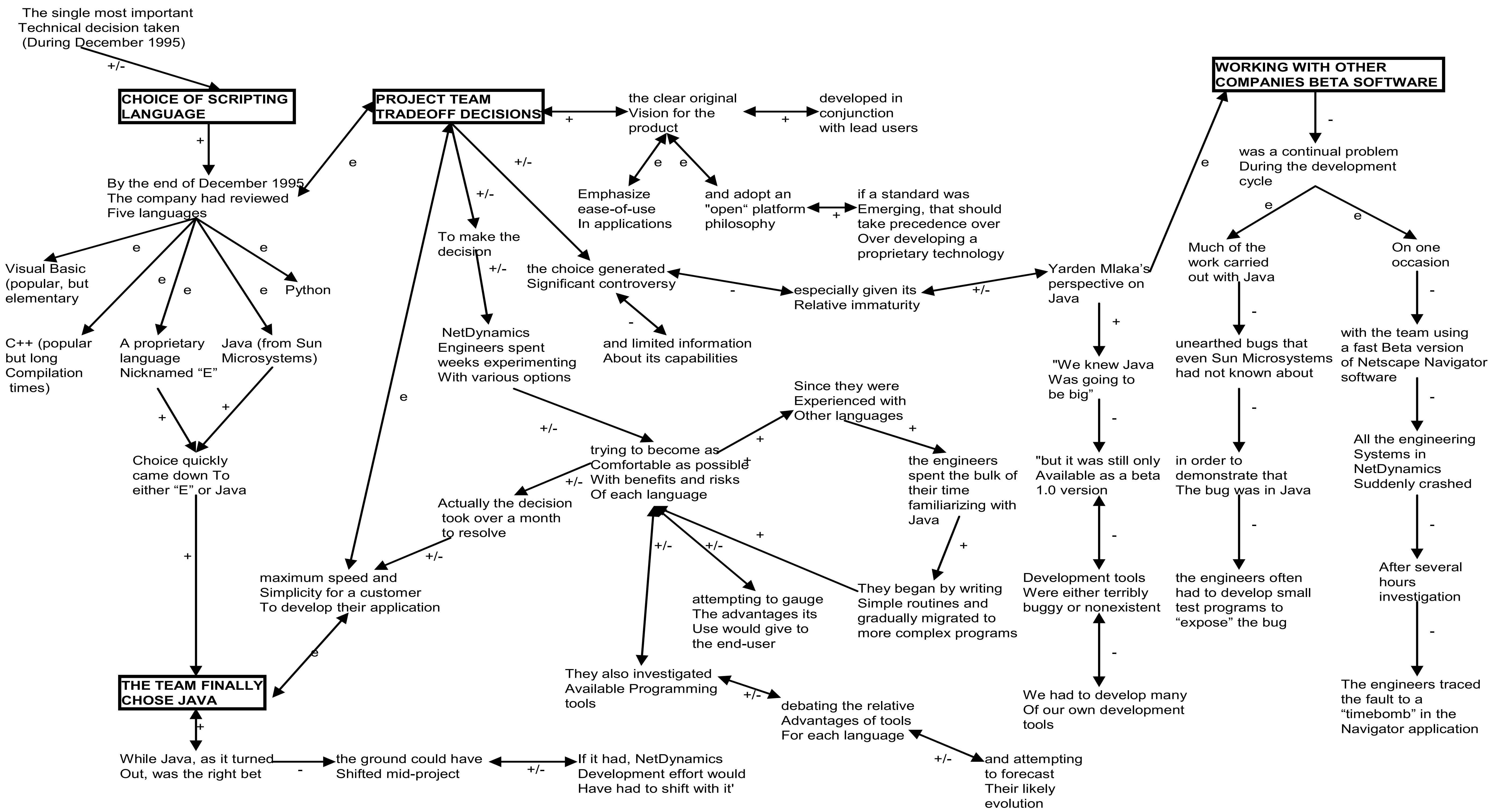


Figure 3.3: Example of Cognitive Map Produced Using NVivo's Model Explorer Tool
Extracted from NetDynamics Case Study - Selection of Java (November-December 1995); Created from data in Iansiti & MacCormack 1996

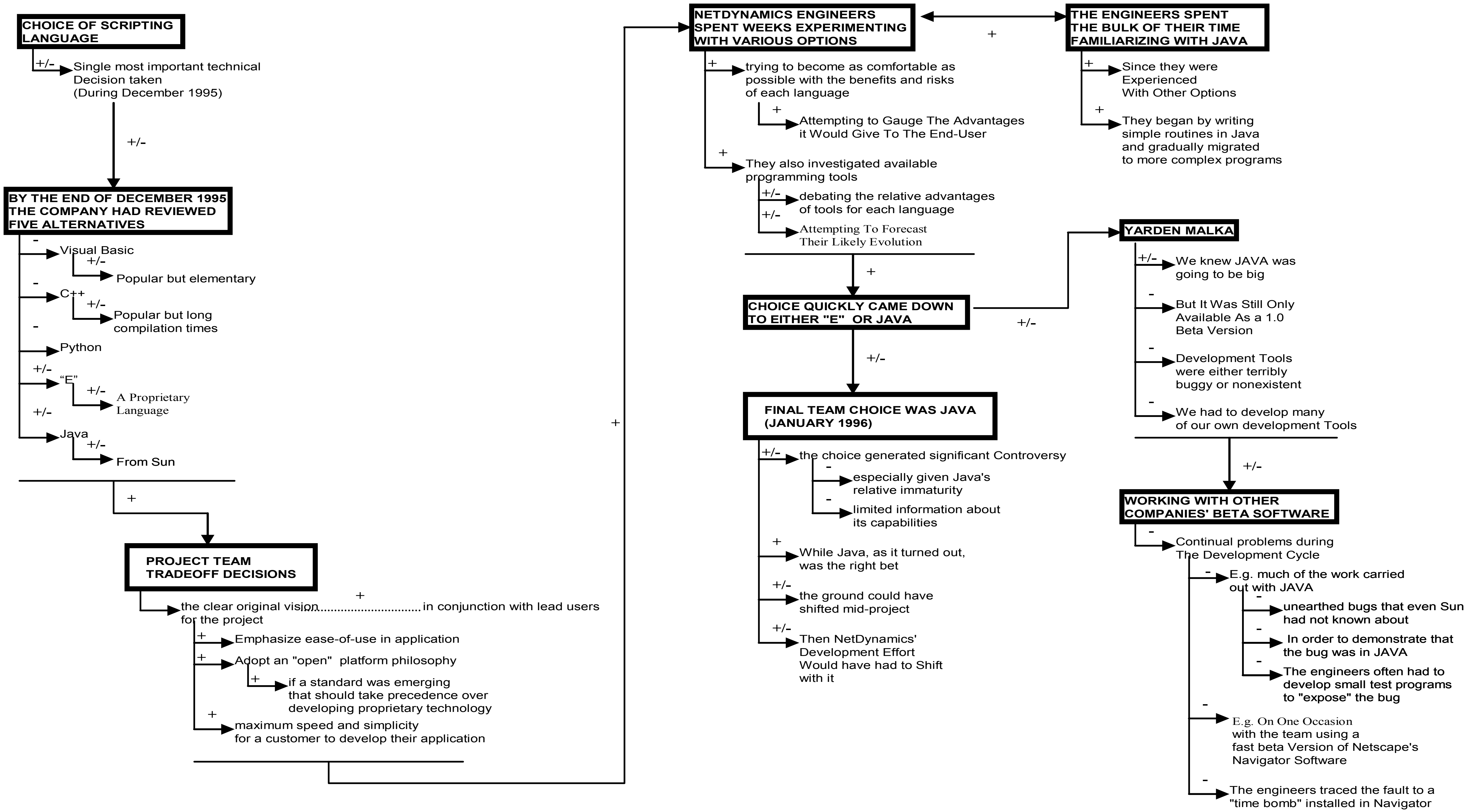


Figure 3.4: Example of Cognitive Map Translated In PowerPoint Based On NVivo Model Explorer Input - Extracted from NetDynamics Case Study - Selection of Java (November-December 1995); Created from data in Iansiti & MacCormack 1996

The second round of interviews was conducted by telephone during April – May 2002. An innovation during the second round was the use of a software application from Veritape Ltd., Veritape (Versions 1 and 2 – see Veritape Ltd. (2002 and 2003), enabling direct recording of telephone interviews to .wav or mp3 computer files. Where this application was used, permission was gained from the participants to record the interview, prior to commencing the interview.

Most participating decision-makers were interviewed twice for the project. There were three variations. Two decision-makers were the subject of an extended telephone interview that covered relevant models and was followed up with e-mail dialogue to validate inputs to these models.

One respondent completed an extended face-to-face interview which covered all models and maps for that case study and was followed with e-mail dialogue.

In one case study both decision-makers participated in a joint interview on the DSA model and event history map, but were followed up separately through telephone interviews on the relevant cognitive maps.

All models and maps were reviewed and updated from the interviews, plus a number of new DSA models and cognitive maps were developed. An example of an updated cognitive map from the NetDynamics Case study is presented in Figure 3.5.

Developing new DSA models and cognitive maps required an extension of the initial research methodology. New "vignettes" were written based on inputs from interviews and supporting documents. These vignettes were treated as extracts for the development of new DSA models or cognitive maps.

Compilation of the final thick case study descriptions using updated and new models and maps plus discussion on extension questions was incorporated in the final write-up of each case study chapter.

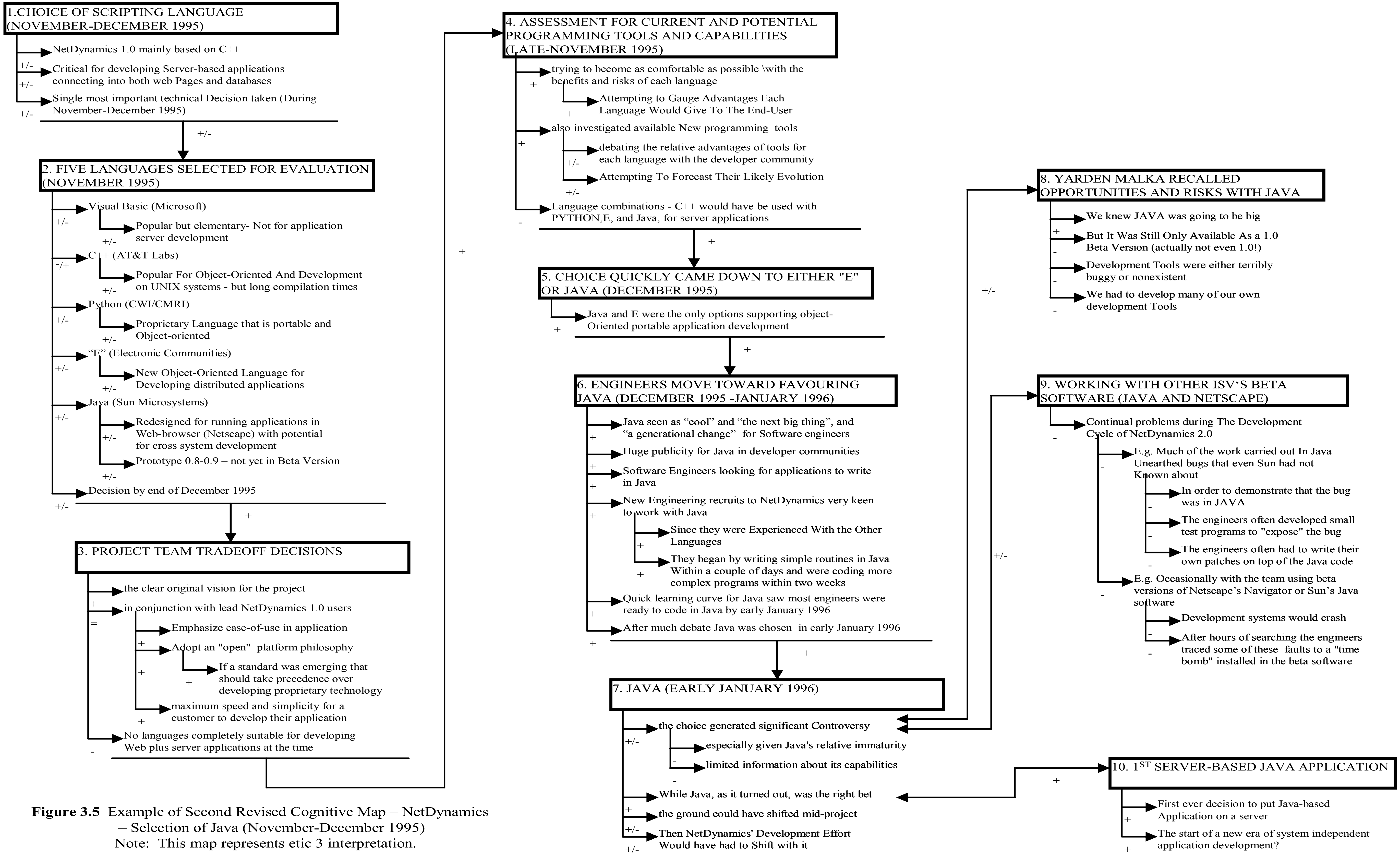


Figure 3.5 Example of Second Revised Cognitive Map – NetDynamics – Selection of Java (November-December 1995)
Note: This map represents etic 3 interpretation.

3.6 Expert Validation of Cognitive Mapping Processes

Research on cognitive perspectives of strategy is a form of humanistic inquiry and it is important to subject cognitive mapping processes and outputs to expert evaluation. Cognitive map evaluation was based on expert assessment against Hirschman's (1986) humanistic research criteria of:

- Credibility
- Transferability
- Dependability
- Confirmability.

In addition to Hirschman's (1986) criteria the maps were also evaluated for:

- Thickness (of content and description)
- Justification (value of revisiting accounts through the maps)

An External Auditor's Evaluation Form (EAEF) was developed, packaged and sent to eight researchers regarded as the top experts in the field of cognitive mapping and decision-making analysis. The EAEF Package is presented in Appendix 2.

The full package sent to the experts included:

- The EAEF cover survey
- One case study abstract (from the NetDynamics Case study)
- One cognitive map generated from NVivo and copied to PowerPoint (the map was selected from the NetDynamics Case study)
- The same cognitive map translated into new format in PowerPoint
- The same translated cognitive map updated after the interview stages.

While there would have been significant value in undertaking the expert validation using test cognitive maps prior to the interview stage, in the end more valuable insights and validation were gained from showing experts a full completed mapping process including updates from the interviews.

Three completed responses were received. All three experts validated the cognitive maps for credibility and confirmability. Two out of three positively validated transferability and dependability.

The third respondent did not invalidate transferability and dependability, but raised some inconsistencies between the case study abstract and the cognitive maps. This problem was partly due to incorporating the abstract rather than the relevant original HBS extract in the package. However, the comments were still quite relevant to the cognitive mapping process.

All three respondents were positive about the degree of thickness of description, although one respondent believed that it was too difficult to determine quality of thickness until the final cross-case analysis was written up for the project.

All three respondents were very positive about justification in terms of the research being justified through revisiting prior case study environments/information and confirming and adding to the data reported by these cases and building theories in use. There was very strong endorsement of the cognitive map incorporating inputs from interviews.

One out of three respondents saw the cognitive mapping process as adding a lot of significant contributions to the study; one respondent believed that the cognitive mapping process added some minor contributions to the study; while one respondent believed that the maps themselves added little contribution – but added that the cross-case study analysis and write-up would be the place to judge real contributions to the study.

Some general observations were made about the actual cognitive maps. There is considerable enthusiasm for seeing the cognitive perspectives drawn out of the cross-case analysis for the project; in other words the respondents were very keen to see the final analysis and outcomes of the project.

One respondent commented that the complexity and detail of the cognitive maps would restrict their use with commercial practitioners. On the face of it, the comment seems valid and is not unique. However, almost every person actually interviewed for the project went out of their way to compliment the high quality of the maps in terms of clarity, flow, and especially for content!

Overall, the cognitive mapping process was validated and there was considerable support for the research project. There may have been even stronger support if the exact original HBS case extracts had been used rather than the case abstracts, but even with the abstracts the experts' inputs were quite valuable and positive.

3.7 Analysis, Generalisation and Theory Development Stage

Summaries and findings from each case study chapter were further analysed through Word Frequencies in TACT and NVivo. A composite cross-case DSA model was developed. However, time constraints did not permit the development of full cross-case cognitive maps using NVivo.

Nevertheless, further analysis to identify key overall concepts was undertaken using the Copernic Summariser application which was selected in November 2003 (see Copernic Technologies 2003).

Copernic Summariser is a new generation of “semantic web” – based applications using advanced algorithms to determine key concepts in documents. Copernic Summariser appears to be superior to both TACT and NVivo for identifying effective key concepts, although it does not yet have a mapping tool.

At this stage Copernic Summariser concepts could be input to and coded in NVivo for mapping. However, there is great potential for applications to emerge from semantic web analysis that not only enable selection of concepts, but also actual semantic mapping of them.

Cross-case key concepts and summary findings were assessed directly against Structuration propositions. Write-up of these findings, plus additional proposition generation, completed the research methodology for the project. There is more discussion on theory generalisation and generation in Chapter 10.

3.8 Representation of Advanced Hermeneutic Analysis

The advanced hermeneutic analysis framework proposed for the research project is represented as an evolutionary schema starting at the first level of emic 1 and etic 2 data analysis, and then evolving through at least four additional levels to encapsulate at least three rounds of etic data and two rounds of emic data. A template of the full schema as expressed for the maximum of six levels of advanced hermeneutic analysis is presented in Figure 3.6. An advanced hermeneutic analysis was completed for each of the six case-studies in the project, and is reviewed in Chapter 10.

3.9 Summary and Methodological Contributions

The case study research methodology for the project supports advanced hermeneutic analysis and extended DSA with new and extended research collection and analytical techniques and applications.

Six case-studies were created for the project, based on Harvard Business School Case study accounts of decision-making associated with the development and delivery of Internet-based software applications. A rigorous selection process produced a strong final set of six software application companies – all of which either produced disruptive software technologies, or applications that contributed to disruptive software technologies.

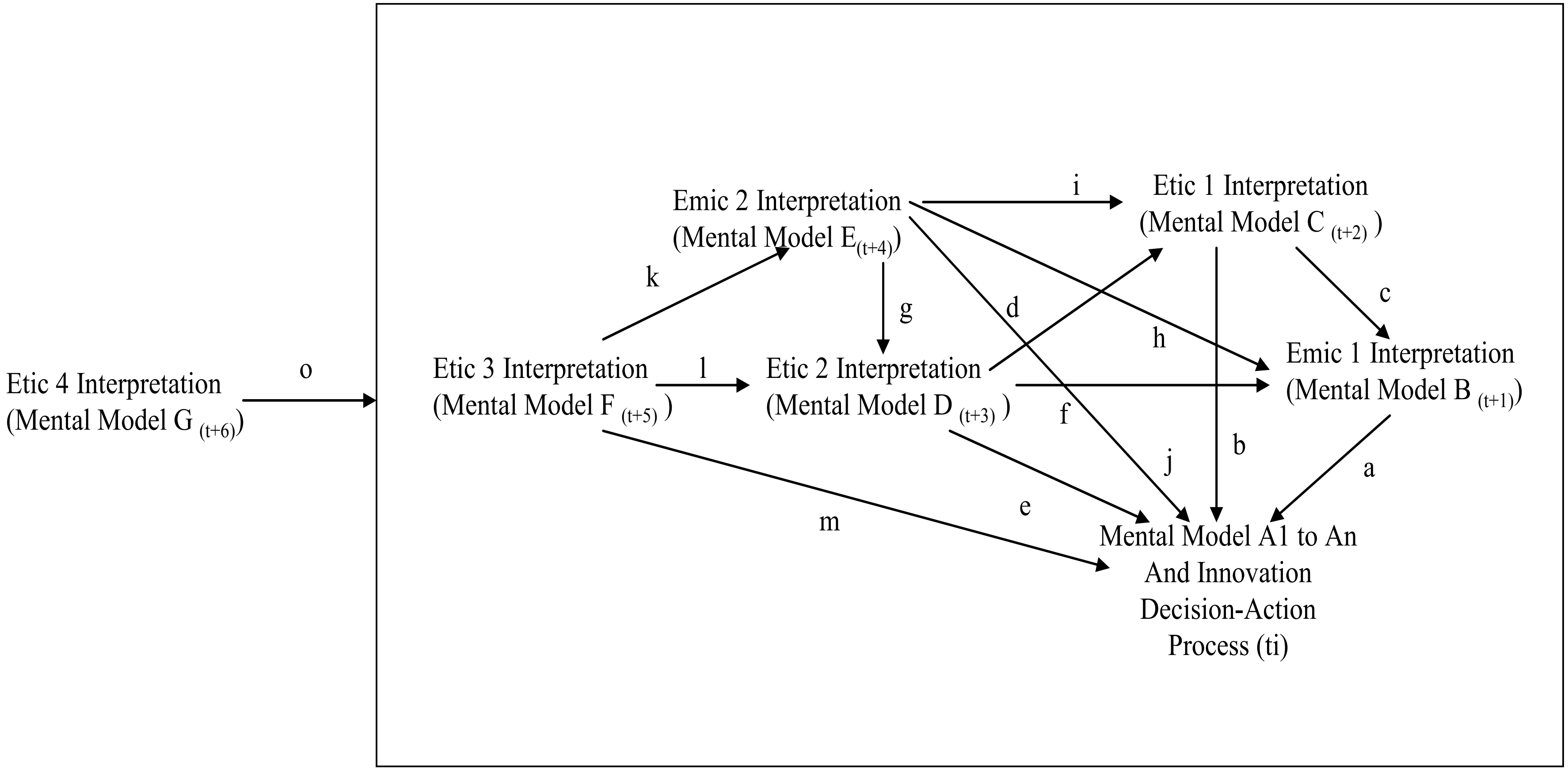
The extended form of DSA for the project included an enhanced representation of DSA models, new event chronology maps, and a new methodology for developing cognitive maps with new formats using the NVivo software package. The extended DSA "platform" was then extended through questions during the interviews covering: shared vision, dominant logic, key leverage points, key strategic marketing issues, plus types of software processes or system development lifecycles used for application development.

The extended DSA package was then validated or updated through multiple in-depth interviews (either face-to-face or by telephone). Telephone interviews were recorded using a new technology from Veritape, enabling direct digital recording of interviews into audio file formats on a PC.

The research methodology for creating both the original cognitive maps and maps updated after the interview process were validated by experts using a new Expert Auditor's Evaluation Form (EAEF) system that has the potential to be developed further as a research tool. The final thick descriptions of the DSA platform plus extensions were written up in a chapter for each case study.

Further cross-case analysis was undertaken using new "semantic-web" based concept identification applications, and further cross-case summary lists were created. A composite DSA model was created. Summary outputs were assessed against Structuration propositions, with final and additional propositions written in a final theory generalisation and development chapter.

Figure 3.6 Representation of Six-Level Hermeneutic Analysis



The following six chapters are detailed case-studies containing complete advanced hermeneutic analysis complemented by extended DSA (the exception is the Red Hat Case study where the complete analysis is a combination of the Introduction and the Red Hat Case study chapters):

- Chapter 4 – Red Hat
- Chapter 5 – Zaplet
- Chapter 6 – NetDynamics
- Chapter 7 – Kana Communications
- Chapter 8 – Intuit QuickBooks
- Chapter 9 – Trilogy.

Methodological contributions from the project include:

3.9.1 Use of a combination of software applications for word analysis, word and concept coding, and cognitive mapping

3.9.2 Use of a software application for mapping of cognitive perspectives related to marketing decision analysis within business-to-business contexts, and in the software industry

3.9.3 Creation of a new format for cognitive maps

3.9.4 Recording of telephone interviews directly to digital audio computer files using the Veritape application in a marketing research context (and possibly a business research context)

3.9.5 Development of a process for collecting experts' evaluation of humanistic research related to business and marketing cognitive perspectives

3.9.6 Development of a representation of multilevel advanced hermeneutic analysis

3.9.7 Application of "semantic-web" concept identifier software application to cross-case summary analysis in business-related cognitive research.

The first six methodological contributions are applied effectively in the case-studies for the project. The last methodological contribution is applied in cross-case analysis within the final chapter of the dissertation.

Chapter 4: Case Study – Red Hat

Abstract

The Red Hat Case study chapter is a continuation of the Introduction Chapter in which a **multilevel hermeneutic analysis** was presented, complemented by an **extended DSA package with cognitive mapping** for one key aspect of the case study (early development).

The chapter covers additional cognitive mapping and extension questions extending an account of decision-making associated with the design, development and delivery of applications through the new and emerging "open-source" development community' using storytelling methodologies – an advanced hermeneutic analysis, including an extended form of **decision systems analysis** (DSA) incorporating **cognitive mapping**.

Strategic thought associated with the creation of the Linux operating system and its subsequent packaging with installation and version management applications tools by Red Hat Inc is mapped and discussed in the case study.

Key findings from the case study highlight very high levels of shared vision, a form of collective coordinating dominant logic and indications of a new form of "open-source" marketing.

The Internet is highlighted as an exemplar of a computer-mediated environment for collective, global, concurrent and rapid application development and delivery.

4.1 Background

On 11 August 1999, Red Hat Inc. of Raleigh, North Carolina became the first "open-source" software company to go public on the NASDAQ stock exchange. Red Hat is a major provider of the Linux operating system packaged with several value-added applications, plus the company is actively involved in the development of various open-source application technologies.

The story of Red Hat augments and parallels that of "open-source" software. It is an account of a new and still emerging philosophy and methodology for application development that depends on thousands of developers, either as individuals or in companies or various organisations using the Internet to concurrently develop and update applications. It is also about the emergence of Linux as an operating system for Web-

Servers to compete against established Microsoft and UNIX operating system competition.

The Red Hat Case study explores the creation of the Linux operating system as an extension of the “open-source” philosophy and organisation, and then packaging by Marc Ewing into Red Hat’s version of Linux. The study chronicles Red Hat founders Marc Ewing and Rob Young’s interest and motivations in Linux from 1993 through to the formation of Red Hat in 1995, and on to the public float of the company in 1999. The case study extends to Red Hat’s development philosophies in 2001-02.

Red Hat Inc. was included in the final case study list for the project as it met the initial criteria:

- Software has to be “beta”-tested and commercially introduced
  - Linux and Red Hat’s supporting application met this criteria with several versions released over the timeframe covered by the case study
- Case-studies were written by well-known authors
  - The original case study was a Harvard Business School Case study supervised by Professor Alan MacCormack, who is recognised as an expert in software product development
- The application development process studied must be over a timeframe of more than 12 months
  - the original case study covers a timeframe of over six years from 1993 to 1999, which extension to the early 1980s for open-source software history
- Names of case study companies and participants can be disguised if requested
  - Permission was gained from Rob Young, then CEO of Red Hat Inc. for interviews and use of that information in the research project without requirement for disguise
- Applications must be Internet-based
  - Open-source application development of Linux is by definition conducted through the Internet, and is also distributed through the World-Wide Web. Red Hat’s applications are also available over the World-Wide Web, plus they assist management of Linux, which is one of the most popular Web-server operating systems in the world

Additional case study selection criteria included:

- Size: Large
- Success: Yes
- Disruptive Technology: Yes, Linux (and emerging open-source development)
- Harvard Business School Case: "Red Hat And The Linux Revolution", MacCormack & Herman (2000)

The initial DSA model, events chronology map and the three cognitive maps for the Red Hat case were developed from extracts drawn from MacCormack & Herman’s (2000) case study.

The DSA models, events chronology maps and cognitive maps, plus further insights into the Red Hat case study in the context of business-to-business marketing DSA were presented in Pattinson, Woodside & Miller (2002). Additional insights into the creation and development of Linux were reviewed and incorporated from Torvalds & Diamond (2001).

MacCormack & Herman’s (2000) coverage of “open-source”, Linux and Red Hat’s application development provides adequate detail for the development of a representative DSA model and to undertake further analysis on application development and delivery.

The DSA model and maps for the case were updated, plus extension questions on dominant logic, shared vision, key leverage points and strategic marketing issues were addressed through an interview with the then CEO and one of the founders of Red Hat (see Young 2001).

The outline and chapter structure for this case study are presented in Figure 4.1, continuing on from the Introduction chapter through addressing two additional cognitive maps and extension questions.

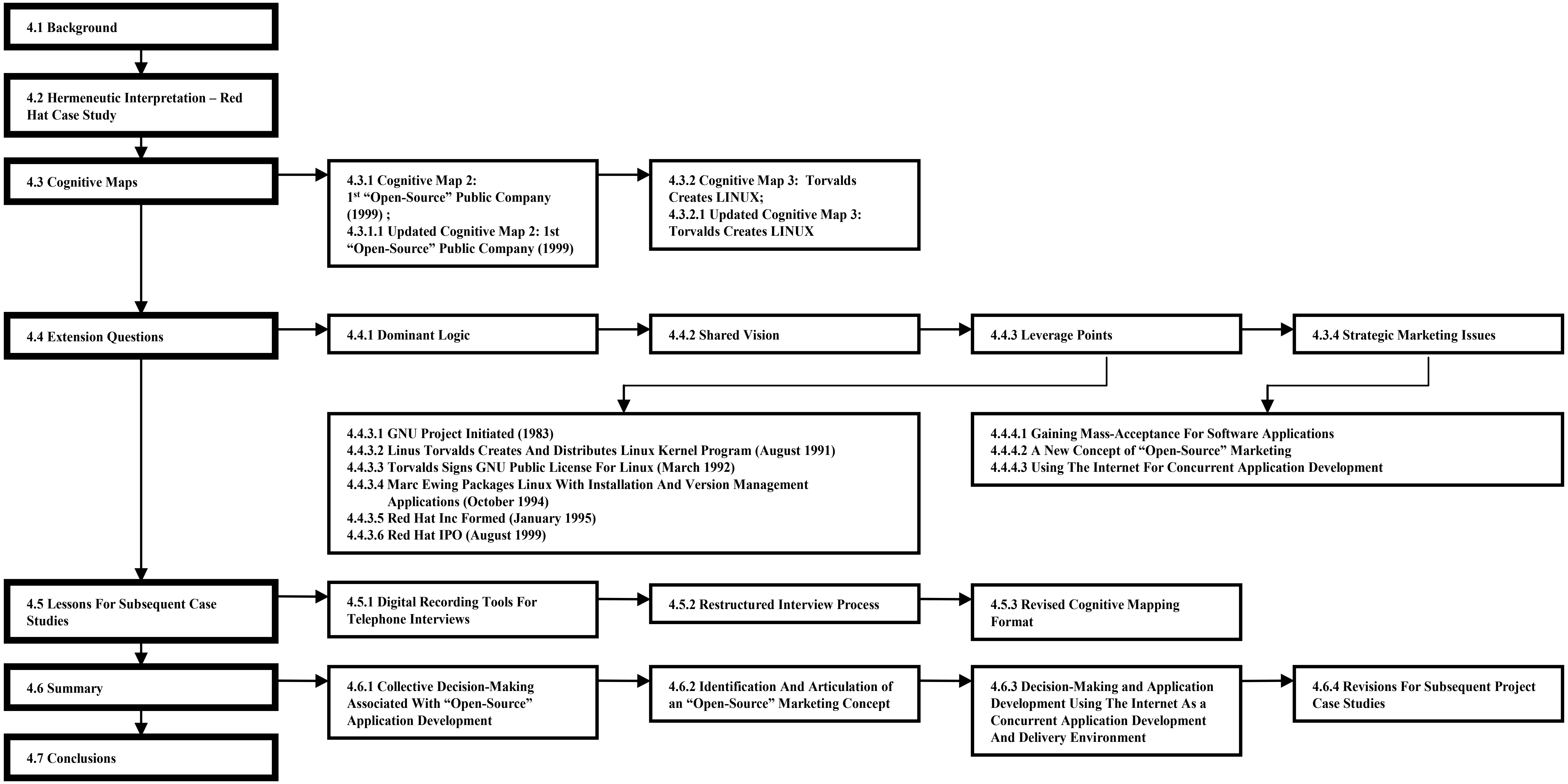


Figure 4.1 Red Hat Case Study – Chapter Structure

4.2 Hermeneutic Interpretation – Red Hat Case Study

The Red Hat case study was advanced as an application of the proposals described in Chapter 1 for advancing hermeneutic research in B2B decision-making. This section recaps and summarises hermeneutic interpretation for the whole Red Hat Case study

Five levels of hermeneutic analysis were developed to address decision-making issues, events, and linkages in the Red Hat case study. These levels are presented in Exhibits 1.3 to 1.7.

The original case study by MacCormack & Herman (2000) serves as the etic 1 report of the case study – although there are elements of emic 1 reporting in the form of direct quotes from decision-makers in the account. Quotes and perspectives from decision-makers were encapsulated in the Level 1 emic analysis in Exhibit 1.3. Key case study issues as articulated by MacCormack & Herman (2000) are highlighted in the Level II analysis presented in Exhibit 1.4. These issues included:

- Account of evolution of open-source software philosophy – creation of Free Software Foundation (late 1980s); GNU Project (late 1980s); General Public Licence; emergence of open-source initiative (OSI – late 1990s)
- Account of Linus Torvalds' RDC of Linux
- Description of emergence of Red Hat
- Description of Red Hat's application development and distribution

The Level II analysis also explored mental models and decisions/actions associated with these issues.

The third level of analysis, as presented in Exhibit 1.5, is an etic 2 representation based on the current researcher's summarisation of event milestones and the emic 1 sensemaking views identified in the data in the MacCormack & Herman (2000) case study. Subsequently to developing Exhibit 1.5, emic 2 and etic 2 interpretations, along with DSA and event maps, provide revisions to Exhibit 1.5.

MacCormack & Herman's (2000) coverage of the creation of Red Hat Inc. and development and packaging of versions of Red Hat's Linux incorporated sufficient detail for the development of an initial (etic 2) representative DSA model and to undertake further analysis on application development and delivery. The initial DSA, event, and cognitive maps reported below offer details supporting Exhibit 1.5. Etic 2 perspectives are mainly based on the initial DSA model, initial events chronology map, and initial cognitive maps.

Key etic 2 issues included:

- New open-source development community and philosophy
- Red Hat created from complementary needs between Ewing &Young
- Importance of Internet for open-source community application development
- Red Hat's approach to symbiotic interaction with open-source development community.

The etic 2 DSA model and maps for the case were updated following questioning of the accuracy and completeness of the original MacCormack & Herman's (2000) case – the collected data is emic 2 material and was added during the fourth level of analysis (See Exhibit 1.6).

Thus, additional (emic 2) data were collected for etic 3 descriptions and interpretation of the Red Hat decision-making process, as reported in the original case study and for a period of two years beyond that reported MacCormack & Herman. Emic 2 data consisted of responses from a long interview with the CEO of Red Hat in 2001 (see Young 2001).

Exhibit 1.7 is the final level of analysis; with the addition of etic 3 data summarising emic 2 interpretations of mental models and events covered in the original case study, as well as the work completed for the etic 2 interpretation – including the DSA, event, and cognitive maps developed for the etic 2 interpretation.

The revised DSA, event chronology, and cognitive maps presented in subsequent sections below follow from the emic 2 interpretations, and these maps are part of the etic 3 interpretation.

The new data from the emic 2 and etic 3 rounds of interpretation validated etic 2 data and highlighted new insights related to:

- Extension of open-source approach to developing advanced software applications
- Open-source approach applied to software-based information products and services
- "Open-source" marketing concept.

Overall, the five-level hermeneutic framework for the Red Hat case study provided a strong and comprehensive framework for capturing and extending dynamic sensemaking for Red Hat's decision-makers.

DSA models, event chronology maps and the first of three sets of cognitive maps were addressed in Chapter 1 and incorporated in the advanced hermeneutic analysis. The rest of this chapter complements the hermeneutic analysis with the remaining two sets of cognitive maps and discussion on extension questions.

4.3 Cognitive Maps

All cognitive maps created for the Red Hat case study were designed to complement DSA models and events chronology maps through additional exploration of selected decision-making issues and contexts.

All cognitive maps were created using the following procedures::

1. Selected extracts from MacCormack & Herman's (2000) case study were extracted to text files
2. One text file version of each extract was processed through the TACT package to produce a complete word-frequency list
3. Word frequencies were reduced through one or two iterations to ranked lists of words with frequencies ranging from over one to three words
4. A second version of the text file was input to NVivo
5. Ranked words were input to NVivo and coded as nodes in association with the text file
6. The overall extract was deconstructed into selected phases and sentences and coded directly as nodes through the InVivo tool within NVivo
7. The nodes associated with each extract were input to the Model Explorer and mapped using links incorporating codes of association plus addition node "items"
8. Several iterations of dynamic mapping were undertaken with the final cognitive map exported from NVivo to PowerPoint
9. Each PowerPoint version of the cognitive map was reformatted into a more readable and project-consistent format
10. The revised PowerPoint cognitive maps were presented to interviewees and validated or modified
11. The updated PowerPoint cognitive maps incorporated changes after the interview process.

Three sections were extracted from MacCormack & Herman's (2000) case study for further text analysis and creation of cognitive maps. Three initial cognitive maps were created for the Red Hat case study. The first cognitive map was discussed in Chapter 1, while the remaining two cognitive maps are addressed in this chapter. The three cognitive maps addressed the following themes:

Early Development (1993-96) (See Chapter 1)

4.3.1 1st “Open-Source” Public Company (1999)

4.3.2 Torvalds Creates Linux

The first cognitive map is an extended insight into the creation and early growth of Marc Ewing’s packaged version of Linux – and was addressed in the Introduction chapter.

The second cognitive map traces Red Hat Inc.’s path to become the 1st publicly listed “open-source” company in August 1999.

The third cognitive map provides deep insights into the genesis of the “open-source” concept and Linus Torvalds’ creation of Linux.

4.3.1 Cognitive Map 2 – 1st “Open-Source” Public Company (1999)

Development details for the second cognitive map include:

- Extract section, size and exhibit:
  - Opening quotes, introduction “Open-Source Software and the Linux Revolution” section, MacCormack & Herman (2000, pp. 1-2); ½ page; Exhibit 4.1
- TACT word frequency lists and comments on key words:
  - First & final pass: 19 words with frequency > 2
  - Final ranked word list: Table 4.1
  - Highly ranked words appear to capture the main issue of the extract that is, *Rob Young*, as CEO of *Red Hat* describing *develop*ment of the *Linux operating system* using *open-source software* (words in italics were ranked highly in the Word List).
- NVivo ranked word, sentence and phrase node count = 27
- Association of concepts coding used: +, -, +/-, “e”
- NVivo PowerPoint cognitive map: Figure 4.2
- Reformatted PowerPoint cognitive map: Figure 4.3.

The final cognitive map presented in Figure 4.3 provides a cognitive description of Red Hat’s (or more precisely its CEO Bob Young’s) perception of its positioning and success at the time of the company’s initial public offering on 11 August 1999.

Exhibit 4.1 Text Extract: Red Hat – Early Development (1993-96)

Red Hat and the Linux Revolution

What we realized very early on is the one unique thing we were delivering was not technology at all. This is not a better operating system. Solaris is a better operating system than Linux for many things. Windows NT is a better operating system for many other things. The one unique thing that we could do that no one else could do was, for the first time, we were giving control of the operating system to the user. – Bob Young, CEO, Red Hat Inc.

Bob Young ducked as the first champagne cork flew over his head and out through the open window of the office. It was 10:55am on August 11, 1999. Red Hat's stock was about to start trading on the NASDAQ. Priced at $14 per share during the IPO process, the initial demand had been strong. The management team were waiting for the first price indication to appear on the trading screens at the office of their lead underwriter, Goldman Sachs, but the screens were a little confusing. "So what does the number 46.25 beside our ticker symbol [RHAT] mean?" Young asked. "Actually, that's the opening price Bob – not bad for a company that gives its products away for free!"

As CEO of Red Hat, the first open-source software company to go public, Bob Young would clearly have a lot to occupy his mind over the next few years. Indeed, he wasn't really sure what Red Hat would look like at the end of that timeframe. One pressing issue however, would have to be resolved in the next few weeks. It concerned the focus for Red Hat's future development efforts. Several managers had raised the idea of developing applications that would run on Red Hat's main product – the Linux operating system. But Young was not sure that an open-source approach would prove effective for these types of products. If it would not, should Red Hat consider the use of a different approach to development?

Exhibit 4.1 Text Extract: Red Hat – Early Development (1993-96) (Cont'd)

Open-Source Software and the Linux Revolution

By August 1999, the Linux (pronounced 'Lien-uks') operating system had become the fastest growing server operating system in the world. In contrast to the most popular competing server operating system of that time – Microsoft's Windows NT – however, Linux was not developed by a single company. Instead, it was developed through the efforts of thousands of individual programmers around the globe, who contributed new features, tested new versions, and wrote "patches" to fix the errors found during these tests. These efforts were coordinated by a handful of highly skilled software developers, the most important of whom did not even work on the project full time. This approach to development, founded upon widespread access to the code base of the evolving product, was known as "open-source" development. By 1999, open-source projects such as Linux had achieved widespread recognition in the mainstream business press. Their roots however, could be traced back to the beginnings of the computer era.

Source: MacCormack & Herman 2000, 1-2

Table 4.1 Red Hat – First "Open-Source" Public Company (1999) – Ranked Word List

Anchor Concepts	Frequency
young	8
red	8
operating	8
hat	8
develop	8
system	7
Linux	7
that	7
open-source	5
not	6
would	6
first	4
time	4
thing	4
product	4
software	3
price	3
could	3
1999	3

Source: data analysis performed using TACT (1997)

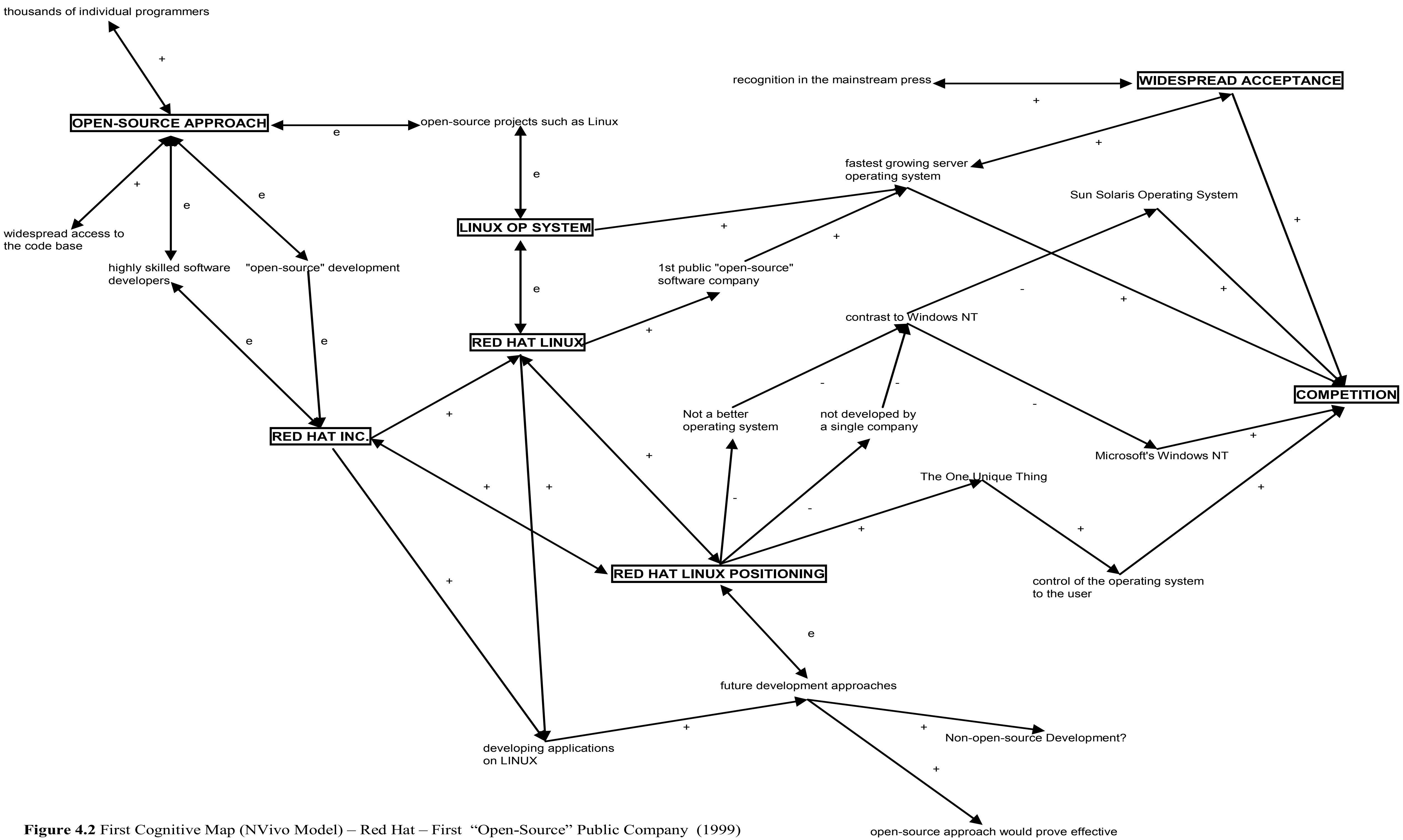


Figure 4.2 First Cognitive Map (NVivo Model) – Red Hat – First "Open-Source" Public Company (1999)
Created from data in extract (Exhibit 4.1)

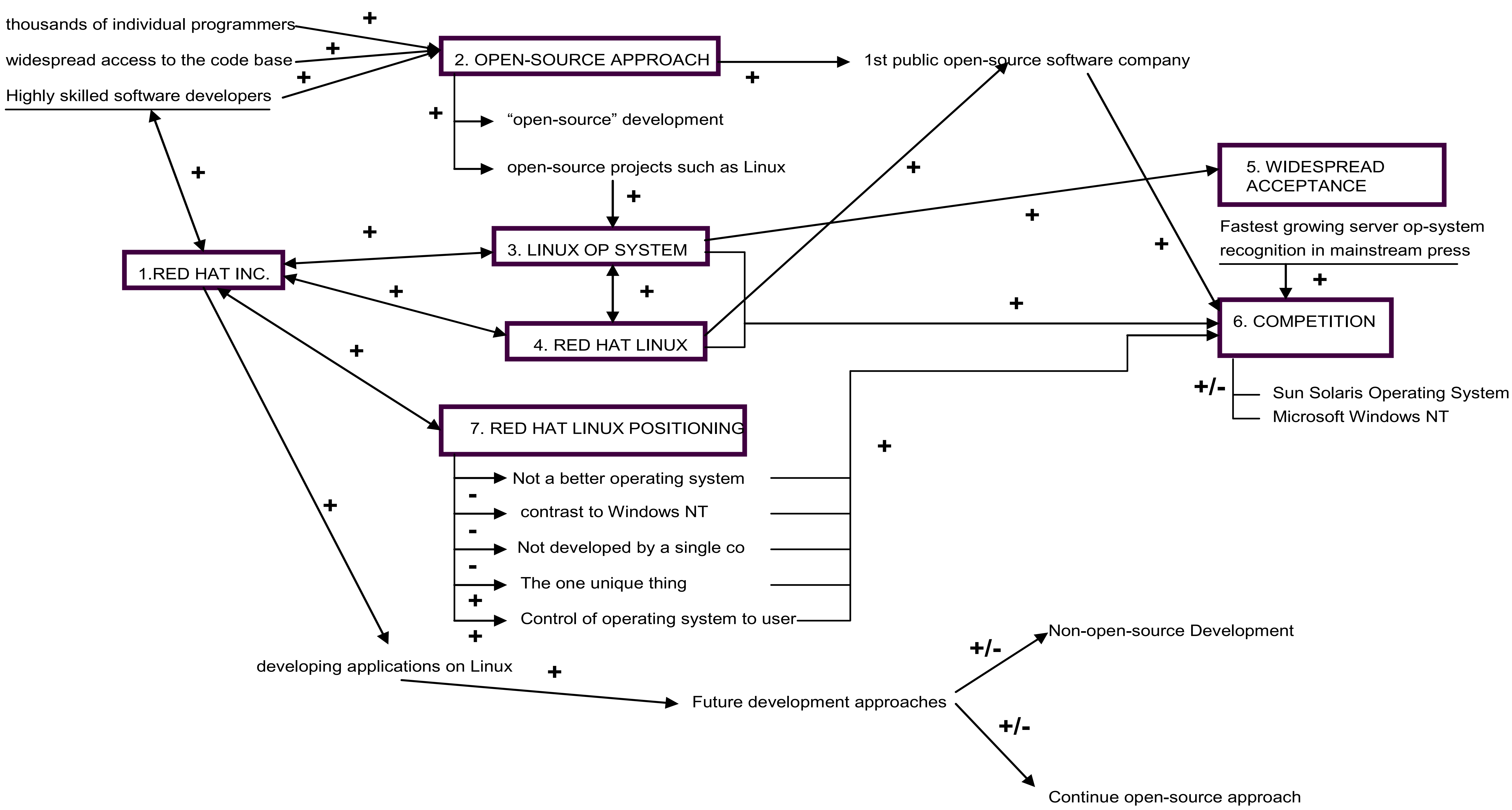


Figure 4.3 Revised Cognitive Map – Red Hat – First "Open-Source" Public Company (1999)
Translation of NVivo Map
Note. This revision was shown to participant in emic 2 stage for his (Young's) interpretation.

Although the cognitive map is a snapshot of a later timeframe from MacCormack & Herman's (2000) case study, it is not a set of insights that have any chronological sequence. Consequently associating concepts logically was based using the terms (or "items") Red Hat Inc., Red Hat's Linux, Linux, open-source, and competition as multiple and interconnected "anchors" in the cognitive map.

Red Hat's linkages to the open-source community (as discussed in Chapter 1) are deep and multidimensional. Red Hat obtained the Linux operating system directly through the open-source community but also worked on open-source projects that were shared throughout the same community.

Red Hat is effectively an open-source ISV packaging Linux with applications to improve the installation and management of Linux.

A major aspect of Red Hat's decision to use, package and develop open-source applications was the pool of thousands on individual programmers and highly skilled software developers within the open-source community who could concurrently develop applications and share the source code for further upgrading, modification and redistribution.

As the first open-source company to go public in August 1999, there was considerable interest in how a company could succeed by effectively offering a product for "free" using open-source software licensing. At the time Red Hat was the dominant Linux supplier and Bob Young's positioning of Red Hat's Linux was representative of Linux and other open-source applications.

Young outlined Red Hat's Linux's positioning as an operating system not developed by one company, but focused on giving control of installing, managing, upgrading and managing the operating system directly to the user.

Giving such a high level of control to users was attractive enough to make Linux the fastest growing server operating system during 1998-99 – both in terms of press recognition and real operating system shipment numbers (see IDC figures in MacCormack & Herman 2000, p. 15).

By 1999 Linux was serious competition for the Solaris operating system (Sun's version of UNIX) and Microsoft Windows NT (although Windows NT was also rapidly growing in market share at the same time).

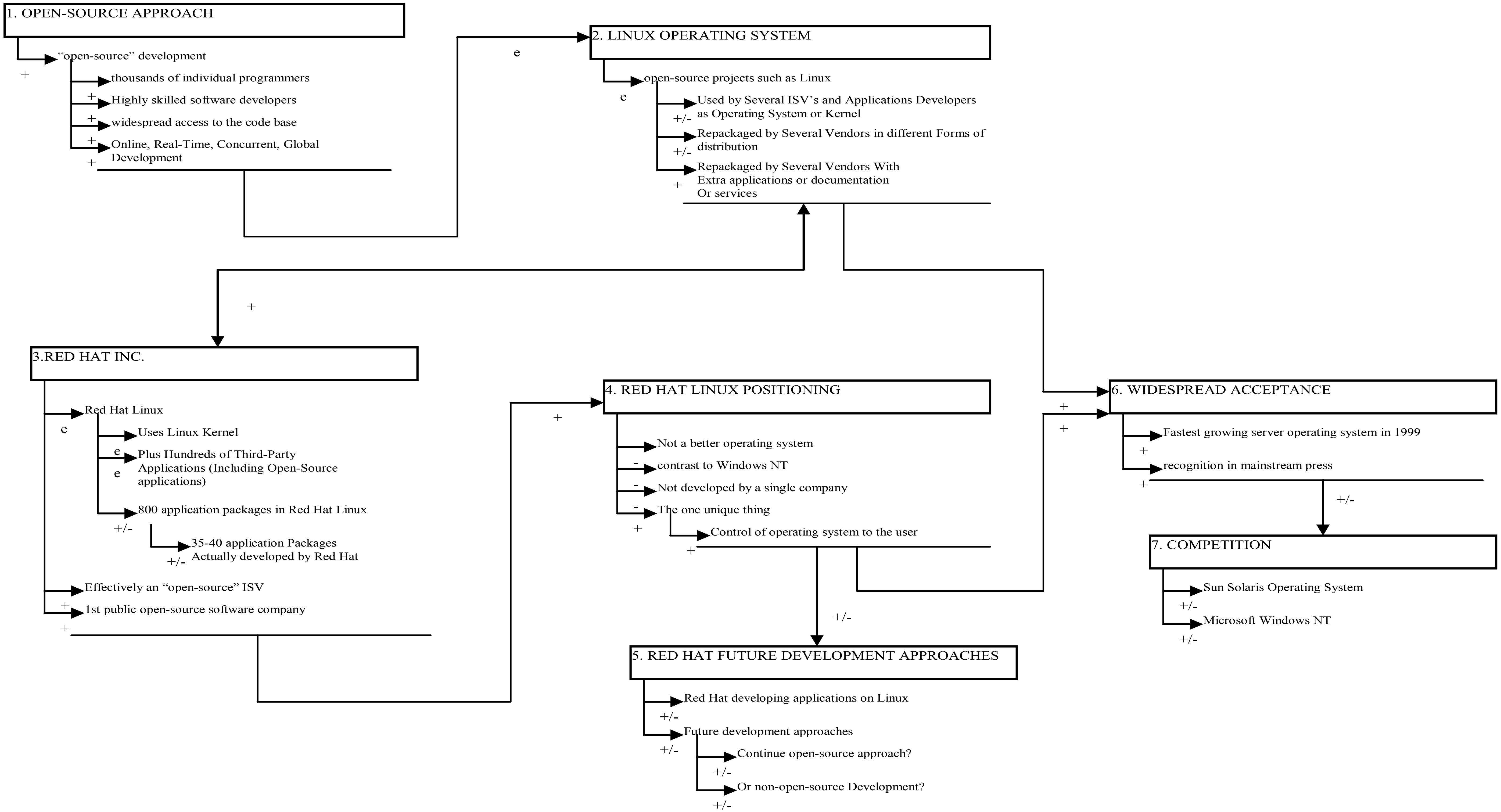


Figure 4.4 Second Revised Cognitive Map – Red Hat – 1st "Open-Source" Public Company (1999)
Note. This map represents etic 3 interpretation.

The cognitive map captures a key decision-point regarding whether or not Red Hat should continue to develop new applications using an open-source approach. The map does not contain answers or options for this decision-point. DSA modelling does not address any alternative to an open-source approach – but the updated DSA model highlights the main existing and potential revenue source as service and support contracts.

4.3.2.1 Updated Cognitive Map 2 – 1st "Open-Source" Public Company (1999)

The cognitive map addressing Red Hat as the 1st "Open-Source" Public Company (1999) was not directly discussed during the interview with Young (2001). However, Young provided insights during the interview that were incorporated into an updated version of the cognitive map, as were other comments and structural changes (see Figure 4.4).

The open-source approach to application development can be characterised as "online", "real-time", "concurrent", and "global" (see additional comment associated with new box 1 on the Open-Source Approach).

The section on Linux Operating System (box 2, previously box 3) was expanded to position application developers using Linux effectively as ISVs for Linux. Linux could be freely downloaded through OSI and repackaged for different forms of distribution as a component within ISV applications.

Sections of the original cognitive map addressing Red Hat positioning (box 4), Widespread Acceptance (box 6) and Competition (box 7) were upgraded slightly for structure and format. The actual decision-point regarding continued application development using the open-source approach was consolidated into a new section, "Red Hat Future Development Approaches (box 5).

The updated cognitive map improves on the decision-making and insights of the original cognitive map with some additional insights and depiction of strategic thought and decision-making faced by Red Hat at the time of its IPO in August 1999.

4.3.3 Cognitive Map 3 – Torvalds Creates Linux

Development details for the third cognitive map include:

- Extract section, size and exhibit:
    - "The Linux Operating System", MacCormack & Herman (2000, p.3); ¾ page; Exhibit 4.2
- TACT word frequency lists and comments on key words:
    - First pass: 35 words with frequency > 1
    - Second & final pass: 16 words with frequency > 2
    - Final ranked word list: Table 4.2

- The ranked word list highlights themes related to *Torvalds develop*ing his *own kernel program* that became the *GNU* project's *operating system* (italicised words were ranked highly in the Ranked Word List).

- NVivo ranked word, sentence and phrase node count = 42
- Association of concepts coding used: +, -, +/-, "e"
- NVivo PowerPoint cognitive map: Figure 4.5
- Reformatted PowerPoint cognitive map: Figure 4.6

The map tracks insights and events associated with Linux Torvalds creating Linux up to Torvalds delegating ongoing Linux development to core development team in 1993.

Exhibit 4.2 Text Extract: Red Hat – Torvalds Creates Linux

The Linux Operating System
While the GNU project succeeded in developing many of the essential components that surround an operating system, by 1990 it had still not developed the core of such a system – the kernel.
Given that each of the GNU programs was designed to be compatible with existing UNIX systems, this was not at first a problem. Indeed, the GNU C Compiler and GNU Emacs editor had become extremely popular programs in their own rights. However, without the kernel, the GNU project could not claim to have met its initial aims – that of developing a completely free operating system. Luckily, help was at hand in the form of a Finnish undergraduate student named Linus Torvalds.

In 1991, Torvalds began to develop a UNIX-like kernel, driven by his frustrations with using the Dos operating system that came as standard on his PC. His initial efforts were small; the first version, based upon a scaled-down version of UNIX known as Minix, comprised less than 10,000 lines of code. Furthermore, it worked with only one type of architecture – Intel's 386. After receiving requests for copies of his new program, Torvalds posted it on an Internet software newsgroup, and soon thereafter began receiving e-mails from users containing "patches" to the bugs they had found and suggesting ideas for new features to improve the kernel. By December of that year, more than 100 people worldwide had joined the Linux newsgroup mailing list.

Exhibit 4.2 Text Extract: Red Hat – Torvalds Creates Linux (Cont'd)

In early 1992, Torvalds combined his kernel with the components that had been successfully developed by the GNU project. The world finally had a completely free operating system. Torvalds continued to direct its development with the help of an increasing number of users around the world (see Exhibit 1). As development progressed however, he quickly realized that the system was growing at a rate beyond which he could exert control. He decided to limit himself mainly to supervising changes to the kernel, and in 1993 delegated code review duties to a core group of five kernel developers, several of whom were located in other countries. The decision to do this was possible because of the modular nature of the program's design. As Torvalds described:

> it became clear very quickly that we want to have a system which is as modular as possible. The open-source development model really requires this, because otherwise you can't easily have people working in parallel ... Without modularity I would have to check every file that changed ... to make sure nothing was changed that would affect anything else.

Source: MacCormack & Herman 2000, 3.

Table 4.2 Red Hat – Torvalds Creates LINUX – Ranked Word List

Anchor Concepts	Frequency
system	9
kernel	7
develop	6
gnu	6
Torvalds	6
his	5
operating	5
program	4
change	3
development	3
module	3
not	3
project	3
this	3
UNIX	3
world	3

Source: data analysis performed using TACT (1997)

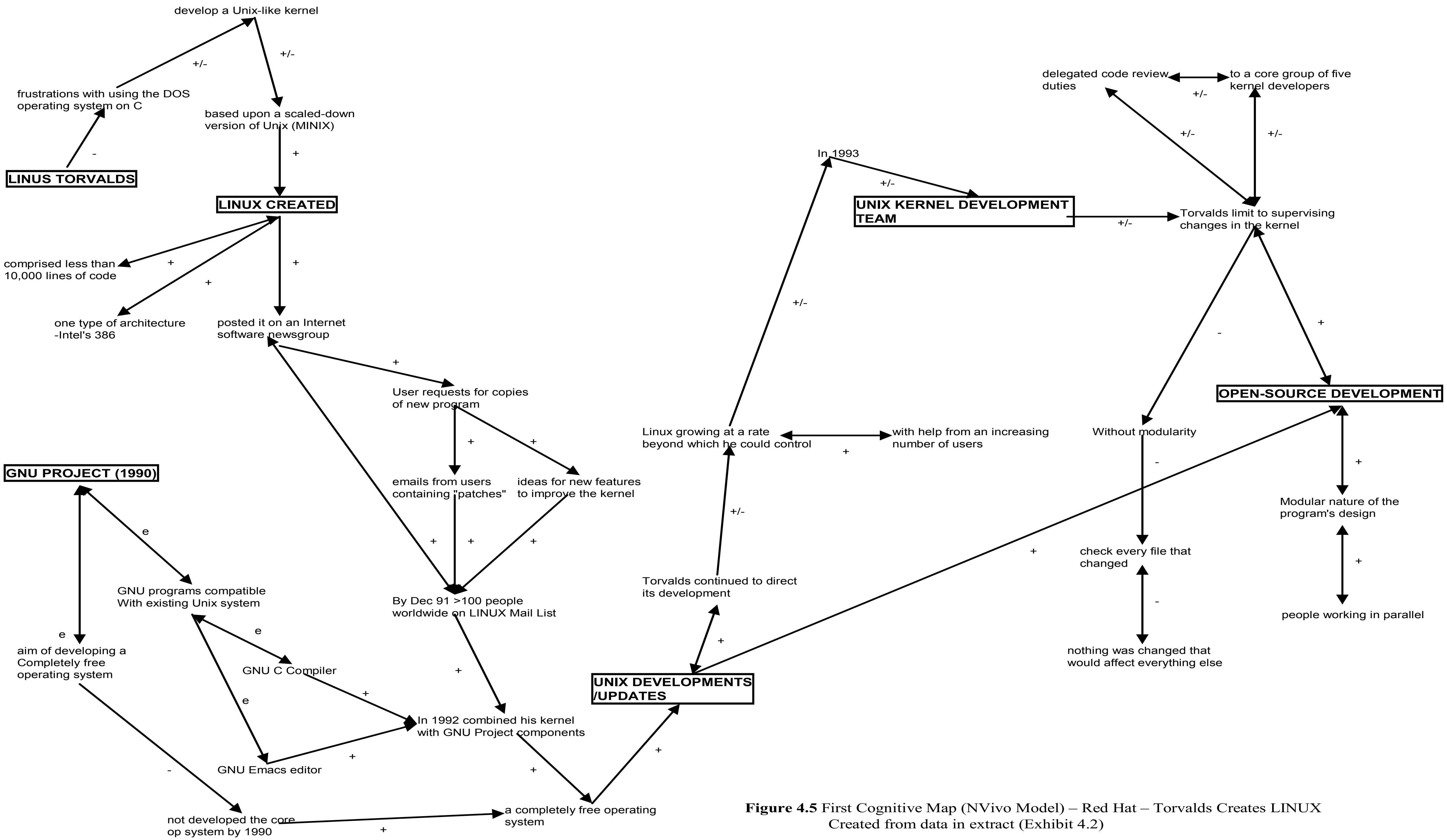


Figure 4.5 First Cognitive Map (NVivo Model) – Red Hat – Torvalds Creates LINUX
Created from data in extract (Exhibit 4.2)

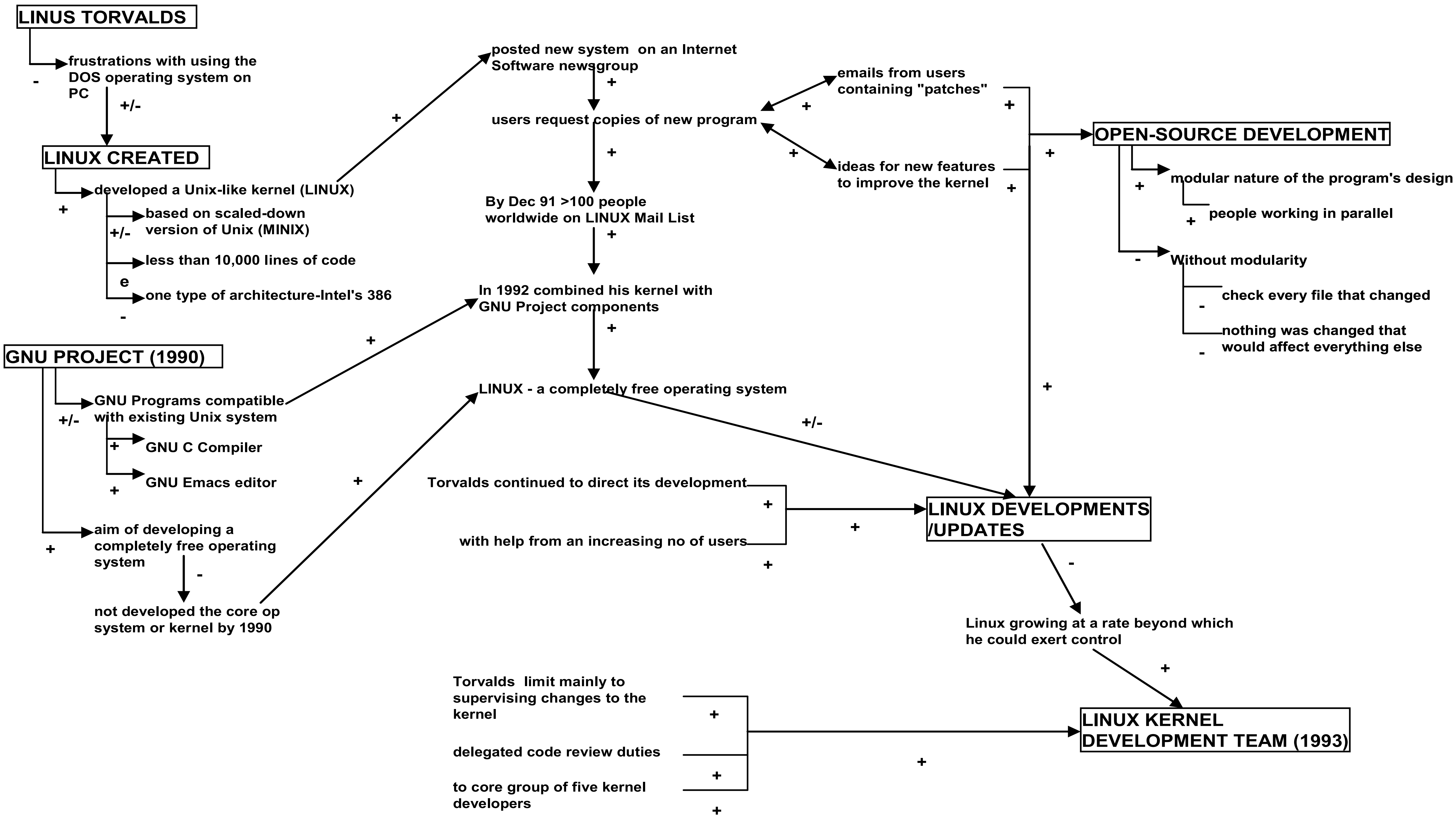


Figure 4.6 Revised Cognitive Map – Red Hat – Torvalds Creates LINUX
Translation of NVivo Map
Note. This revision was planned to be shown to participant in emic 2 stage

Linus Torvalds, a student at the University of Helsinki, was frustrated with the poor functionality of the MS-Dos operating system that drove PCs in the late 1980s and into the early 1990s. Torvalds acquired Minix, an operating system that resembled UNIX and was designed for PCs.

Torvalds used Minix as a starting point to develop his own kernel program, the basic core of an operating system. Torvald's kernel was small and only ran on PCs using Intel's 80386 (i386) processor chips and supporting chipsets. Torvalds' timing in releasing the kernel was critical, as no operating system at the single-user PC level was sufficiently advanced to provide developer tools and capabilities that UNIX could offer on graphics and engineering workstations.

IBM was at that time working with Microsoft on OS/2 which was expected to be the great new operating system. Microsoft's own Windows operating system was about to be launched into it's very successful Version 3 – but prior to V3 Windows was not visible to Microsoft's MS-Dos system.

Torvalds posted his kernel on an Internet Software Newsgroup (comp.os.Linux) in September 1991. There was substantial development interest in his program, as several users e-mailed Torvalds with actual code patches and new ideas on how to upgrade the kernel. Linux newsgroup membership exceeded 100 by the end of December 1991 – not a huge number for Internet Newsgroups, but a very energetic group dedicated to strong and direct involvement in ongoing development of the kernel program.

Meanwhile by 1990, the GNU Project group still had not developed an operating system. The way that Linux became an open-source software application highlights different ways in which software may be seen to become "open-source".

Some applications were developed specifically by members or associates of the GNU Project and gained official GPL recognition (for example GNU C Compiler and GNU Emacs Compiler).

Other applications were offered freely by their developers through newsgroups or through contacting the developer, and gained de facto recognition as "open-source" software, and not necessarily with recognition from the GNU project group.

Torvalds combined the GNU Project components with his kernel to create Linux as the free operating system the GNU Project was looking for, in March 1992. Interest and direct involvement in developing Linux into a fully fledged operating system grew rapidly through 1992, with Torvalds successfully directing development, plus coordinating ideas and directing coding inputs from an increasing numbers of developers and users.

Torvalds attributed the ongoing rapid development of Linux to the "open-source" approach where programs can be modularised such that programmers from around the globe could work concurrently on programs.

Torvalds believed that without modularity every file would have to be checked by him and everything in the overall application would be interdependent, meaning that a change in one program would require changes in other programs. A non-modular approach would be almost impossible for one person such as Torvalds to manage ongoing development.

Linux was growing so fast in 1993 in terms of actual development that Torvalds could not manage ongoing development by himself. Torvalds decided to limit his role in Linux development to supervising changes to only the kernel program. He delegated code review duties to a core group of five Linux kernel developers based around the world.

The cognitive map captures insights behind key decision-points for Torvalds in electing to develop the Linux kernel and his involvement in ongoing Linux development. Rapid development using an open-source approach suggests that delegation of vision and decision-making is required within 6 to 12 months of releasing the application.

4.3.3.1 Updated Cognitive Map 3 – Torvalds Creates Linux

The cognitive map addressing Linus Torvalds creating Linux was not directly discussed during the interview with Young (2001). However, further insights were drawn from Torvalds & Diamond (2001) and, with other comments and structural changes, were incorporated into an updated version of the cognitive map (see Figure 4.7).

Significant additional comments and insights, plus restructuring of sections, were incorporated into the updated cognitive map. Torvalds acquired a Sinclair Q-Dos PC in 1987 (Torvalds & Diamond 2001, p. 41) and an MS-Dos – based i386 based PC in January 1991 (Torvalds & Diamond 2001, p. 60) (see box 1).

As a student at University of Helsinki majoring in computers Torvalds had a vision of running UNIX on PCs – mainly to enable him to complete computing assignments on an affordable PC at home.

Torvalds ordered Minix after he bought the PC, but was disappointed with various features of the operating system, in particular its poor terminal emulation program, which Torvalds needed to dial-up university computers or to go to Internet Newsgroups (Torvalds & Diamond 2001, pp. 61-62).

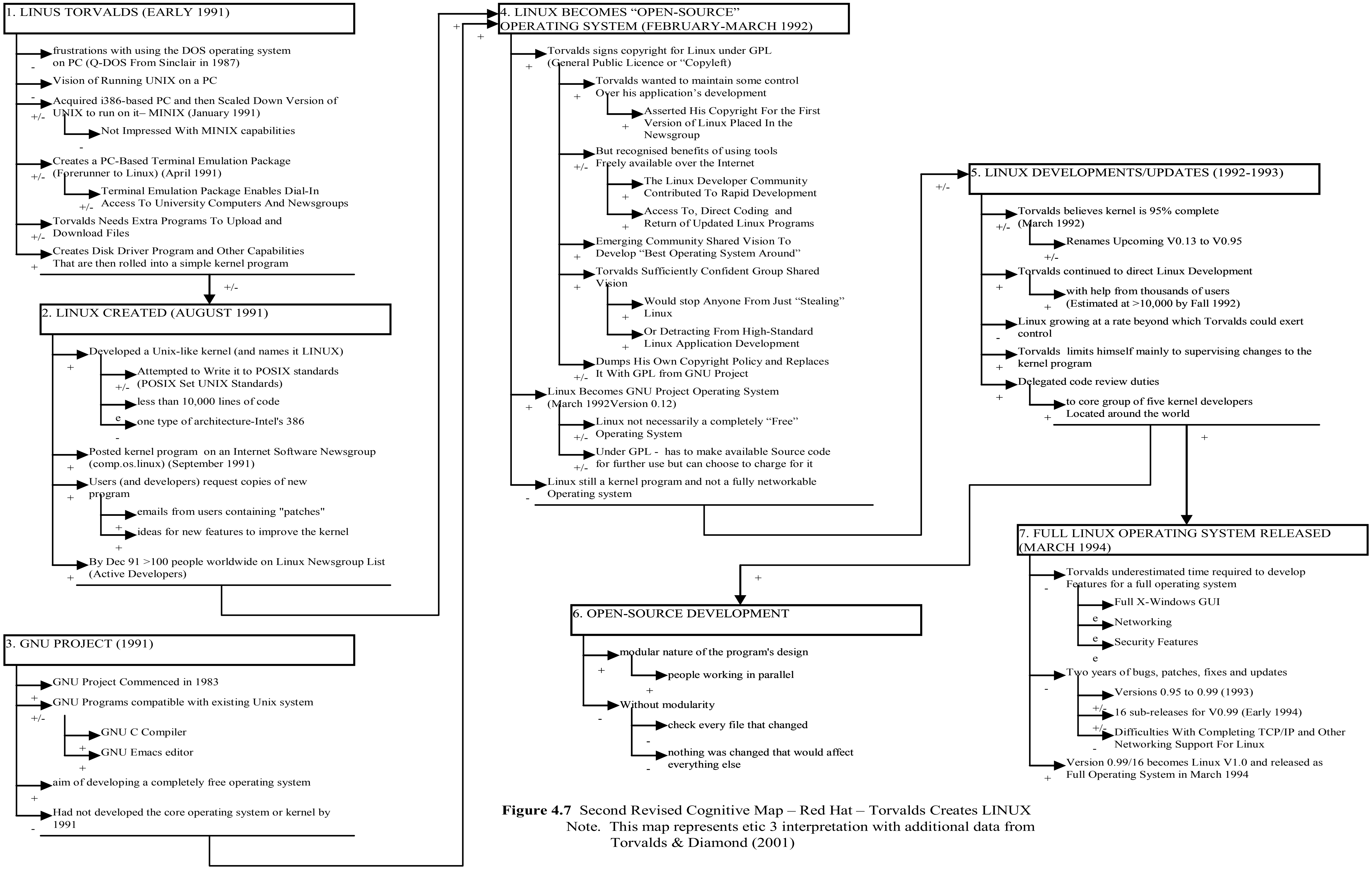


Figure 4.7 Second Revised Cognitive Map – Red Hat – Torvalds Creates LINUX
Note. This map represents etic 3 interpretation with additional data from Torvalds & Diamond (2001)

Torvalds decided to write his own terminal emulation program for his PC in March – April 1991. Torvalds used the terminal emulation program to access e-mail and to participate in Newsgroups – but he wanted to download and upload files. Torvalds needed a disk driver and other features that would be rolled together into a kernel program (Torvalds & Diamond 2001, p. 77).

Torvalds actually created the kernel program from April through to August 1991 (box 2). He attempted to code according to common guidelines so that his code could be usable by others. However, the standards set for UNIX programming were administered by the POSIX group, and Torvalds found it very difficult to find their detailed requirements, plus he believed that formal POSIX certification would be quite expensive and time-consuming (Torvalds & Diamond 2001, p. 79). Nevertheless, Torvalds found the relevant POSIX standards manuals and coded his program in accordance with those standards.

During his search for the POSIX manuals at the University of Helsinki, Ari Lemke offered to set up a subdirectory on the University of Helsinki's FTP site for Torvalds' new kernel program, thus enabling anyone interested in downloading the program (at ftp.funet.fi) (Torvalds & Diamond 2001, p. 80).

Torvalds named the kernel program Linux and after further development posted it to an Internet Software Newsgroup (comp.os.Linux) in September 1991. Comments on user (and developer) responses with patches and new ideas were carried through unchanged from the original cognitive map.

The section from the original cognitive map on the GNU Project (see box 3) was unchanged, except that the envisioned operating system had been developed by 1991, rather than 1990.

A new section was created in the updated cognitive map, addressing how Linux became an "open-source" operating system (see box 4).

A significant revision addressing Torvalds' creation of the Linux kernel as the first "open-source" operating system is incorporated into the map. Torvalds did not combine GNU Project components to create a GNU Project operating system. However, mapping Torvalds' line of thought right up to Linux becoming an "open-source" operating system unearths insights that even in late 2003 have huge reverberations for the current debates raging over the actual current and future ownership of Linux.

Torvalds faced two apparently opposite personal goals for ongoing Linux development.

On the one hand, Torvalds possessed a vision of maintaining control over ongoing Linux development as its original creator. Torvalds had asserted his copyright of Linux in the copyright policy of the very first version of the kernel put up in the Newsgroup. Torvalds wanted access to source code associated with improvements to Linux so that he could use the improvements himself (Torvalds & Diamond 2001, pp. 94-95).

On the other hand, Torvalds recognised the value of the Internet-based user community's efforts to download copies of his kernel program and to direct program updates, patches, fixes and new features, and to return the updated program back to him.

By February 1992, Torvalds was sufficiently confident that the apparent shared vision of the Linux developer community was focused on developing "the best operating system" around – and that the community would not tolerate anyone "stealing" Linux or presenting coding or actions that detracted from the shared vision (Torvalds & Diamond 2001, p. 95).

Torvalds accommodated his different goals for Linux by dumping his own assertion of copyright for Linux and signing a General Public Licence (GPL) from the GNU Project. The GPL was included with Linux Version 0.12.

It is important to reiterate that as the first GNU Project operating system, Linux was not necessarily a completely "free" operating system. Under the GPL for Linux, Torvalds had to make the program's source code available but he was permitted, if he wished, to charge for the source code.

While Linux was classified as an operating system by the GNU Project in March 1992, it was at that stage only a kernel program with virtually no user interface, no security features and no networking capability.

Comments and insights in the original cognitive map linked to "Linux Developments" and "Linux Kernel Development Team (1993)" were largely unchanged, but consolidated into an upgraded "Linux Developments/Updates (1992-93) section (see box 5).

Torvalds was confident that the next version of Linux would be 95% complete so he renamed V0.13 to V0.95 (released also in March 1992) (Torvalds & Diamond 2001, p. 117). Torvalds stayed in charge of direct ongoing Linux development, but with over

10,000 users working with Linux by the fall of 1992, it was becoming too big for one person to control. Torvalds decided to restrict his decision-making to supervising changes to the kernel program while actual code review duties were delegated to five kernel developers located around the world.

The section on "open-source development" was incorporated largely unchanged into the updated cognitive map (box 6). Torvalds believed that application development using a large online developer community was possible through using a modular program. Developers could work in parallel on program modules, and at least there could be consistent management and review of each module.

Nevertheless, if thousands of developers were working on copies of the same program, there was still considerable effort in consolidating the copies all into one compiled "build" of that program. High levels of coordination, plus rapid and frequent program builds and updates would be required to maintain a fast but consistent flow of ongoing application development.

A new section was created to reflect up to two years of additional development work required to actually produce a complete Linux operating system (box 7).

Torvalds, by his own admission, underestimated the time and effort required to incorporate features for a full operating system such as a full X-Windows – compliant Graphical User Interface (GUI), full networking capability and security features.

Torvalds' vision for Linux was for a fully functional "UNIX-like" operating system for PCs. Torvalds' logic for believing that Linux V0.13 (V0.95) was 95% of what was required for a full operating system was the incorporation of the X-Windows – compliant Graphical User Interface.

However, incorporating networking and security took up about fifty version releases (beyond V0.95 there were 20 releases, plus 26 patches in V0.99) and two years of further development work. Setting up operating system support with appropriate security levels for TCP/IP and other networking standards proved to be very difficult and took two years to sort out for Linux (Torvalds & Diamond 2001, pp. 116-117).

Torvalds finally decided that Linux V0.99/Patch 16 was at an acceptable usability level to be released officially as the full operating system Linux V1.0 in March 1994.

The updated map provides rich insights into Linux Torvalds' decision-making through from articulating a simple vision for a UNIX-like operating system for a PC in early 1991, through to achieving the vision in March 1994.

Torvalds received a lot of help from tens of thousands of developers around the world using a contrarian application licensing approach that actually provided source code to encourage them to change the programs, thus facilitating much faster application development – and a different, team-based coordination and management approach.

4.4 Extension Questions

Extension questions were asked during the interview stage covering Dominant Logic, Shared Vision, Leverage Points, and Strategic Marketing Principles.

4.4.1 Dominant Logic

The Red Hat case study highlights a broad, but variable and complex form of dominant logic setting. According to Young (2001) dominant logic within Red Hat varied by project and for each project team. Marc Ewing and Paul Cormier were noted as significant contributors for dominant logic related to technology issues in Red Hat – but even then that was only for the final assembly of all the programs into the application.

Young (2001) sees that Red Hat, as a highly visible group within the open-source community, is heavily influenced by – but also contributes to – that community's collective dominant logic.

Dominant logic is much more difficult to define in relation to the open-source community – where collectively tens of thousands of developers are working to create new and improved applications.

Comments from the cognitive map analysis on Torvalds' creation of Linux suggest that for the open-source community there are coordinators and reviewers who consolidate and integrate application development. The actual level on which these coordinators actually drive application development direction, as against creating a collective logic drawn from the community's inputs, would be worthy of further research.

4.4.2 Shared Vision

From its inception in 1995, Red Hat has focused on mainly delivering value-added tools to open-source applications. Young (2001) sees an analogy between Red Hat as an applications version of a discrete manufacturer, such as General Motors or Boeing, assembling many parts from various suppliers into a distinct product.

There are over 800 applications within the Red Hat Linux application, and Young believes that shared vision is incorporated from all of the suppliers contributing to it.

Combining a high level of third-party shared vision with open-source shared vision (which is assumed here to be the same as open-source dominant logic) with technical fabrication vision within Red Hat would produce quite a different type of vision when compared to every other case in this project.

4.4.3 Leverage Points

While leverage points were not discussed during the interview, six leverage points were identified through review of the updated events chronology map. These are outlined in chronological order:

- 4.4.3.1 GNU Project Initiated (1983)
- 4.4.3.2 Linus Torvalds Creates and Distributes Linux Kernel Program (August 1991)
- 4.4.3.3 Torvalds Signs GNU Public Licence for Linux (March 1992)
- 4.4.3.4 Marc Ewing Packages Linux with Installation and Version Management Applications (October 1994)
- 4.4.3.5 Red Hat Inc. Formed (January 1995)
- 4.4.3.6 Red Hat IPO (August 1999)

4.4.3.1 GNU Project Initiated (1983)

Richard Stallmann's response in 1983 to his inability to access software vendors' source code for modification to more effectively work in his MIS environment may well over time represent a leverage point of techno-economic revolutionary proportions.

Twenty years later, Stallmann's decision to develop the "open-source" concept (initially a freely available UNIX-like operating system with a "Copyleft" licensing system), has bloomed into a very real alternative business approach for software developers and users everywhere.

The concept will quickly move well beyond software into media content and information products and services development – and into a different way of information creation and sharing.

Within the context of Red Hat, Stallman's establishment of the GNU project set the scene for the emergence of Linux, the "open-source" community and Red Hat as the first publicly listed "open-source" company.

4.4.3.2 Linus Torvalds Creates and Distributes Linux Kernel Program (August 1991)

Linus Torvalds articulated a very similar vision to Richard Stallmann, to create a UNIX-like operating system – except that Torvalds was focused on developing it for a PC. 1991 was a time where the existing prevailing PC operating systems were text-based (no GUI), single-user and capable of running very simple applications.

The timing of Torvalds' development of Linux is critical, as Microsoft was still in the early stages of developing its Windows family of operating systems, IBM was focused on OS2 and there was a clear gap for someone to step into with a more advanced PC-based operating system.

However, Torvalds' approach to sharing the Linux kernel program using the Internet saw rapid dissemination of the program and set up an exemplar for a global open-source developer and user community. Thousands of developers and users concurrently designed coded and shared updates for Linux.

4.4.3.3 Torvalds Signs GNU Public Licence for Linux (March 1992)

Torvalds' signing of a GNU Public Licence for Linux was a symbolic milestone in that it legitimised the application development approach for Linux up to that time. GNU project endorsement of Linux positioned the application on a different dimension – it was now expected to have all the features of a fully functional UNIX operating system – which would take two more years of development.

Linux was a form of fulfilment of Stallman's original vision for an operating system, but it was the catalyst for the GNU project to move on up to more advanced software – a shift toward what would become the Open Source Initiative (OSI).

4.4.3.4 Marc Ewing Packages Linux with Installation and Version Management Applications (October 1994)

Marc Ewing could see great potential in creating software development tools for the new and emerging open-source Linux operating system. However, Ewing decided instead to focus on providing tools to make it easier to install Linux and to track updates to it. Ewing was one of the first Linux “ISVs” to place his package on CD, thereby addressing a major issue with obtaining Linux in 1994.

4.4.3.5 Red Hat Inc. Formed (January 1995)

Ewing’s package might not have proceeded further except for a strong set of potential mutual requirements. Young’s desire for a better Linux and Ewing’s need for funding to continue his package development came together in their meeting at the Fall 1994 UNIX Expo and led to the formation of Red Hat Inc. in January 1995.

Red Hat was formed with a major competitive advantage – a version of Linux plus installation and management applications all packaged on CD. Red Hat developed software engineering capabilities to enable ongoing competitive advantage with high growth to set up for an IPO five and a half years after its formation.

4.4.3.6 Red Hat IPO (August 1999)

Floating Red Hat in an IPO represented not only a leverage point for the company but for the whole open-source community. As the first ever open-source company to go to IPO, analysts focused on its unusual development and business model. Red Hat’s IPO prompted questions as to whether the “open-source” approach could be applied to application development beyond operating systems.

Looking back from late in 2003, Red Hat’s IPO and the questions asked regarding open-source extension (which were answered positively), heralded the start of a new era where open-source development moved from not just basic applications but for consideration in complete enterprise-wide solutions, for sharing of information including media and education content, and onto changing copyright and business models for a wide range of industries.

4.4.4 Strategic Marketing Issues

Three strategic marketing issues were identified from the interview with Young (2001):

4.4.4.1 Gaining Mass-acceptance for Software Applications

4.4.4.2 A New Concept if “Open-source” Marketing

4.4.4.3 Using the Internet for Concurrent Application Development

4.4.4.1 Gaining Mass-Acceptance for Software Applications

Young (2001) referred to the High-technology Life Cycle by Moore (1995), emphasising Red Hat’s sense of urgency to “Cross the Chasm” through to gaining mass acceptance both for Linux as an open-source application and for Red Hat’s packaging of it. Mass acceptance here means users selecting Linux as their preferred Web-server operating system.

Just how much the open-source community itself actually influenced mass acceptance of Red Hat's Linux is a dimension worthy of further research.

4.4.4.2 A New Concept of "Open-source" Marketing

Young (2001) identified ongoing communication between Red Hat and the open-source community as a significant marketing issue. "Open-source" community communication involves a wide range of concurrent elements denoted here as "co" – co-conceptualisation, co-design, co-engineering, co-coding, co-testing, and co-release.

There is an emerging discussion on open-source software development and its contribution to innovative and collaborative product development and delivery – see Von Hippel (2001); Von Krogh (2003); and Chesbrough (2003a).

The addition of "Copyleft" licensing to co-conceptualisation, co-design, co-engineering, co-coding, co-testing, and co-release concepts, results in a new concept of "open-source" marketing.

"Open-source" marketing requires very high levels of shared vision, shared logic and a commitment to sharing what would be traditionally regarded as revenue-able intellectual property for the purpose of creating even higher value applications, products and services created by other parties.

Further articulation and development of "open-source" marketing can be expected over the next five years as the "open-source" approach is adopted beyond the software industry.

Indeed, recent research and commentary has focused on "open-source" product development, and visions for open-source business, science, law, design, and society – examples include Rivken (2003); Goetz (2003); Chesbrough (2003b); and Tuomi (2003).

4.4.4.3 Using the Internet for Concurrent Application Development

Young (2001) highlighted the role of the Internet in concurrent application development. Offering open-source applications over the Internet to thousands of users greatly increased the speed of application development for Linux.

Red Hat as an open-source community member was able to take every new version of Linux through the Internet and package its installation and management tools with it –

and then offer their package on open-source form to gain further updates and improvements to it. Red Hat acts as an “open-source ISV” in this context.

While application development may be quite rapid using the “open-source” approach, there is still a need for coordination and actual compilation of the thousands of concurrent feeds of application updates. Development and maintenance of application “moderators” for the open-source community represents a very interesting area for future research.

4.5 Lessons for Subsequent Project Case-studies

The Red Hat case study was the first case undertaken for the project and provides some unique insights and lessons to be applied to the other five cases in the project. Revisions for subsequent cases included:

4.5.1 Digital Recording Tools for Telephone Interviews

4.5.2 Restructured Interview Process

4.5.3 Revised Cognitive Mapping Format

4.5.1 Digital Recording Tools for Telephone Interviews

Following the telephone interview with Young (2001) for the Red Hat case study and with Chandra (2001) for the Zaplet case study, new tools for recording and collecting telephone interviews were evaluated.

The Veritape system for digitally recording interviews and saving them into .Wav or MP3 formats was selected and used for almost all subsequent telephone interviews. The Veritape software application enables recording via telephone modem and represents a very efficient and cost-effective innovation, and as such represents a methodological contribution for this project.

4.5.2 Restructured Interview Process

Following telephone interviews with Young (2001) for the Red Hat case study and Chandra (2001) for the Zaplet case study (the Zaplet case study varied as there were multiple interviews with Roberts (2002) as well as the interview with Chandra), the structure of the interview process was modified such that almost all subsequent interviews were broken into two stages, with validation of the DSA model and events chronology maps covered in the first interview (usually a face-to-face interview) and the cognitive maps in subsequent interviews (usually telephone interviews).

The main reason for the modification was that the timeframe to cover the DSA models and events chronology maps took around one to one and a half hours – an acceptable timeframe for a scheduled meeting. Coverage of a whole package of models and maps would have taken around three hours if attempted in one interview which was regarded as difficult to schedule in one block.

Breaking up the sessions into multiple interviews enabled more detailed discussion and exploration of the issues and insights for each case study.

4.5.3 Revised Cognitive Mapping Format

Subsequent to the interview with Young (2001) for the Red Hat case study, formats for updating the cognitive maps after having been presented to the interviewees were modified to be more consistent in terms of numbering, nesting and sectional conventions.

These format revisions enabled final cognitive maps to be produced that were consistent for the whole project. The Red Hat case study cognitive maps were revised to the updated format during the write-up of the case study in 2003, even though they were not fully validated through the interview in 2001.

4.6 Summary

The Red Hat case study provides insights into decision-making issues associated with an emerging and quite new "open-source" application developer community.

Decision-making issues associated with software development, and specifically with the development of Linux from 1991 to 1995 and Red Hat from its inception in 1995 to 1999, were captured an advanced hermeneutic framework, incorporating an extended DSA approach.

Hermeneutic analysis for the Red Hat case study encompassed five levels and assisted in defining detailed insights from the case study. Such hermeneutic depth was remarkable, given the limitations of the study in terms of the interview and its early placement in the research project.

Extended DSA provide very strong illumination of decision-making through a systematic development and interpretation of DSA models, event chronology maps, cognitive maps and extension questions.

The interview process, while restricted to one substantial interview (Young, 2001), complemented by detailed insights into the creation and development of Linux from

Torvalds and Diamond (2001), enabled extension of the case study timeframe to analyse, in substantial detail, a revised application development cycle for Red Hat in 2001.

Key insights from the case study include:

4.6.1 Collective Decision-making Associated With "Open-source" Application Development

4.6.2 Identification and Articulation of an "Open-source" Marketing Concept

4.6.3 Decision-making and Application Development Using the Internet as a Concurrent Application Development and Delivery Environment

4.6.4 Revisions for Subsequent Project Case-studies

4.6.1 Collective Decision-Making Associated With "Open-Source" Application Development

The "open-source" developer community exhibits very high levels of collective shared vision and shared dominant or "coordinating" logic dispersed across tens of (and by late 2003 maybe hundreds of) thousands of software developers spread across the globe.

For those companies associated with open-source development, such as Red Hat, there is a symbiotic and highly interdependent relationship between them and the open-source community.

Apart from perhaps developers of the Internet itself, open-source application development represents one of the broadest; most diverse and powerful collective examples of third-party shared vision and shared logic of any business-related product or service development and delivery environments in the world.

The Red Hat case study introduces the reader to the "open-source" environment, highlighting some of the key decision-making issues associated with it – all of which represent great opportunities for future research.

4.6.2 Identification and Articulation of an "Open-source" Marketing Concept

The "open-source" approach to application development prompts a different articulation of marketing. Concurrent application engineering, design, delivery, communication of benefits and almost real-time customer interaction, including direct application modification using a "Copyleft" philosophy, challenges us to articulate a new form of marketing concept.

While not yet clearly articulated, "open-source" marketing requires very high levels of shared vision, commitment to trading some intellectual ownership to gain much higher value-added applications that are developed much faster through the "open-source" developer community.

Organisational models to coordinate activities across such a community will be an essential area for further development of an "open-source" marketing orientation to complement an "open-source" marketing concept.

Although not apparent by the end of the timeframe of the Red Hat case study (1999 with a DSA model revision for 2001), open-source application has, in 2002-03 evolved beyond basic operating systems – answering the questions raised in MacCormack & Herman's (2000) Harvard Business School Case study.

Applications such as Sun Microsystems Star Office represent a higher level of applications now being offered as "open-source" applications, with whole "open-source" e-commerce and web services applications (such as Java and Sun One) released under GPL conditions in 2003. The Linux Revolution is well underway, but the "open-source" applications era has just begun.

4.6.3 Decision-making and Application Development Using the Internet as a Concurrent Application Development and Delivery Environment

Literally from its first steps at birth, Linux has grown up in the Internet. Linus Torvalds placed his first Linux kernel program in an Internet Newsgroup.

The Newsgroup was the genesis of a Linux Developer Community that became part of the open-source developer community when Linux became an open-source application. Tens of thousands of Linux users and developers downloaded copies of Linux and made their modifications and updates – many of which were returned via the Internet for integration into a new version of Linux.

Coordination of incoming files with revisions, patches, fixes and new ideas is a critical capability for software developing in such an Internet-based environment. However, in the case of Linux and Red Hat's packaged version of Linux, the Internet represents a vehicle for quite rapid application development, deployment and delivery.

Furthermore, the Internet can provide a multimedia, interactive infrastructure (particularly through the World-Wide Web) for a whole community culture focused on rapid application development and delivery.

4.6.4 Revisions for Subsequent Project Case Studies

The Red Hat case study provided exciting insights into the emerging “open-source” application developer community. However, as the first case study for the project, it also provided an opportunity to explore upgrades in interview recording, interview structure and the format and presentation of cognitive maps for subsequent project case-studies.

4.7 Conclusions

The Red Hat case study raises and addresses a number of new and innovative theoretical and practical issues associated with mapping strategic thought for development of Internet-based applications, using the Internet as a computer-mediated development environment (CME).

Conclusions are presented in the sections below:

4.7.1 Theoretical Issues Addressed in the Red Hat Case Study

The Red Hat case study addressed mapping strategic thought associated with application development and delivery using an “open-source” approach.

The storytelling method used for analysis in the case study, advanced hermeneutic analysis and extended DSA, were effective at identifying key decision-making issues, events, and linkages.

The methodology provided mapping and validating of initial accounts of decision-making, followed up with subsequent revisions of those accounts using a systematic emic/etic representation system.

Layers of issues unearthed during the course of case study research included:

- Understanding the Internet as an exemplar of a computer-mediated environment for collective application development
- Articulation of Shared Vision and Dominant Logic across a diverse developer environment
- Conceptualisation of software houses as “fabricators” analogous to discrete manufacturers such as automobile and aircraft manufacturers.

Further exploration and consolidation of these issues stimulates consideration of a new “meta-issue” or “metatheory” of an “open-source” marketing concept.

4.7.2 Red Hat Case Study Limitations

As the first case study undertaken for the project, there were lessons learned that could be applied to rest of the project. Revisions in interview structure and processes, cognitive map formats, and interview recording systems were adopted for subsequent case-studies.

The case study was genuinely undertaken with the view that it would be completed as a full and complete case study for the project. Findings and methodologies from the case study were reported in Pattinson, Woodside & Miller (2002).

However, in practice, the case study interview process was curtailed to one long and substantial interview with one of the founders of Red Hat. Nevertheless, additional material drawn directly from the Red Hat website (Red Hat, 2003) and from Torvalds & Diamond (2001), overcame the challenges of gaining accurate details and accounts, in particular for some of the cognitive maps.

Nevertheless, the Red Hat case study would have further benefited from additional multiple interviews with multiple interviewees. Although the interview timeframe was targeted as late 2001-early 2002, given the very high visibility of "open-source" development, there may have been benefits from obtaining a snapshot of development processes in 2002-03.

4.7.3 Contributions from the Red Hat Case Study to Strategic Sensemaking

The Red Hat case study represents the first ever research project addressing mapping decision-making associated with "open-source" application development. The study offers contributions in one dimension of understanding and rendering decision-making for collective concurrent development of an application, using the Internet as a computer-mediated development environment or platform.

The study offers contributions to a second dimension in the context of decision-making associated with a software house acting as an "open-source" ISV, taking an "open-source" application and packaging it with its own applications – and then also offering it to and through the "open-source" developer community.

Findings from the Red Hat case study stimulate the researcher to package these contributions into a "metatheory" of an "open-source" marketing concept which in fact may be a very significant contribution as researchers and practitioners try to understand implications for the dissemination of an "open-source" approach in product and service development beyond the software industry.

4.7.4 Contributions from the Red Hat Case Study to Practice

At the time of writing the project (2003), the concept of “open-source” application development and use was one of the most important issues for both software houses and Chief Information Officers (CIOs)

“Open-source” development is now a major business consideration for software houses, but will become a major issue for any business that creates and distributes any form of information electronically through computer-mediated environments.

The case study provides vital insights for application developers and software house executives (and especially those in the software industry) in understanding both decision-making for application development using the open-source approach. The case study highlights in particular challenges associated with working with tens of thousands of developers on the same applications.

For those developers who choose to use the “open-source” environment to rapidly create and bring the applications to market, the cognitive maps in particular emphasise a major requirement for very strong coordination and integration skills in order to dynamically integrate and compile very high numbers of application code inputs.

The study also unearthed risks associated with managing intellectual property associated with the “open-source” approach. Although this point is a small finding in the context of the Red Hat case study, it requires further research and is a major issue for both software house executives and enterprise CIOs.

4.7.5 Red Hat Case Study Endnote

Red Hat Inc. was targeted for the project in 2001 because it appeared to represent a new and different software house using a different application development using the Internet.

Although the original set of DSA models and cognitive maps were relatively simple to create, there was only partial completion of the interview process, and some consideration was given to putting aside the case study.

However, by late 2003, considerable extra information was available through Torvalds’ own written account of creating Linux (Torvalds & Diamond 2001) and some extra material from the Red Hat website (2003).

These updates in 2003 enabled the Red Hat case study to be revised with a set of findings on the “open-source” approach that challenge researchers to ponder a new form of the marketing concept based around open-source philosophies, organisational form, and computer-mediated development environments.

In 2000 the original Red Hat case study identified Linux as a revolution, Red Hat Inc. as a new type of "open-source" company, and asked if applications beyond operating systems could be created using the open-source approach.

By late 2003 it was clear that Red Hat represented a new type of application development, that indeed quite advanced applications could be created using the "open-source" approach, and that applications developed thus represented just the beginning of a new way to create and manage information-based service and products. Indeed the Linux, or more appropriately, the "open-source" revolution has just begun.

Chapter 5: Case Study – Zaplet

Abstract

The Zaplet case study is an account of decision-making associated with the creation, prototyping and delivery of an innovative Internet-based application technology.

The account extends the concept of "delivery" in two dimensions – one, focused on commercialisation of the technology through to creation of a company – and two, focused on creation of specific software applications using the new application technology. The account is analysed using two storytelling methods – **an advanced hermeneutic** framework, and an **extended form of decision systems analysis (DSA)** incorporating **cognitive mapping**.

Key findings from the case study highlight changes in decision-making over time during commercialisation, ranging from maximising application idea generation through to focusing on the development of highly targeted enterprise-based applications.

The venture capitalist is noted as a significant contributor not only of funds, but of moderation for creating potential application ideas and recruitment of high quality development resources.

A concept of a marketing role in software houses encompassing both product management and product marketing activities is highlighted.

Finally, there may be a case for the development and marketing of integrated intra – and – inter enterprise applications using an emerging "ecosystem" philosophy.

5.1 Background

Zaplet Inc., of Redwood Shores California developed applications based on Zaplet Technology (an Internet-based application), linking e-mails to a range of server-based applications that can be viewed or activated on opening the e-mails.

Zaplet's technology is designed on Java technology, enabling the integration of Internet (Web)-based applications and the delivery of these across the World Wide Web.

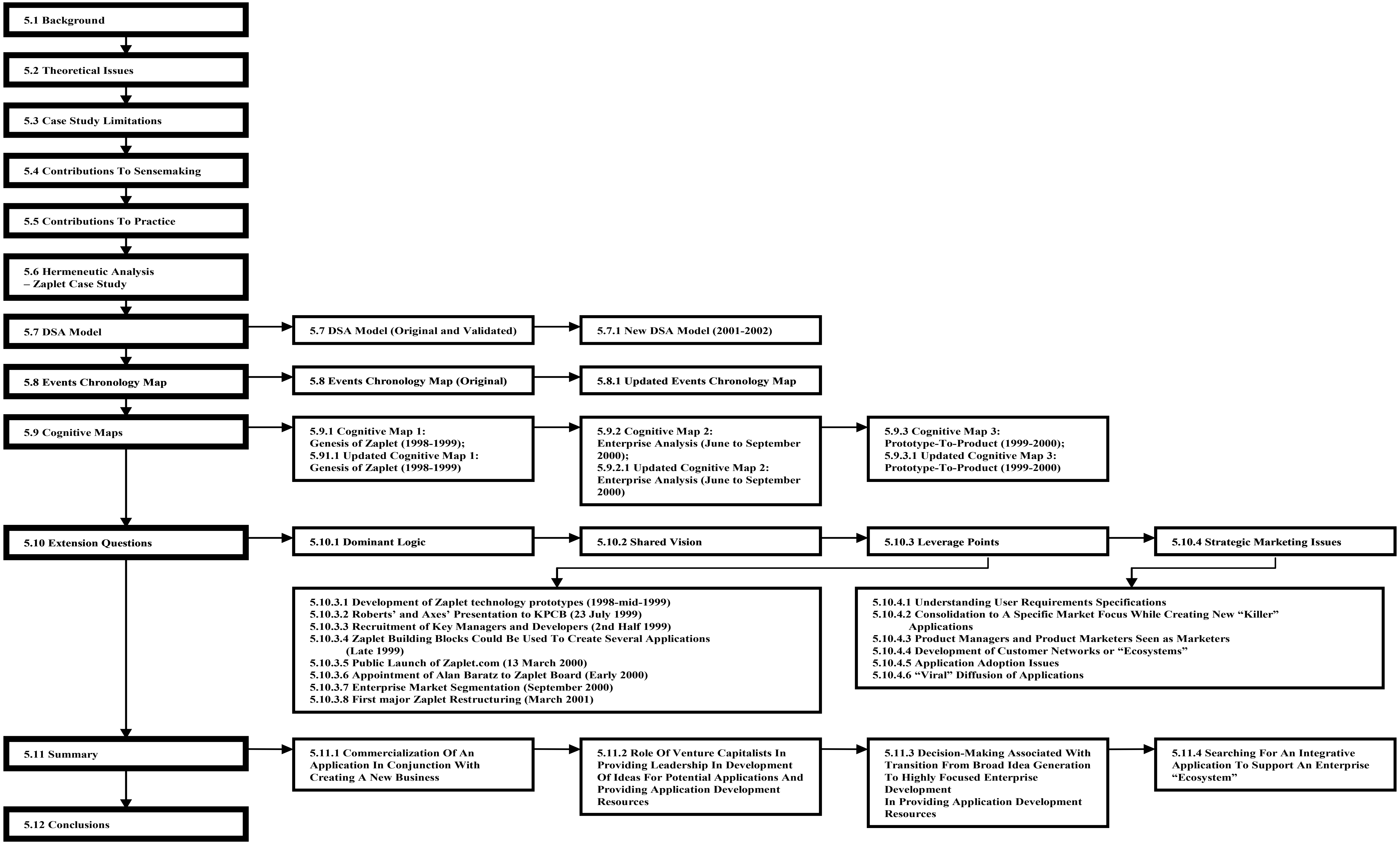


Figure 5.1 Zaplet Case Study – Chapter Structure

Zaplet was included in the final case study list for the project as it met the initial criteria:

- Software has to be "beta"-tested and commercially introduced
    - Zaplet's technology application and "building block" applications met this criteria
- Case-studies were written by well-known authors
    - The original case study is a Harvard Business School Case study supervised by Professor Dorothy Leonard, who is recognised as an expert in technology innovation, adoption and diffusion
- The application development process studied must be over a timeframe of more than 12 months
    - the original case study covers a timeframe of over two years from early 1998 through to mid-2000
- Names of case study companies and participants can be disguised if requested
    - Permission was gained from Mala Chandra and David Roberts for interviews and use of that information in the research project without requirement for disguise
- Applications must be Internet-based
    - Zaplet's technology application was based on Java – a powerful application development environment for Internet-based applications. Zaplet's applications based on Zaplet's technology were Internet-based

Additional Case study selection criteria included:

- Size: Small
- Success: Yes
- Disruptive Technology: Yes, direct enhancements to e-mail which is defined by Christensen as a disruptive technology
- Harvard Business School Case: "Zaplet Inc. (A)", DeLacey & Leonard (2001).

The Zaplet case study is an account of the parallel emergence and development of Zaplet, the start-up company and the Zaplet application technology. An advanced hermeneutic analysis, plus the Initial DSA model, events chronology map and the three cognitive maps for the case were developed from extracts drawn from DeLacey & Leonard's (2001) Harvard Business School Case study "Zaplet, Inc. (A)".

DeLacey & Leonard (2001) describe the genesis of Zaplet from 1998 through to late 2000, with an emphasis on developing concepts to take to venture capitalists to seek funding, and then after the application technology has been developed, assessing potential uses for the application.

DeLacey & Leonard's case provides sufficient description of the Zaplet application development process for development of a relevant DSA model, an events chronology map and three supporting cognitive maps.

The DSA model and maps for this case were updated, plus extension questions on dominant logic, shared vision, key leverage points and strategic marketing issues were addressed through a interview with the one of the Zaplet founders and head of development at Zaplet (see Chandra 2001; Roberts 2002).

Chapter structure for the Zaplet case study is presented in Figure 5.1.

5.2 Theoretical Issues (or Questions) for the Zaplet Case Study

The Zaplet case study addresses creation of an application technology plus identifying potential user applications that could be built using the technology. The case study also explores issues associated with the conversion of potential application ideas into working application prototypes.

The storytelling methodology used for analysis in the case study – an advanced hermeneutic analysis, including an extended form of **decision systems analysis** (DSA) incorporating **cognitive mapping** – were effective at identifying key decision-making issues, events, and linkages.

The methodology provided mapping and validating of initial accounts of decision-making, followed up with subsequent revisions of those accounts using a systematic emic/etic representation system.

5.3 Red Hat Case Study Limitations

The interview process for the Zaplet case study was limited to one long face-to-face interview (Roberts 2002) that provided complete validation and updates for the full package of DSA models, events chronology maps, and cognitive maps, plus one long telephone interview (Chandra 2001) addressing all application development – related models and maps for the case study.

There would have been additional value in further follow-on multiple interviews with both interviewees and possibly one or two additional interview respondents.

Interviews with representatives from a Venture Capitalist group supporting Zaplet were sought and were almost obtained, but dropped due to logistics issues. The Venture Capitalist perspective would have brought a different angle or set of insights into Zaplet's approach to creating ideas for its technology.

5.4 Contributions from the Zaplet Case Study to Strategic Sensemaking

The Zaplet case study explored decision-making issues, events, and linkages using the storytelling method developed for the project. An advanced hermeneutic analysis provided several levels of emic/etic representation of decision-making, highlighting key case study issues. The Hermeneutic framework incorporated an extended form of Decision Systems Analysis (DSA) highlights specific subsets of decision-making, through mapping of decision-making from original accounts with revisions, through subsequent contact and discussions with original decision-makers. The extended DSA also explored changes over time in shared vision and new application development.

The case study initially mapped Zaplet's decision-making for attempting to leverage a new application technology toward creating new user applications. Mapping highlighted strategic thinking for rapid generation of ideas for user applications – and subsequent attempts to develop prototype applications out of these ideas.

Based on inputs from the interview process, the case study mapped Zaplet's decision-making processes for an organisation focused on enterprise applications. Mapping for Zaplet in 2001-02 addressed defined application development cycles for both their application technology and for focused applications using their technology.

The Zaplet case study highlights a symbiotic relationship between new and emerging application creation, commercialisation and further development, and new business creation and development.

5.5 Contributions from the Zaplet Case Study to Practice

The Zaplet case study provides valuable insights into the development and commercialisation of a new application technology and then the challenges associated with exploring ideas for extending the technology into potential user applications.

The study highlights important contributions that venture capitalists can offer to emergent technology companies beyond funding. While funding was critical to Zaplet growing into a viable company, its venture capital company provided valuable support and assistance with management skills, encouragement to brainstorm potential application ideas, and assistance with recruitment of very high-quality application developers.

The case study also highlights how contributions from venture capitalist may change over time, and as a company restructures its business and market focus.

5.6 Hermeneutic Interpretation – Zaplet Case Study

The Zaplet case study offers an application of the proposals described in Chapter 1 for advancing hermeneutic research in B2B decision-making.

Five levels of hermeneutic analysis were developed to address decision-making issues, events, and linkages in the Zaplet case study. These levels are presented in Exhibits 5.1 to 5.5.

The original case study by DeLacey & Leonard (2001) serves as the etic 1 report of the case study – although there are elements of emic 1 reporting in the form of direct quotes from decision-makers in the account. Quotes and perspectives from decision-makers were encapsulated the Level 1 emic analysis in Exhibit 5.1.

Exhibit 5.1 Level I Analysis of Zaplet Case

Emic 1 view:
•"What's the target market?
•Should we be an applications company or a platform company?
•Do we want to focus on the consumer space or the enterprise space?
•What is the value of this technology?

1995-2000. Initial mental model:
•Several ideas for application prototypes based on new Java-based technology applied to online e-mails
•Shepherd Approach To Development – process of discovery of new ideas for applications
•Development of Management Structures

Decisions-actions:
•RDC of Zaplet technology; Startup company created in secret "stealth" mode; early VC funding plus assistance with idea generation for potential applications using Zaplet technology; VC also sought out of Java programming resources; Proliferation of Business/Product Management Units; Multi-stage Market Segmentation Analysis; New CEO and Management Structure; Zaplet refocused on Enterprise Market Segment

Key case study issues as articulated by DeLacey & Leonard are highlighted in the Level II analysis presented in Exhibit 5.2. These issues included:

- RDC of Zaplet technology focused on development potential of applications for the technology
- VCs play critical roles in funding, management support, catalysts for idea generation, and provision of application development resources
- Transition of Organisation structure from diverse small units to executive team focused on enterprise market segment
- Shortcuts in development greatly assisted Zaplet to produce more applications for more potential end-uses
- Company-wide involvement in multi-stage market segmentation analysis.

The Level II analysis also explored mental models and decisions/actions associated with these issues.

The third level of analysis, as presented in Exhibit 5.3, is an etic 2 representation based on the current researcher's summarisation of event milestones and the emic 1 sensemaking views identified in the data in the DeLacey & Leonard (2001) case study. Subsequently to developing Exhibit 5.3, emic 2 and etic 2 interpretations, along with DSA and event maps, provide revisions to Exhibit 5.3.

DeLacey & Leonard's (2001) coverage of Zaplet's development of its technology, application "building-blocks", and attempts at market segmentation provide adequate detail for the development of an initial (etic 2) representative DSA model and to undertake further analysis on application development and delivery. The initial DSA, event, and cognitive maps reported below offer details supporting Exhibit 5.3. Etic 2 perspectives are mainly based on the initial DSA model, initial events chronology map, and initial cognitive maps.

Key etic2 issues included:

- RDC via incubators and VC support provides wide range of services for start-up companies
- Phases of application and business development

- Transition between generating new ideas for potential application development and then reduction to viable application prototypes
- Building block approach to Zaplet provides strong application development capabilities
- Is “enterprise market segment” still too broad for Zaplet’s resources and capabilities?

Exhibit 5.2 Level II Analysis of Zaplet Case

Etic 1 view:
•RDC of Zaplet Technology Focused on Development Potential Applications for the Technology
•VC's play critical roles in funding, management support, catalysts for idea generation, and provision of application development resources
•Transition of Organization Structure from diverse small units to Executive Team focused on enterprise market segment
• Shortcuts In Development Greatly Assisted Zaplet to produce more applications for more potential end-uses
•Company-wide Involvement in Multi-Stage Market Segmentation Analysis

Emic 1 view:
•"What's the target market?
•Should we be an applications company or a platform company?
•Do we want to focus on the consumer space or the enterprise space?
•What is the Value of this technology?

1995-2000. Initial mental model:
•Several ideas for application prototypes based on new Java-Based technology applied to online emails
•Shepherd Approach To Development –process of discovery of new ideas for applications
•Development of Management Structures
Decisions-actions::
•RDC of Zaplet technology; Startup company created in secret "stealth" mode; early VC funding plus assistance with idea generation for potential applications using Zaplet technology; VC also sought out of Java programming resources; Proliferation of Business/Product Management Units; Multi-Stage Market Segmentation Analysis; New CEO and Management Structure; Zaplet refocused on Enterprise Market Segment

Exhibit 5.3 Level III Analysis of Zaplet Case

Etic 1 view:
•RDC of Zaplet Technology Focused on Development Potential Applications for the Technology
•VC's play critical roles in funding, management support, catalysts for idea generation, and provision of application development resources
•Transition of Organization Structure from diverse small units to Executive Team focused on enterprise market segment
• Shortcuts In Development Greatly Assisted Zaplet to produce more applications for more potential end-uses
•Company-wide Involvement in Multi-Stage Market Segmentation Analysis

Etic 2 view:
•RDC via incubators and VC Support Provides Wide Range of Services For Start-Up Companies
•Phases of Application and Business Development Transition between generating new ideas for potential Application development and then reduction to viable Application prototypes
•Building Block approach to Zaplet provides strong application development capabilities
•Is "Enterprise Market Segment" still too broad for Zaplet's resources and capabilities?

Emic 1 view:
•"What's the target market?
•Should we be an applications company or a platform company?
•Do we want to focus on the consumer space or the enterprise space?
•What is the Value of this technology?

1995-2000. Initial mental model:
•Several ideas for application prototypes based on new Java-Based technology applied to online emails
•Shepherd Approach To Development –process of discovery of new ideas for applications
•Development of Management Structures
Decisions-actions::
•RDC of Zaplet technology; Startup company created in secret "stealth" mode; early VC funding plus assistance with idea generation for potential applications using Zaplet technology; VC also sought out of Java programming resources; Proliferation of Business/Product Management Units; Multi-Stage Market Segmentation Analysis; New CEO and Management Structure; Zaplet refocused on Enterprise Market Segment

The etic 2 DSA model and maps for the case were updated following extensive questioning of the accuracy and completeness of the original DeLacey & Leonard's (2001) case – the collected data is emic 2 material and was added during the fourth level of analysis (see Exhibit 5.4).

Thus, additional (emic 2) data were collected for etic 3 description and interpretation of the Zaplet decision-making process, as reported in the original case study and for a period of two years beyond that reported by DeLacey & Leonard. Emic 2 data consist of responses from interviews with the VP of Development in 2001 and one of the founders of Zaplet (see Chandra 2001; Roberts 2002).

Exhibit 5.5 is the final level of analysis with the addition of etic 3 data summarising emic 2 interpretations of mental models and events covered in the original case study as well as the work completed for the etic 2 interpretation – including the DSA, event, and cognitive maps developed for the etic 2 interpretation.

The revised DSA, event chronology, and cognitive maps presented in subsequent sections below follow from the emic 2 interpretations, and these maps are part of the etic 3 interpretation.

The new data from the emic 2 and etic 3 rounds of interpretation validated etic 2 data and highlighted new insights related to:

- Application development intertwined with business development/start-ups
- VC's can be critical in providing management, marketing, application development, and funding resources
- Decision-making to switch from "idea discovery" mode to " market segment" mode is transformational
- How can integrative applications be developed to support an "ecosystem" of customers?

Overall, the five-level hermeneutic framework for the Zaplet case study provided a strong and comprehensive framework for capturing and extending dynamic sensemaking for Zaplet's decision-makers.

Exhibit 5.4 Level IV Analysis of Zaplet Case

Emic 2 view:
•Reaffirmed all of Emic 1 view with extensions for distinct application development cycles for the Zaplet Technology and Applications using the Technology
•Balancing Development of new "killer applications" with reduction and focus to distinct market segments
•Valuing Product Managers and Product Marketer as Marketers
•Application Diffusion Challenges
•Viewing Enterprise Customers as networks or "ecosystems"

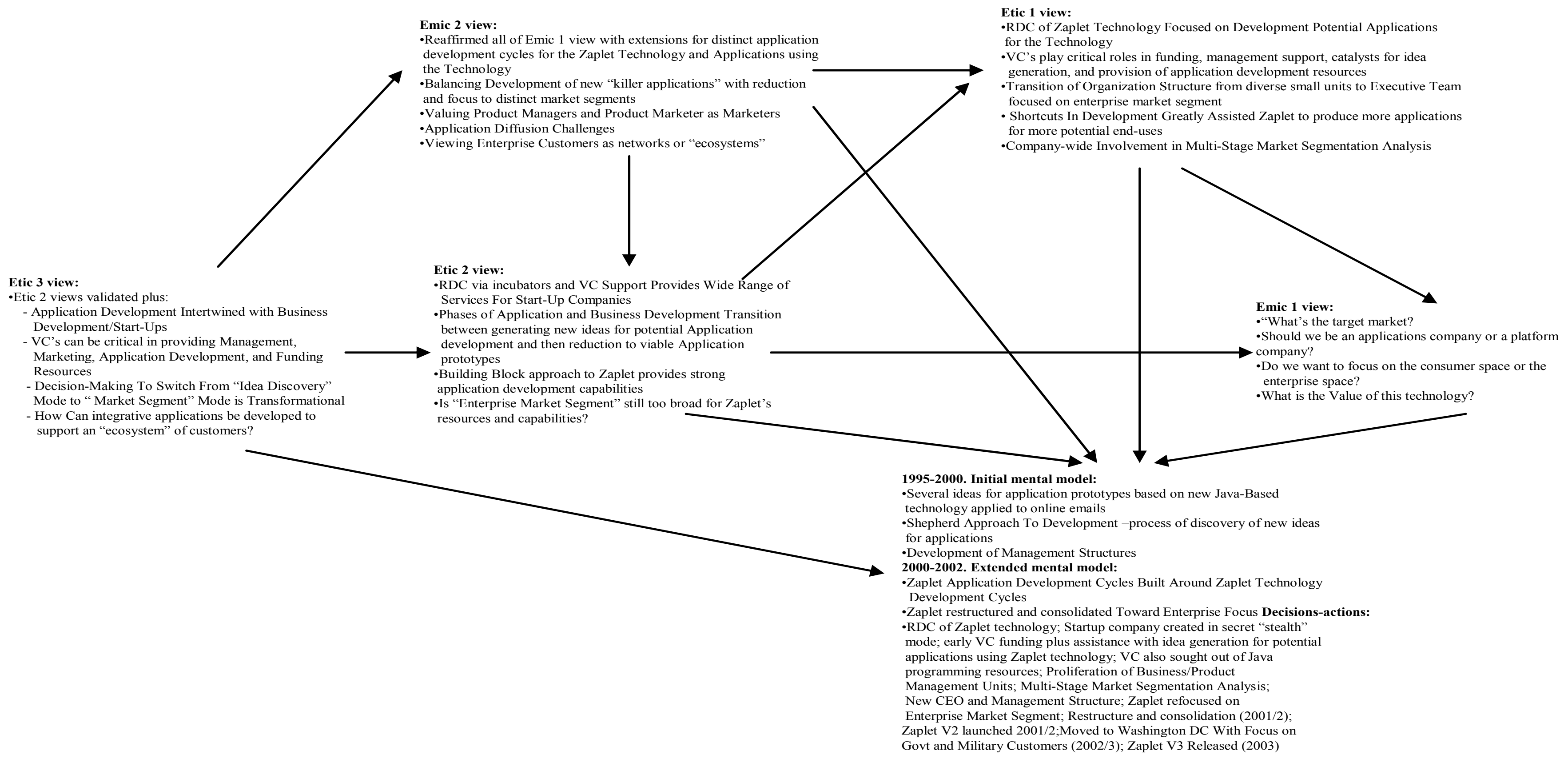
Etic 1 view:
•RDC of Zaplet Technology Focused on Development Potential Applications for the Technology
•VC's play critical roles in funding, management support, catalysts for idea generation, and provision of application development resources
•Transition of Organization Structure from diverse small units to Executive Team focused on enterprise market segment
• Shortcuts In Development Greatly Assisted Zaplet to produce more applications for more potential end-uses
•Company-wide Involvement in Multi-Stage Market Segmentation Analysis

Etic 2 view:
•RDC via incubators and VC Support Provides Wide Range of Services For Start-Up Companies
•Phases of Application and Business Development Transition between generating new ideas for potential Application development and then reduction to viable Application prototypes
•Building Block approach to Zaplet provides strong application development capabilities
•Is "Enterprise Market Segment" still too broad for Zaplet's resources and capabilities?

Emic 1 view:
•"What's the target market?
•Should we be an applications company or a platform company?
•Do we want to focus on the consumer space or the enterprise space?
•What is the Value of this technology?

1995-2000. Initial mental model:
•Several ideas for application prototypes based on new Java-Based technology applied to online emails
•Shepherd Approach To Development –process of discovery of new ideas for applications
•Development of Management Structures
2000-2002. Extended mental model:
•Zaplet Application Development Cycles Built Around Zaplet Technology Development Cycles
•Zaplet restructured and consolidated Toward Enterprise Focus **Decisions-actions:**
•RDC of Zaplet technology; Startup company created in secret "stealth" mode; early VC funding plus assistance with idea generation for potential applications using Zaplet technology; VC also sought out of Java programming resources; Proliferation of Business/Product Management Units; Multi-Stage Market Segmentation Analysis; New CEO and Management Structure; Zaplet refocused on Enterprise Market Segment; Restructure and consolidation (2001/2)
Zaplet V2 launched 2001/2

Exhibit 5.5 Level V Analysis of Zaplet Case



5.7 DSA Model

The original DSA model was developed from a thorough review of the DeLacey & Leonard (2001) case study. Sections of the case describing the types and development of Zaplet building blocks were explored further and became the main components for the DSA model. The original DSA model is presented in Figure 5.2.

Selection of new specific applications or services to be built (boxes 1 to 3) using Zaplet technology leads to a key decision-point regarding the ease of customising an existing Zaplet application to provide the new service

Where there was not an existing Zaplet application available that could be easily customised to the defined service (The "No" option for the Decision Diamond box 3), a group would be set up to decide design objectives (box 7).

The group comprises staff from the Engineering Group, Product Marketing Group and a set of "Chartered Customers" (boxes 4, 5 and 6). The group also manages and reviews the stages of application development (boxes 8, 9, 10 and 11), right up to the final design (box 11 is where the application is designated as a "Zaplet Building Block").

The Engineering Group is responsible for the design and development of the new Zaplet Building Block, with further customisation for the defined service if required (boxes 12 and 14).

Where a Zaplet Building Block is available that is customisable for the new service, the Engineering Group completes the tasks for customising the application (box 13), without necessarily having to set up a group to develop a new Building Block.

Based on the case review, during 1998-99 the Engineering Group was the major driver of Zaplet Application development with some input from a Product Marketing Group and a selected group of customers ("chartered customers").

However, it was unclear from the case as to who was in the "chartered customers" group (for example, were they classic "lead-users?), and the level of input they contributed to Zaplet application development – and how much Zaplet adapted building to meet their specific requirements.

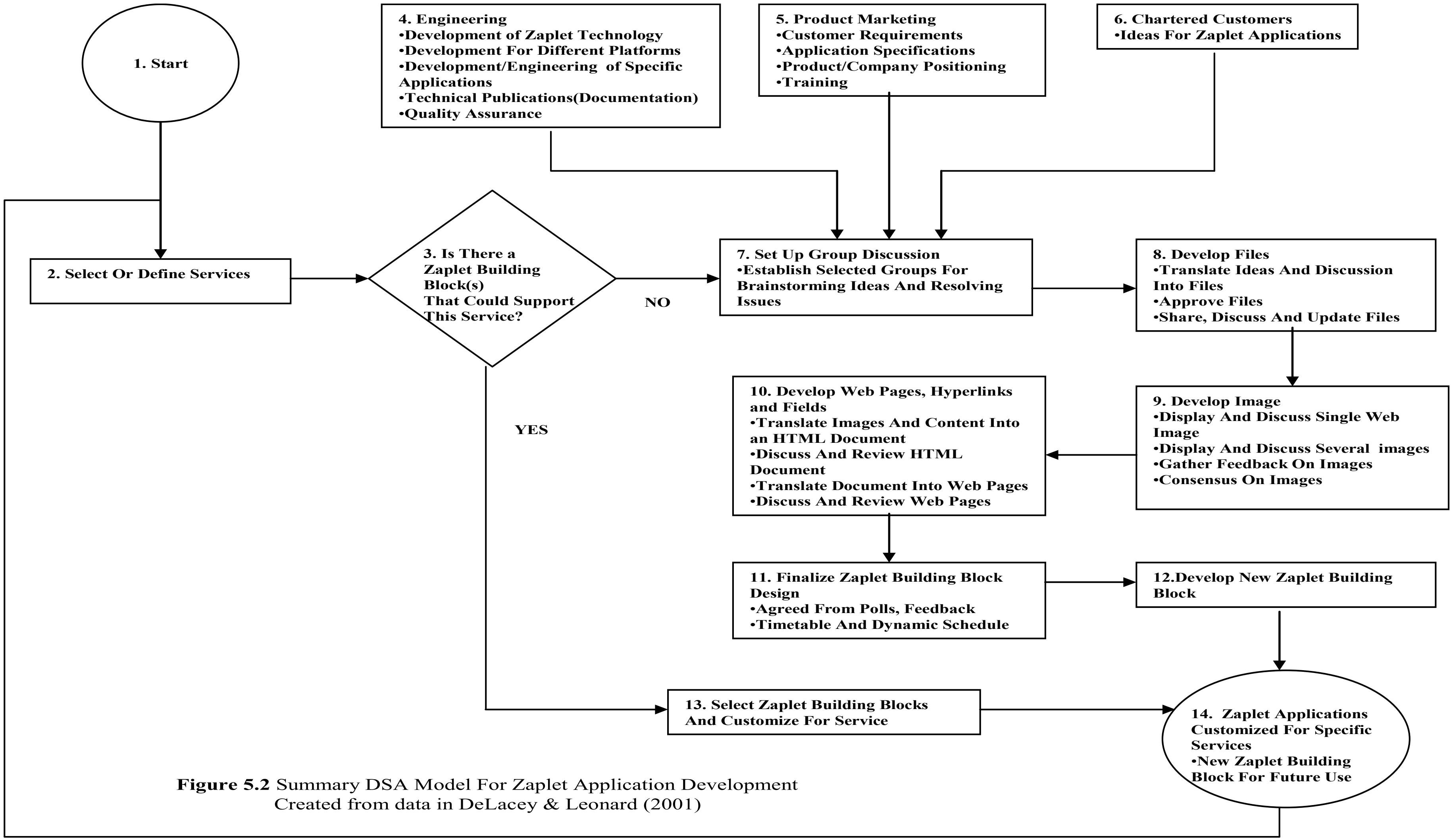


Figure 5.2 Summary DSA Model For Zaplet Application Development
Created from data in DeLacey & Leonard (2001)

5.7.1 New DSA Model

Interviews with Zaplet executives (Chandra 2001; Roberts 2002) highlighted the need to create a new DSA model to reflect application development cycles in 2001-02.

A vignette capturing the development of Zaplet technology and applications was written based on extracts from the interview with Roberts (2002). The vignette contains sufficient detail to describe the development processes for the first three versions of the Zaplet technology platform (see Exhibit 5.6).

Exhibit 5.6 Text Extract: Zaplet – Zaplet Technology Development

ZAPLET APPLICATION PLATFORM DEVELOPMENT
Zaplet technology was originally developed by Brian Axe in 1999, while he was at Reactivity. The first official version of Zaplet (V1.0) was never officially released but was recognisable at about the time of the launch of Zaplet.com (13 March 2000). Zaplet V2.0 was released in March 2002, and Zaplet V3.0 is currently under development.

Zaplet technology is designed to:

- Run on and across virtually all commercial desktop platforms, e-mail programs, and web browsers
- Deliver lightweight applications to e-mail and the web without added IT infrastructure
- Offer a fully interactive experience with current generations of HTML-enabled e-mail.

Zaplet technology is a platform developed around J2EE interfaces and Java technology standards. Specific Java technology utilised by Zaplet includes:

- Java Database Connectivity (JDBC) to access the database
- Enterprise JavaBeans (EJB) to encapsulate business logic
- JavaServer Pages (JSP) to handle dynamic HTML generation.

The software development process for Zaplet V1.0 as described by David Roberts, took around twelve months and was separate from the software development process for the Zaplet Building Blocks. Three steps were identified in the process:

1. Initial Planning and Design
2. Coding and Development
3. Release

1. Initial Planning and Design

Initially, Brian Axe developed application prototypes of the Zaplet technology with programmers at Reactivity. As the Zaplet organisation developed under the FireDrop umbrella, further development of the Zaplet technology was driven by a development group including the founders, Axe and David Roberts, core engineering representatives and the early product managers and product marketers.
Some objectives for the new technology had been set by Axe while developing early prototypes but were extended with the development group to incorporate the following design objectives:

- Running on and across virtually all commercial desktop platforms, e-mail programs, and web browsers
- Delivering lightweight applications to e-mail and the web without added IT infrastructure
- Offering a fully interactive experience with current generations of HTML-enabled e-mail.

The development group filtered key design inputs such the technology architecture and features. The Zaplet technology was developed for UNIX and Windows operating system environments, with SOLARIS as the UNIX environment and Windows 2000 as the Windows environment.

Exhibit 5.6 Text Extract: Zaplet – Zaplet Technology Development (Cont'd)

Axe's development of the early Zaplet technology prototypes using Java technology anchored the Java technology as the preferred architecture for Zaplet, with J2EE interfaces. Java technology provided components that would enable Zaplet to develop a powerful set of features into its technology including:

- Java Database Connectivity (JDBC) to access databases
- Enterprise JavaBeans (EJB) to encapsulate business logic
- JavaServer Pages (JSP) to handle dynamic HTML generation.

In later versions of the Zaplet technology (Version 2.0) additional features such as collaboration applications, event & condition features and links to ERP applications were filtered by the development group. Most application features were developed using Java technology.

2. Coding and Development

The core engineering group developed the Zaplet technology, including the application coding. For Zaplet V1.0 the first prototype was developed in six weeks. According to Roberts this was a full prototype with all the features as requested by the development group!

The alpha version of the technology was subjected to testing by a set of QA Engineers in India. The QA Engineers completed "shallow coding" or bug fixes to the technology during testing.

There was not a formal set of alpha and beta iterations in the development of Zaplet V1.0, but selected or "chartered customers" that is, some of those developers or ISVs interested in Zaplet's technology were involved in some testing of the beta version of the technology. In Zaplet V2.0, "chartered customer" involvement in beta testing was more formalised than with the first version.

3. Release

There was no formal full release date for Zaplet V1.0, but the technology would be cleared by the core engineering group to be ready for extension to and accommodation of Zaplet Building Blocks. The focus of application development would switch to the selection and development of the Building Blocks. This switch of focus also meant that the approximate twelve-month Zaplet technology cycles were not continuous, but punctuated with the Zaplet Building Block development process. However, there was a formal release of Zaplet V2.0 in March 2002 and there is expected to a formal release of a future Zaplet V3.0.

This vignette was prepared from: Roberts, David, (2002), Personal Interview (face-to-face) with Hugh Pattinson, Redwood Shores, CA. (20 February).

However, Roberts' (2002) account of application development emphasises a need to separate development cycles for the basic Zaplet technology and applications built using the technology.

Consequently two new DSA models were developed – one addressing Zaplet Technology Platform Development (see Figure 5.3) and the other focusing on applications using Zaplet technology (see Figure 5.4).

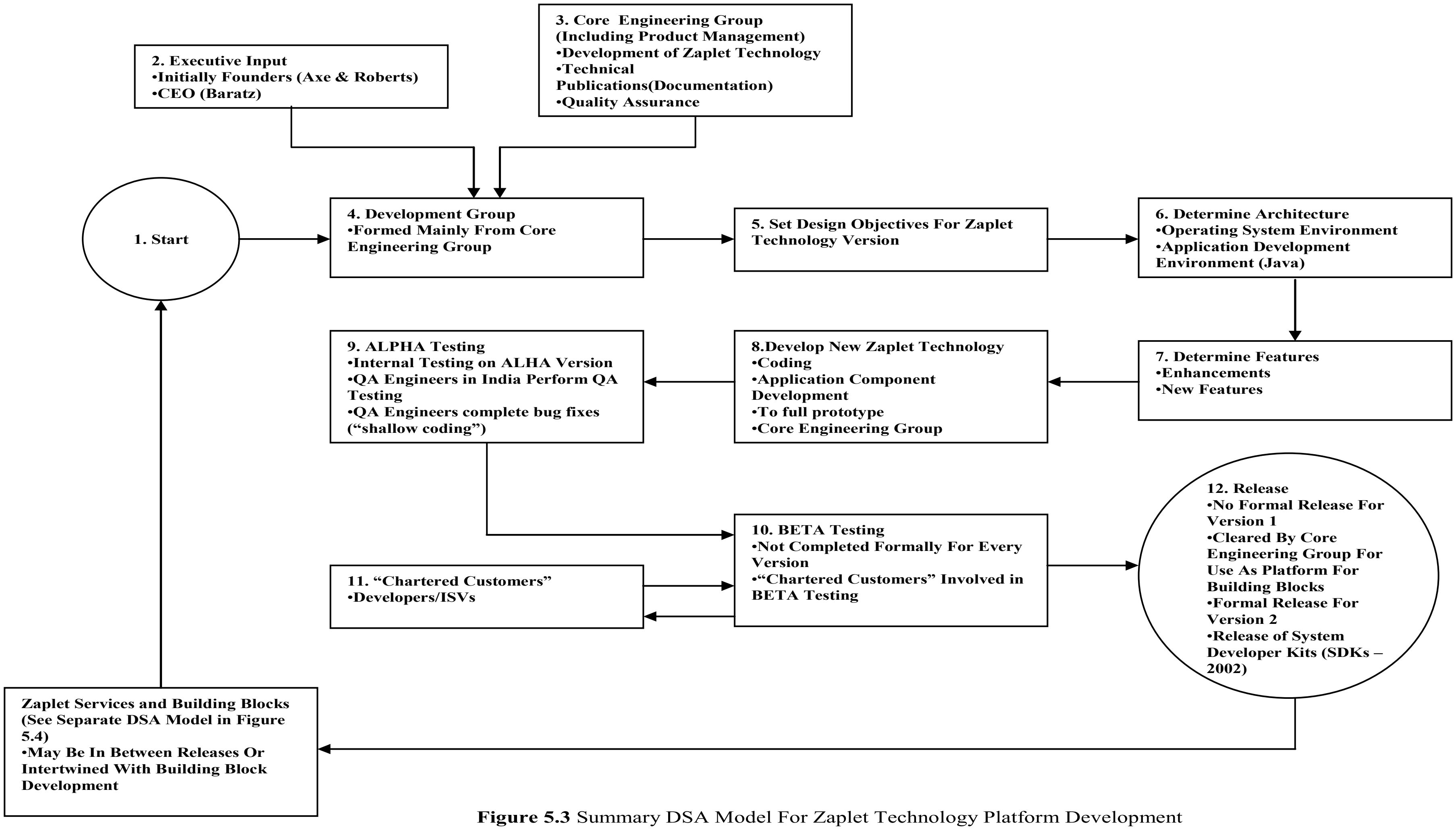


Figure 5.3 Summary DSA Model For Zaplet Technology Platform Development
Created from data in DeLacey & Leonard (2001) and Roberts (2002)

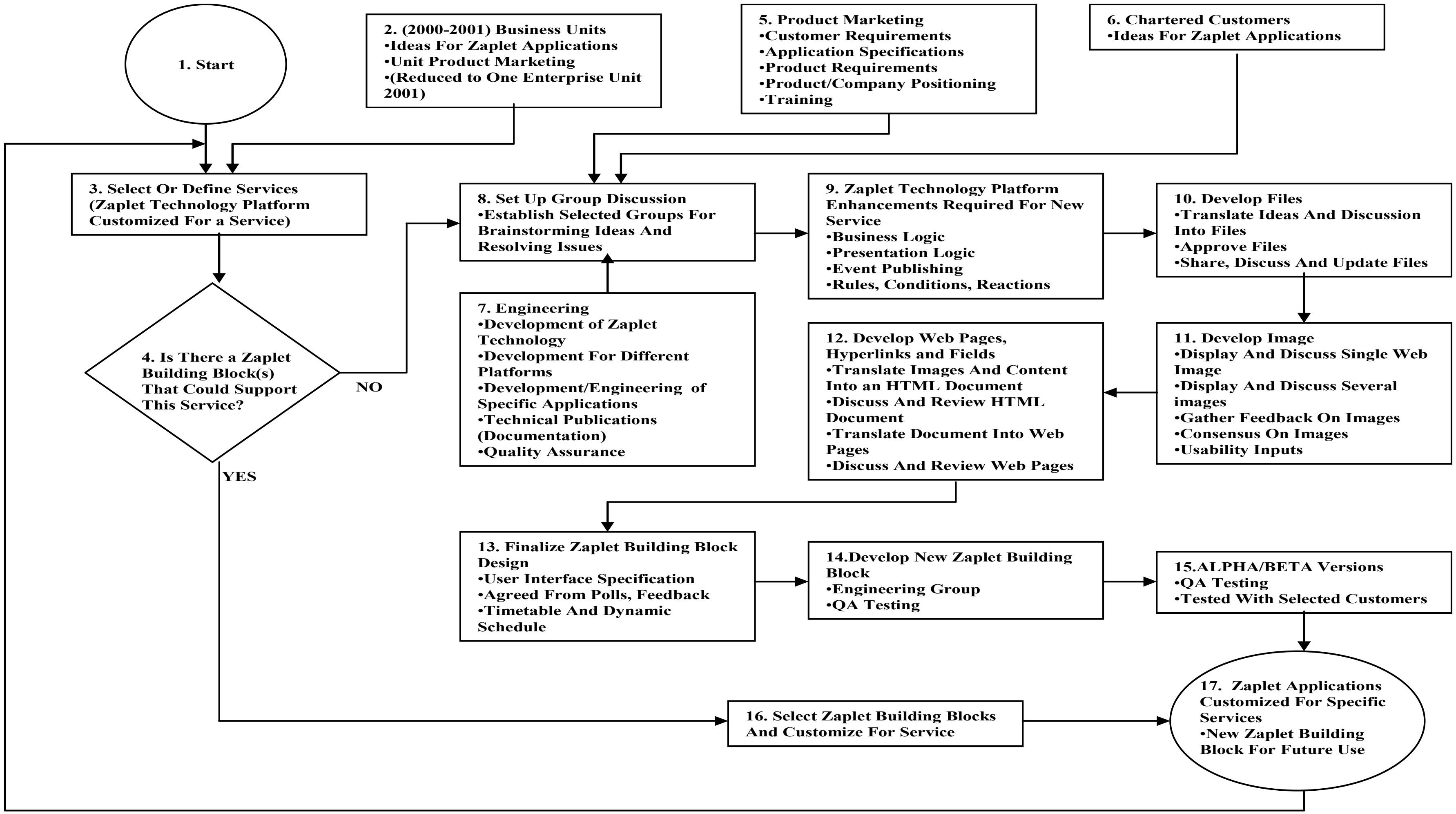


Figure 5.4 Updated Summary DSA Model For Zaplet Application Development
Created from data in DeLacey & Leonard (2001) and Roberts (2002)

The Development Group for the Zaplet Technology Platform consists mainly of engineering staff but has direct executive input from the company founders and the current CEO. As of mid-2002 there was little input to the technology platform development from marketing staff.

The actual development process for the technology platform seems to follow a conventional waterfall software process through setting design objectives (box 5), architecture selection (box 6), feature determination (box 7), development of prototype (box 8), alpha and beta testing (boxes 9, 10 and 11) and release (box 12). However, some of these processes (such as alpha and beta testing) were not formalised until Version 2 of the technology platform.

Alpha testing is conducted internally with QA testing undertaken by a group of QA engineers in India. “Chartered Customers” become involved with development at the beta testing stage.

“Chartered Customers” are defined as a selection of individual developers or ISVs or others who may be interested in new applications. “Customers” were not corporations or organisations testing the applications for a specific end-user set of tasks.

A formal release date for the first version of Zaplet technology was never officially declared. Indication of the first release was clearance given by the Core Engineering Group for the technology to be used as a platform to develop Zaplet applications. There were formal releases for Version 2, Version 2.5 and for Version 3 (released 16 July 2003).

In 2001-02 Zaplet decided to release System Developer Kits (SDKs) for the Zaplet technology platform and for some of its applications. Zaplet believes that SDKs will enable developers (ISVs in particular) to come up to speed more quickly with using the applications and to incorporate them into their own applications.

Selection of architecture for the Zaplet technology platform architecture is a key decision in the application development cycle (box 6). Java application technology was used in the development of the first prototypes of the Zaplet technology platform while Brian Axe was at Reactivity in 1998.

Selection of Java enabled access to a highly talented group of programmers and developers. The Java environment also contains a powerful set of tools and applications which Zaplet could incorporate into its technology platform.

The Zaplet technology platform may then be applied to the development of specific applications, denoted as “services” or “Zaplet Building Blocks” (box 13). A separate DSA model addressing Zaplet Application Development was created and is presented in Figure 5.4.

Business units within Zaplet (box 2) provide decision inputs for selection and definition of applications or “services” (or, in Zaplet’s jargon, “Building Blocks”) that could be built using Zaplet technology (box 3). Where there are already existing “Building Blocks”, Zaplet’s Engineering may undertake some customisation (box 16), producing a customised Zaplet Building Block (box 17).

Where there is no existing Building Block that could be customised to provide the required solution, then group discussions are set up to brainstorm ideas and issues associated with creating a Building Block (box 8).

A wide range of people contribute to these discussions, including Product Marketing (box 5), Chartered Customers (box 6), and the Engineering Group (box 7). Chartered Customers have much more input to development of Building Blocks than to basic Zaplet technology.

Specific enhancements to Zaplet’s technology platform are defined (box 9) and then developed into prototype files (box 10), images (box 11) and WebPages (box 11). These files are consolidated into a Building design specification and development plan (box 12).

The actual Building Block is developed by the Engineering Group, and it also conducts some QA Testing (box 14). The QA Group becomes more involved testing at the alpha/beta version stages, where the Building Block is also tested with a subset of Chartered Customers (box 15).

The final Zaplet Building Block is released, but is also set up as Building Block that could be used for further application development (box 17)

The new DSA models unearth different decision-making processes for Zaplet technology development when compared to Zaplet application or “services” (or “Building Block) development that was not obvious in the original DSA model.

Interviewees were asked to nominate the type of application development processes used by their organisation. Zaplet employs a mix of waterfall (or linear) and spiral development process. The Zaplet technology platform is based on waterfall processes, but development of Zaplet applications (especially early prototypes) is based on developing several prototypes at a time (spiral development process).

The two new DSA models highlight a larger and more diverse range of inputs to decision-making associated with creating new Zaplet applications. Selected aspects of Zaplet's application development cycles are explored further through the creation of an events chronology map and relevant cognitive maps.

5.8 Events Chronology Map

DeLacey & Leonard (2001) chronicled the emergence of Zaplet from an idea in 1998 through to an established company facing important business development issues in September 2000. There is a timeline exhibit in the case study covering essential events from November 1998 through to July 2000 (see DeLacey & Leonard 2001, p. 27).

An events chronology map covering the period 1995 to July 2000 was created (see Figure 5.5). Some additional symbols were employed for the map including the flowchart symbol for a delay or problem and a six-pronged box for a solution to the delay or problem. Events in this map were derived from reading and reviewing the whole of DeLacey & Leonard's (2001) case, not just the timeline exhibit in their case.

In the original events chronology map Brian Axes identifies a problem with Engineers preferring to use e-mail rather than accessing networked discussion. This problem seemed to be articulated while Axe was at Golfweb from 1995-98 (boxes 1, 2).

Axe articulates a need for an e-mail application that has full Web functionality within it (box 3). In 1998, Axe joined Reactivity to further develop his idea (box 4).

Reactivity was an incubator that encouraged the commercialisation of ideas into new start-up companies. DeLacey & Leonard (2001) define an incubator as "An organisation that helps start-ups develop in an accelerated fashion by providing them with a bundle of services, such as physical space, capital, coaching, common services, and networking connections" (DeLacey & Leonard 2001, p. 2).

Axe developed the Zaplet concept in 1999 (box 5) while at Reactivity, and wanted to develop a business around the concept. David Roberts commenced full-time work on a business plan for the Zaplet concept in 1999, which was ready to be presented to venture capitalists for scrutiny in July 1999 (box 5).

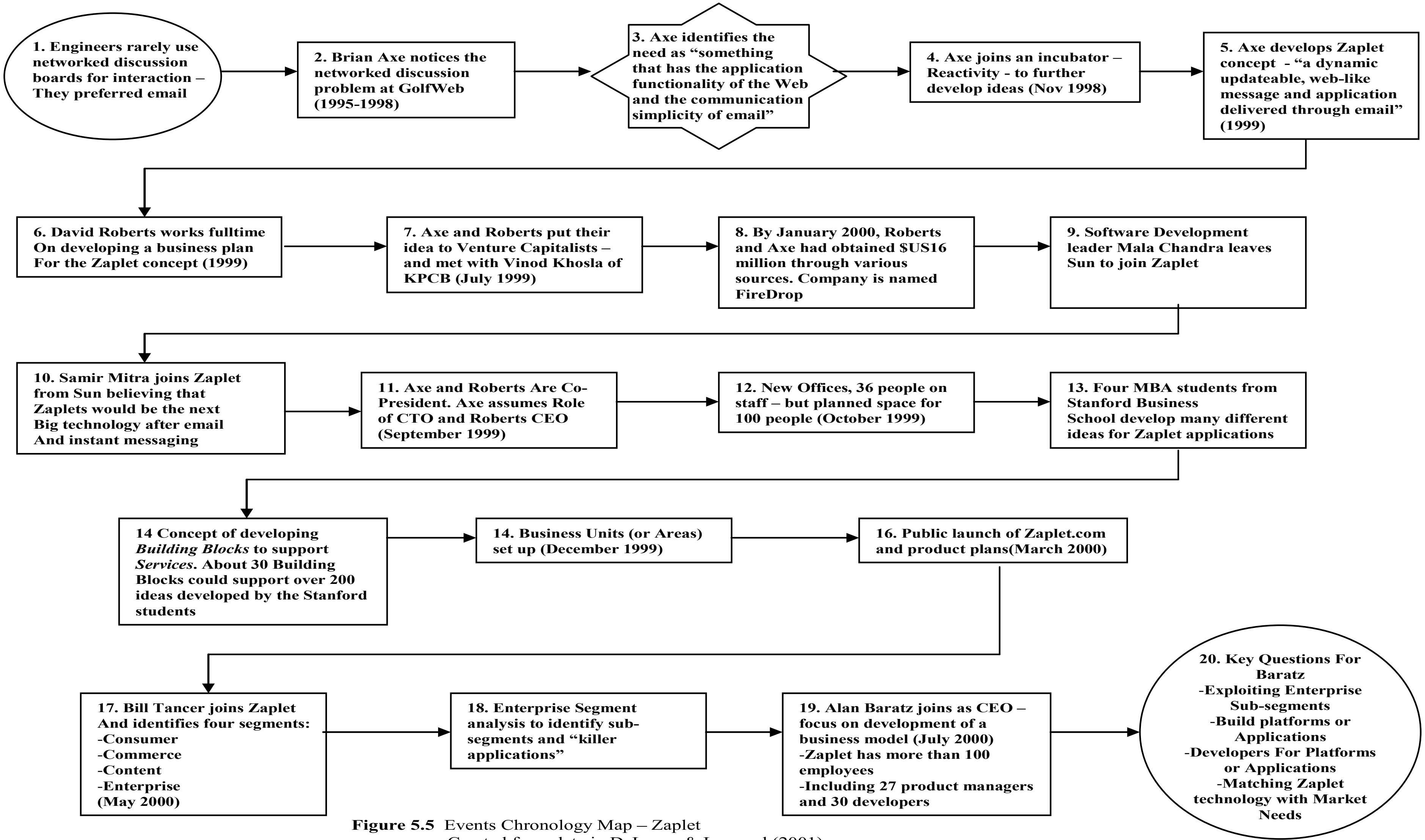


Figure 5.5 Events Chronology Map – Zaplet
Created from data in DeLacey & Leonard (2001)

Axe and Roberts gained an appointment with Vinod Khosla at Kleiner Perkins, Caulfield & Byers (KPCB) on 23 July 1999 (box 7). The appointment and connection with Vinod Khosla were critical to the development of both Zaplet as a start-up company and for the development of its applications.

KPCB was excited by the Zaplet concept and agreed to provide Venture Capital support for the new concept. The new start-up vehicle was tagged “FireDrop”.

Over the next four to five months with facilitation from KPCB, Axe and Roberts were able to obtain $US16 million in funding. They used the funds in assemble an application development team (including Samir Mitra and Mala Chandra who were prominent application developers from Sun Microsystems), and moved into new offices at Redwood Shores (boxes 8, 9, 10).

Axe and Roberts were co-presidents up to September 1999, but then, according to DeLacey & Leonard (2001), roles and responsibilities were changed, with Axe assuming a Chief Technical Offer role and Roberts being appointed CEO.

Khosla actively encouraged the exploration of possible uses for the Zaplet technology. Four Stanford MBA students were bought in ostensibly to define intellectual property for Zaplet, but also to develop a list of ideas for Zaplet applications (box 13).

The list of over 200 ideas was reviewed by the Engineering Group, which discovered that it could potentially create applications to meet most of the ideas through the development of about 30 specific applications or Building Blocks (box 14).

During December 1999, six Business Units were set up to focus on the ideas generated by the Stanford students (box 15). Zaplet was launched with this structure as Zaplet.com on 13 March 2000 (box 16).

According to DeLacey & Leonard (2001), during 2000, Bill Tancer joined Zaplet and initiated a two-stage market segmentation analysis. For the first stage, four segments were identified (box 17). For the second stage the four segments were reduced down to just the Enterprise Segment for further analysis to identify sub-segments and “killer applications” (boxes 18).

DeLacey & Leonard (2001) finalised the case study with the appointment of Alan Baratz as CEO in July 2000, plus a discussion on key questions facing Baratz as he assumed the new role at Zaplet (boxes 19 and 20).

5.8.1 Updated Events Chronology Map

The original events chronology map was updated through the interview stage of the project and is presented in Figure 5.6. There were significant changes to the original map in details, insights and chronology.

Roberts (2002) highlighted not only the technical/business issue, that is, engineers rarely used networked discussion boards, but also that there was a social issue of how collective decisions could be made among friends. This point was added to the event map, encapsulated in a problem box (box 1).

The sequence of events describing identification of the problem, then the need and Axe's move to Reactivity (boxes 3, 4 and 5) were unchanged from the original map.

However, there was additional insight into the development of the Zaplet idea at Reactivity (box 6). As noted in the commentary on the original events chronology map, an incubator provided a bundle of services to encourage the development of start-up companies. Reactivity provided Java programmers who assisted with development of early Zaplet prototype applications in mock-up form.

In one sense the incubator as exemplified by Reactivity was a "pre-VC vehicle" but with some services that a venture capital organisation might provide to a start-up company that might approach them directly.

The fact the Zaplet concept had been developed in an incubator such as Reactivity may have greatly improved the appeal – and pre-qualification – of Zaplet to a venture capital organisation.

The sections in the map (boxes 7 and 8) covering the development of a business plan and meeting Vinod Khosla remain unchanged but the amount of funding gained from July 1999 to January 2000 was updated after the interview stage to reflect a two-stage loan arrangement of $US 5 and $US 7 million for a total of $US 12 million (box 9).

Some job titles were updated or added after the interviews including Mala Chandra (VP, Engineering Management – box 10) and Brian Axe (VP, Product Development – box 12). Steve Evans was identified as Chief Technology Officer and not Brian Axe. The process of developing the list of Zaplet ideas was actually initiated by Brian Axe (boxes 14 and 15), but this was highlighted in the relevant cognitive map and not updated on the events chronology map.

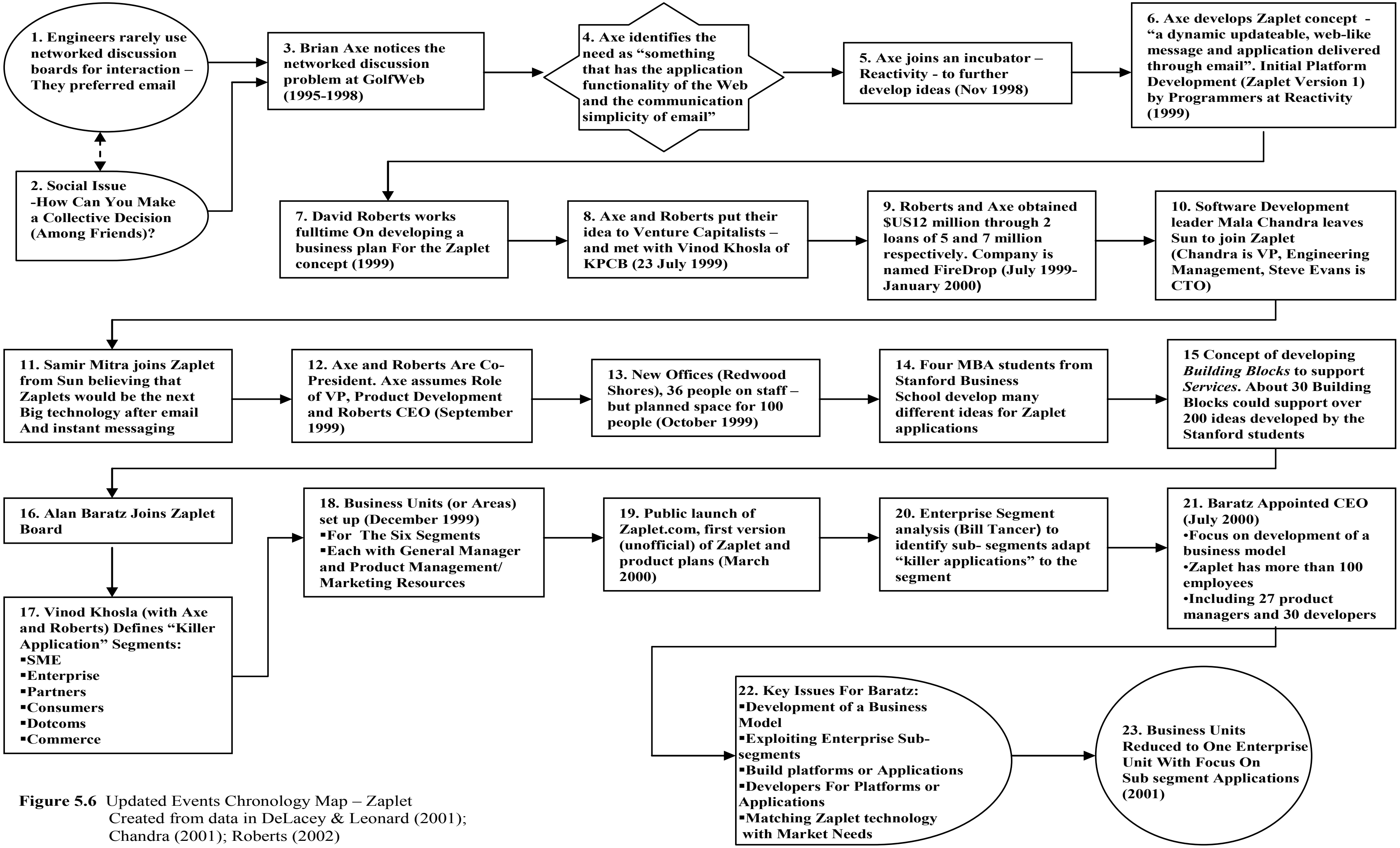


Figure 5.6 Updated Events Chronology Map – Zaplet
Created from data in DeLacey & Leonard (2001); Chandra (2001); Roberts (2002)

Alan Baratz's involvement in Zaplet was significantly updated through the interview stage. Baratz joined Zaplet's Board of Directors in late 1999 (box 16) and was actively involved in identification of Business Units, the public launch of Zaplet and the enterprise market Analysis before he was appointed CEO in July 2000.

Baratz worked with Khosla to define the specific "killer application" segments which became the six business units (boxes 16, 17, 18), that is, SME, Enterprise, Partners, Consumers, Dotcoms and Commerce.

The events chronology map was extended to include the reduction of the six business units into one Enterprise Business Unit in 2001. The Map covers the development Zaplet Version 1 (which was unofficially released) but was not extended to cover the releases of Version 2 and 2.5 in 2002 (Version 3 was released in July 2003).

5.9 Cognitive Maps

All cognitive maps created for the Zaplet case study were designed to complement DSA models and events chronology maps through additional exploration of selected decision-making issues and contexts.

All cognitive maps were created using the following procedures:

1. Selected extracts from DeLacey & Leonard's (2001) case study were extracted to text files
2. One text file version of each extract was processed through the TACT package to produce a complete word-frequency list
3. Word frequencies were reduced through one or two iterations to ranked lists of words with frequencies ranging from over one to five words
4. A second version of the text file was input to NVivo
5. Ranked words were input to NVivo and coded as nodes in association with the text file
6. The overall extract was deconstructed into selected phases and sentences and coded directly as nodes through the InVivo tool within NVivo
7. The nodes associated with each extract were input to the Model Explorer and mapped using links incorporating codes of association plus addition node "items"
8. Several iterations of dynamic mapping were undertaken with the final cognitive map exported from NVivo to PowerPoint

9. Each PowerPoint version of the cognitive map was reformatted into a more readable and project-consistent format
10. The revised PowerPoint cognitive maps were presented to interviewees and validated or modified
11. The updated PowerPoint cognitive maps incorporated changes after the interview process.

Three sections were extracted from DeLacey & Leonard (2001) and subjected to further text analysis and cognitive mapping to enhance insights derived from the DSA model and events chronology map.

The three cognitive maps created for the Zaplet case study addressed the following themes:

5.9.1 Genesis of Zaplet (1998-99)

5.9.2 Enterprise Analysis (June to September 2000)

5.9.3 Prototype-to-Product (1999-2000)

The first two cognitive maps address the earliest and latest snapshots in time and provide a strong contrast of issues associated with creating a company versus refocusing and reorganising a new but then established company.

The third cognitive map provides further insights into the actual development of both the Zaplet technology and applications using the Zaplet technology.

5.9.1 Cognitive Map 1 – Genesis of Zaplet (1998-99)

Development details for the first cognitive map include:

- Extract section, size and exhibit:
    - "History of the Company" Section, DeLacey & Leonard (2001, p.2); 1 page; Exhibit 5.7
- TACT word frequency lists and comments on key words:
    - First pass: 29 words with frequency > 2
    - Second & final pass: 14 words with frequency > 3
    - Final ranked word list: Table 5.1
    - Most of the top words were related to the founders Brian Axe and David Roberts plus development of the Zaplet application.
- NVivo ranked word, sentence and phrase node count = 53
- Association of concepts coding used: +, -. +/-, =

- NVivo PowerPoint cognitive map: Figure 5.7
- Reformatted PowerPoint cognitive map: Figure 5.8

The NVivo cognitive map captures much of the detail related to the development of both the Zaplet idea and the start-up company. The map highlights Brian Axe's logic in identifying a communication problem and articulating a need from it.

The map also emphasises David Robert's background in technology management and the positive combination of his management skills with Brian Axe's technical knowledge.

Roberts believed that with Axe's two prototypes of the Zaplet idea, and his business plan, they were ready to seek venture capital funding. However, seeking venture capital had some risk attached to it. Nevertheless, Roberts and Axe managed to gain an appointment with Kleiner, Perkins, Caulfield and Byers (KPCB) in July 1999.

Exhibit 5.7 Text Extract: Zaplet – Genesis of Zaplet (1999)

History of the Company

Zaplet, Inc. traces its roots to 1998 when Brian Axe and David Roberts were invited by mutual friends to go on a houseboat trip. This chance meeting precipitated a friendship that eventually led to collaboration. Axe graduated from UCLA in 1992 with an Engineering degree and obtained an MS in Engineering Management from Stanford in 1995. He worked for Hewlett-Packard and IBM before moving to GolfWeb in 1995, when dotcoms began to appear. In his work at GolfWeb, Axe noticed an interesting pattern. Even though engineers had access to a networked discussion board for building product specs, they rarely used it to interact or update information in real time. The same thing happened when he tried to get his friends to use club-type websites for scheduling and coordinating their group activities. "I found that we kept going back to e-mail," said Axe.

Axe identified the need to be met: "It dawned on me, if only we could create something that has the application functionality of the web and the communication simplicity of e-mail." In November 1998, Axe left GolfWeb to develop his ideas further at Reactivity – an incubator with the following mission: "Reactivity builds software products from concept to delivery. Reactivity combines the talents and skills of preeminent engineering and design teams to provide the best in New Venture Creation and Client Services." As Axe was developing prototypes of what would become Zaplet technology, he started to think about building a company around the Zaplet idea. He contacted David Roberts, a friend with shared values and key experience, to see if he would become a co-founder.

Exhibit 5.7 Text Extract: Zaplet – Genesis of Zaplet (1999) (Cont'd)

It was the beginning of 1999 when the concept of ZapletTm appmail took form as a dynamic, updateable, web-like message and application delivered through e-mail. The concept did not fundamentally change, although the venture it launched went through many permutations.

Roberts became increasingly interested and involved in planning how to carry this forward into a business.
In early 1999, he quit his job to work full time on a business plan. Roberts brought to the venture over 15 years of technology management experience, having led the development of some of the nation's most complex, state-of-the-art satellite systems. He had been special assistant to the director of the largest single program in the U.S. intelligence budget and served as an executive manager and decorated career officer in the Central Intelligence Agency and the U.S. Air Force. He graduated first in his training class at the CIA and persuaded the CIA to send him to Harvard Business School, where he received an MBA in 1992.
Early prototypes of ZapletTm appmail included an Event Planning application that allowed a group of friends to coordinate their social activities (such as ski trips) and a Group Purchase application to help friends and family purchase items at a volume discount. (See Exhibit 1 for a description of Zaplet technology.)

In July 1999, Axe and Roberts were ready to put their idea to a critical test – the scrutiny of venture capitalists. They selected four Venture Capital firms they wanted to meet. Through a friend at
Reactivity, they were able to arrange an appointment with their first choice, Kleiner, Perkins, Caufield & Byers. (See Exhibit 2 for excerpts from their presentation.)

An Incubator is defined to be "An organisation that helps start-ups develop in an accelerated fashion by providing them with a bundle of services, such as physical space, capital, coaching, common services, and networking connections." Morten T. Hansen, Nitin Nohria, and Jeffrey A. Berger, "The State of the Incubator Marketspace," Harvard Business School Publishing, June 2000.

Source: DeLacey & Leonard (2001, p. 2).

Table 5.1 Zaplet – Genesis of Zaplet (1999) – Ranked Word List

Anchor Concepts	Frequency
Axe	8
friend	7
Zaplet	6
Roberts	5
that	5
their	5
venture	5
with	5
application	4
develop	4
engineer	4
reactivity	4
they	4
when	4

Source: data analysis performed using TACT (1997)

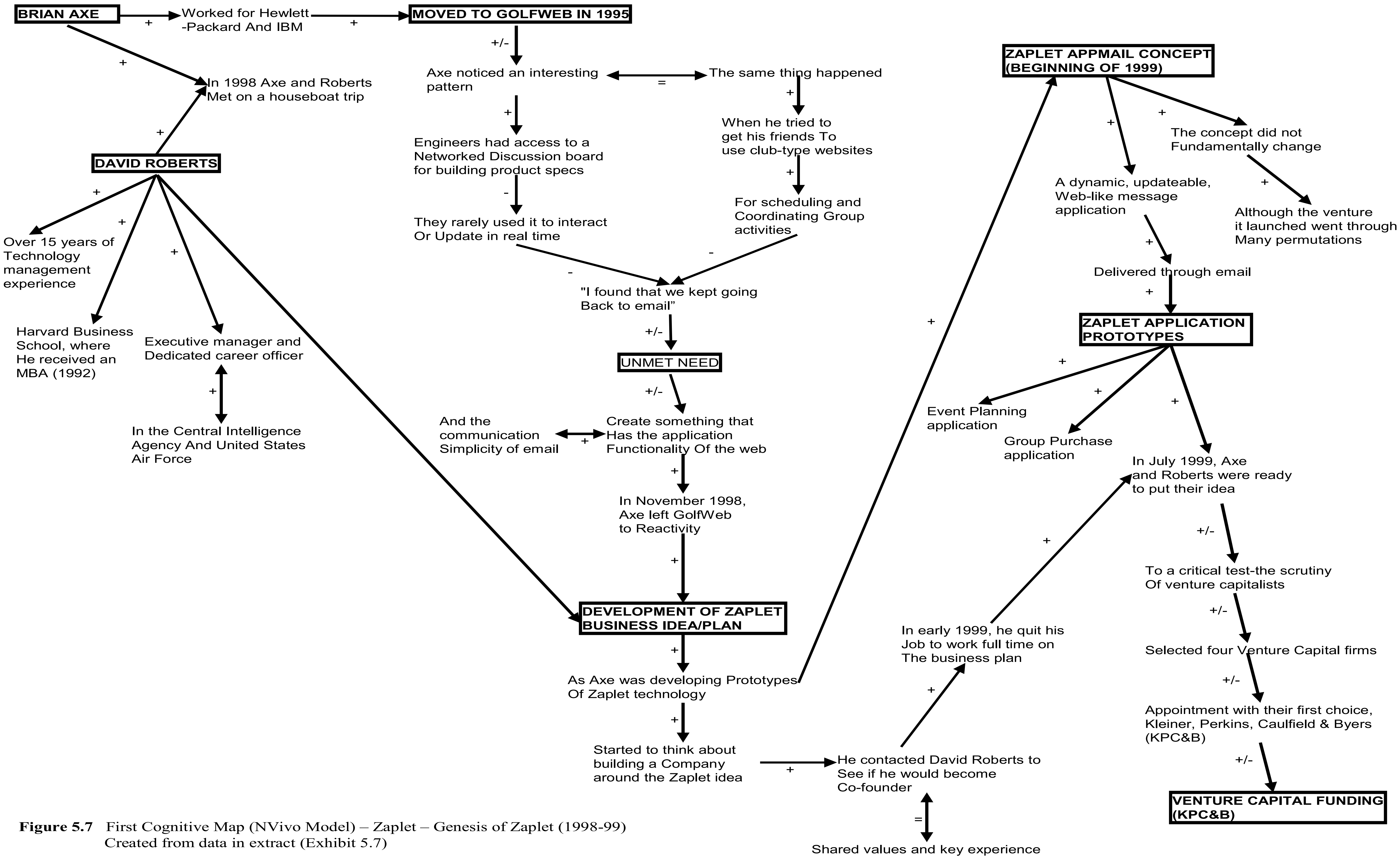


Figure 5.7 First Cognitive Map (NVivo Model) – Zaplet – Genesis of Zaplet (1998-99)
Created from data in extract (Exhibit 5.7)

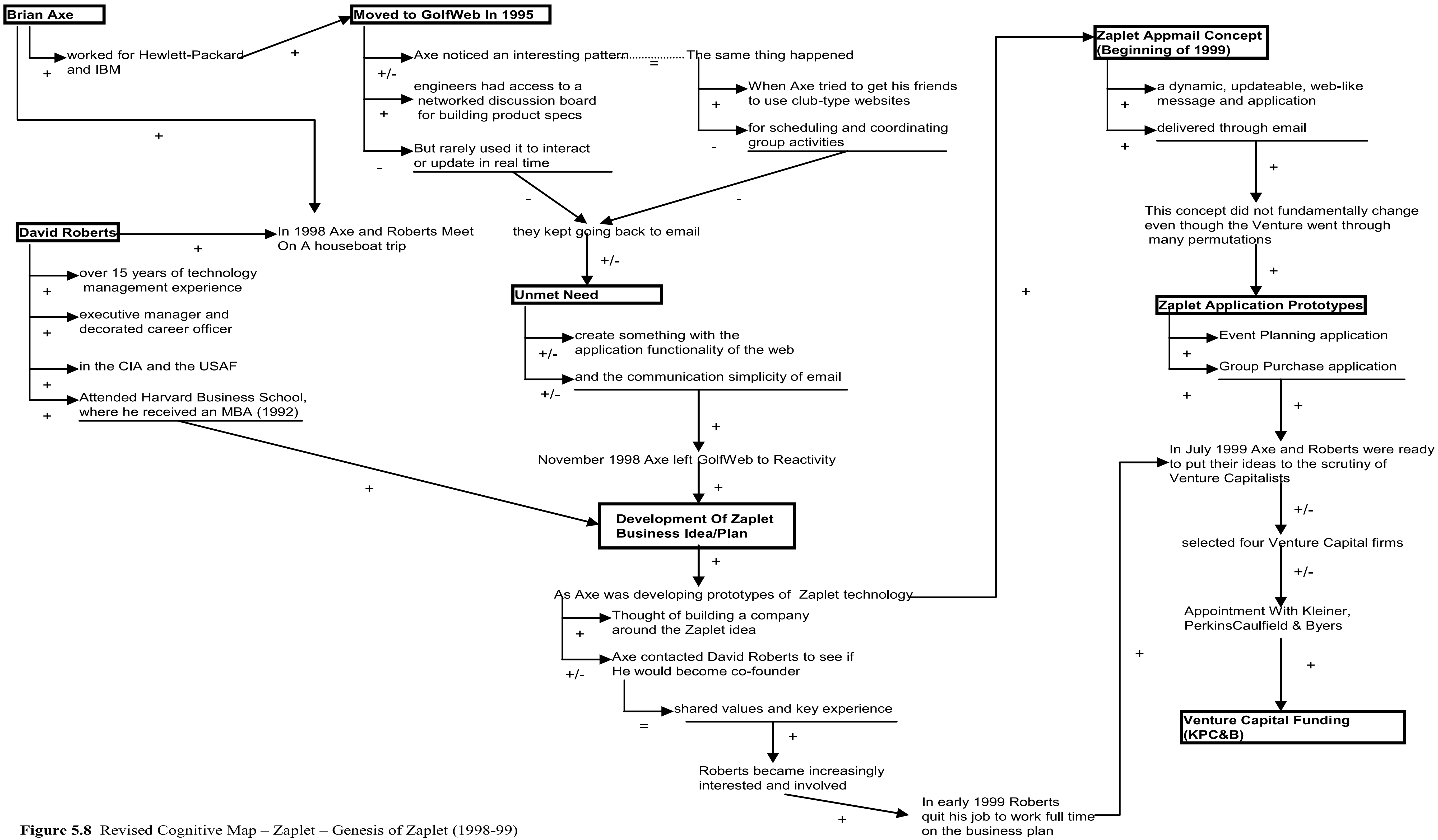


Figure 5.8 Revised Cognitive Map – Zaplet – Genesis of Zaplet (1998-99)
Translation of NVivo Map
Note. This revision was shown to participant in emic 2 stage for his (Robert's) interpretation.

5.9.1.1 Updated Cognitive Map 1 – Genesis of Zaplet (1998-99)

The cognitive map was revised after the interview stage (see Figure 5.9). Overall, there were only minor revisions required in terms of description, but there were significant formatting revisions.

The sections on Axe and Roberts backgrounds were unchanged (boxes 1, 2, 3 and 4). A new section (box 7) relating to Axe and Roberts meeting on a houseboat trip in 1998 was created. Additional commentary on the development of Zaplet technology at Reactivity in 1999 was incorporated into the map (box 5).

Java programmers assisted in developing the first mock-up of Java technology and the subsequent development of the early Zaplet application prototypes (boxes 5, 8 and 9).

A significant additional insight is that early application development undertaken using Java defined the architecture for the Zaplet technology, which was carried through all subsequent development of the technology and Zaplet applications (including an additional application for managing photos in box 9).

Selection of Java technology to build a new and exciting "killer application" for the Internet would have great appeal to Venture Capitalists, and leading-edge Java developers from Sun, where Java was conceived.

Roberts (2002) was keen to modify perceptions that either the Zaplet idea or company was born through discussion between him and Axe on the 1998 houseboat trip. In fact there was over a year of e-mail contact between Roberts and Axe before Roberts decided to focus full-time on developing a business plan for Zaplet idea (boxes 7 and 8).

The remaining sections on Venture Capital selection and funding were unchanged from the original cognitive map.

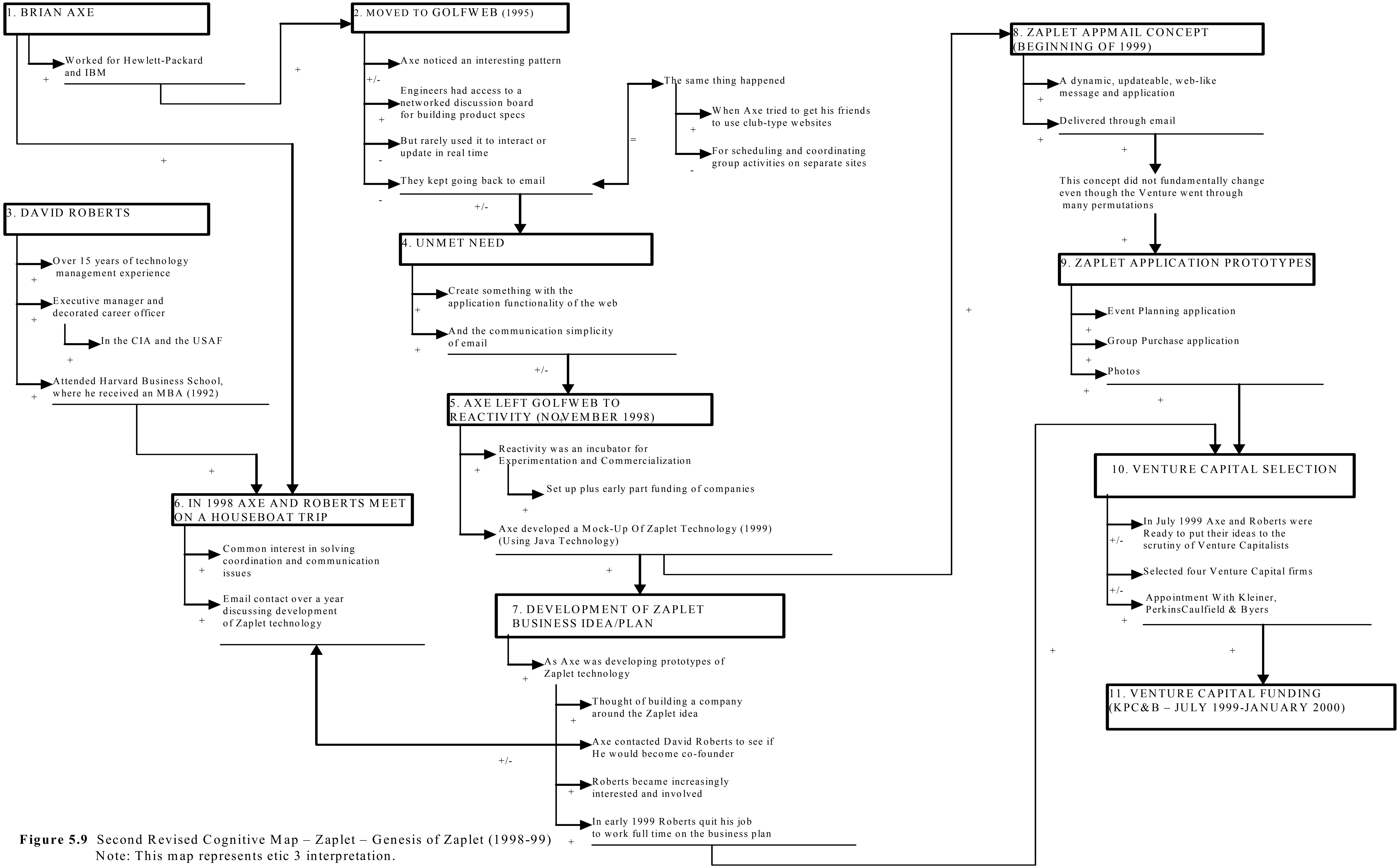


Figure 5.9 Second Revised Cognitive Map – Zaplet – Genesis of Zaplet (1998-99)
Note: This map represents etic 3 interpretation.

5.9.2 Cognitive Map 2 - Enterprise Analysis (June to September 2000)

Development details for the second cognitive map include:

- Extract section, size and exhibit:
    - "Enterprise" Section, DeLacey & Leonard (2001, pp. 13-15); 2½ pages; Exhibit 5.8
- TACT word frequency lists and comments on key words:
    - First pass: 56 words with frequency > 2
    - Second & final pass: 16 words with frequency > 5
    - Final ranked word list: Table 5.2
    - The words "market", "enterprise", "Zaplet" and "application" were prominent, which was not surprising as the extract was focused o applications for the enterprise market. Key marketing words or "jargon" such as "customer", "competition", "opportunity", and "differentiator" figured prominently in the word list.
- NVivo ranked word, sentence and phrase node count = 90
- Association of concepts coding used: +, -. +/-, =, "e"
- NVivo PowerPoint cognitive map: Figure 5.10
- Reformatted PowerPoint cognitive map: Figure 5.11

The final cognitive map presented in Figure 5.11 provides a clear sequential flow of the tasks undertaken in the enterprise market analysis project, with additional insights connected to tasks at the relevant points in the project.

Exhibit 5.8 Text Extract: Zaplet – Zaplet Enterprise Analysis (2000)

Enterprise

The company next turned its focus to the Enterprise area, which encompassed all software used internally by organisations seeking more effective and efficient value chain12 operations. The Market Segmentation Team's goal was to analyse the entire enterprise software market during the next two months and have a set of recommendations for the board of directors meeting mid-September 2000.

Vice President of Business Development Mitra – while gazing out the conference room window at the sidewalk and grass below – envisioned a way to present the enterprise market opportunity. He presented his new view of their enterprise market by drawing a picture on the white board (see Exhibit 6 for "eBusiness Market Definition and Context"). The key activities were labeled front-office (customer-facing); Back-Office (operations); Sell-Side (sales channel); Buy-Side (procurement and supply chain). At the center of all this enterprise activity was the corporate Intranet, focused on internal enterprise communications. This chart came to be known as the sidewalk chart because it looked exactly like the grass and sidewalk Mitra had been gazing at outside the Zaplet building. This served as the roadmap for the analytical work to follow.

First Phase Enterprise Analysis: Determining Revenue Opportunity

Data for the analysis came from different research firms including Forrester Research, Jupiter, Gartner, and IDC. An extra effort was made to normalize the data from these various sources. In addition, the staff was rich with past consultants who could leverage their knowledge and network of industry contacts. During this time, a team of summer interns (10 MBA students and 25 undergrads) working at Zaplet also provided tremendous energy and breadth of coverage in researching the market. They collaborated with friends, vendors, and a large cross-section of enterprise customers for teleconferences and face-to-face meetings.

Still, it was challenging to do a revenue opportunity analysis without having specific product ideas in mind. Half the team wanted to define Zaplet applications prior to estimating the addressable market opportunity. The other half believed describing specific applications was myopic and might miss a whole segment of opportunity – they proposed measuring the addressable market (that is, the fraction of the total market for Enterprise Software where Zaplet technology could be applied).

In the end, the collective team decided to press on without defining specific applications. Instead, they looked at overall market size and made estimates of revenue opportunity based on that.

Teams developed lists of Zaplet differentiators and how these would apply for various target market segments. Tancer led the way in preparing a rich variety of graphical charts to gather and communicate team findings. (See one example of this in Exhibit 7 on "Segmentation Fit.") The Executive Team reviewed these as part of an all-day Executive Team Offsite meeting and subsequently decided to focus on the following key technology differentiators of Zaplets:

"The value chain disaggregates a firm into its strategically relevant activities in order to understand the behaviour of costs and the existing and potential sources of differentiation," Michael E. Porter, Competitive Advantage: Creating and Sustaining Superior Performance (New York: The Free Press, 1985), p. 33.

Exhibit 5.8 Text Extract: Zaplet – Zaplet Enterprise Analysis (2000)(Cont'd)

* Rapid Process Automation by Anyone – allowing enterprises to quickly snap together a "corporate process" solution – something any user could snap together without technical assistance.
* Contextual Information Gathering and Reuse – for example, budget information gathered from people on a real-time, continuously updated basis.
* Integrated Secure Applications Delivery – enabling more efficient collaboration Efficient Collaboration and Decision-Making
* Interaction Without Boundaries – expanding the communication reach companies have, beyond their internal intranet to customers and partners over the Internet

Each segment was then ranked according to:
* market attractiveness, Size of market, five-year projected growth, and profitability.
* specific fit and sustainable value, What is Zaplet's "value proposition fit" for a particular market? How well does it leverage Zaplet appmail technology differentiators?
* Barrier to entry analysis, How difficult would it be to enter this market?

Product Managers reported results to the Executive Team. All of the results were normalized across all the analytical teams regarding the Market Attractiveness, Qualitative Assessments of Fit, and Barriers to Entry.

Second Phase Enterprise Analysis: Defining Killer Applications
The company then analysed various sub-segments of the enterprise market. According to Tancer,
We took all the Product Managers from the first pass and re-divided them into five teams for the second phase. The mission for each of these teams was to come back with the killer applications in: Finance, Customer Service & Order Management, Sales Force Automation & Partner Relationship Management, Intranet & Teamware and Human Resources (HR).

Through this process, team members began to believe that the true value in Zaplet technology was in tying all of these systems together, into a dynamic environment that could always reflect current data in a system. This became a key piece of the overall strategy. For instance, Zaplets could glue together Order Management, Escalation Management, and Interaction Management. (Exhibit 8, the "Customer Support: Drill-down" is typical of the analysis that was presented.) The teams also identified vendors competing in these markets. The team developed their own competitive analysis. According to Tancer:
In defining what companies were competitive with Zaplet, we faced the difficult task of plotting how competitors fit against each of our five differentiators. I decided to use spider charts, similar to a radar chart with five arms radiating from a central point with each arm representing a differentiator. A competitor's strength was plotted on each of the five arms, resulting in a map of a competitor's position relative to Zaplet.

The popular trade press covered the competitive environment as well:
Creating a Zaplet is about as easy as ordering a book from Amazon.com. ... There is competition, sort of. Gizmoz, focusing on sports and pop culture, distributes links to basketball and soccer cards and horoscopes via e-mail but runs only on the Web, popping up within your browser. You can't work on a business plan with an employee across the country. Business-focused 2Way offers systems, priced $2,000 a month to $80,000 a pop, that let you communicate with selected partners and customers.

Exhibit 5.8 Text Extract: Zaplet – Zaplet Enterprise Analysis (2000)(Cont'd)

Bigger stirrings abound in the area of real-time collaboration. Lotus Development's Sametime software hit the market last year, and Microsoft's Exchange 2000 will roll out later this year. But both programs require special software to communicate with other parties.

As one person described the final stages of the Market Segmentation Analysis:
The whole company was starting to come together. It was really fascinating to see this evolve. We started by asking "What do we do?," subsequently we decided to "Concentrate on the enterprise," and then we turned to "Concentrate on specific segments." Now we are moving towards specifying attractive applications within these sub-segments. We're getting more and more focused on what we want to do.
Meanwhile, engineers were building the basic platform (upon which specific features and applications, when defined, would be built.) Key capabilities such as security, connectors to thirdparty software applications like database and customer relationship management systems were defined generally and independent of specific applications. Engineering started to include and prioritise some of these capabilities into their development plans.

Source DeLacey & Leonard (2001, pp. 13-15)

Table 5.2 Zaplet – Enterprise Analysis (2000) - Ranked Word List

Anchor Concepts	**Frequency**
market	20
team	16
enterprise	13
Zaplet	13
applications	10
analysis	8
customer	7
define	7
manage	7
all	6
competition	6
develop	6
differentiator	6
opportunity	6
software	6
specific	6

Source: data analysis performed using TACT (1997)

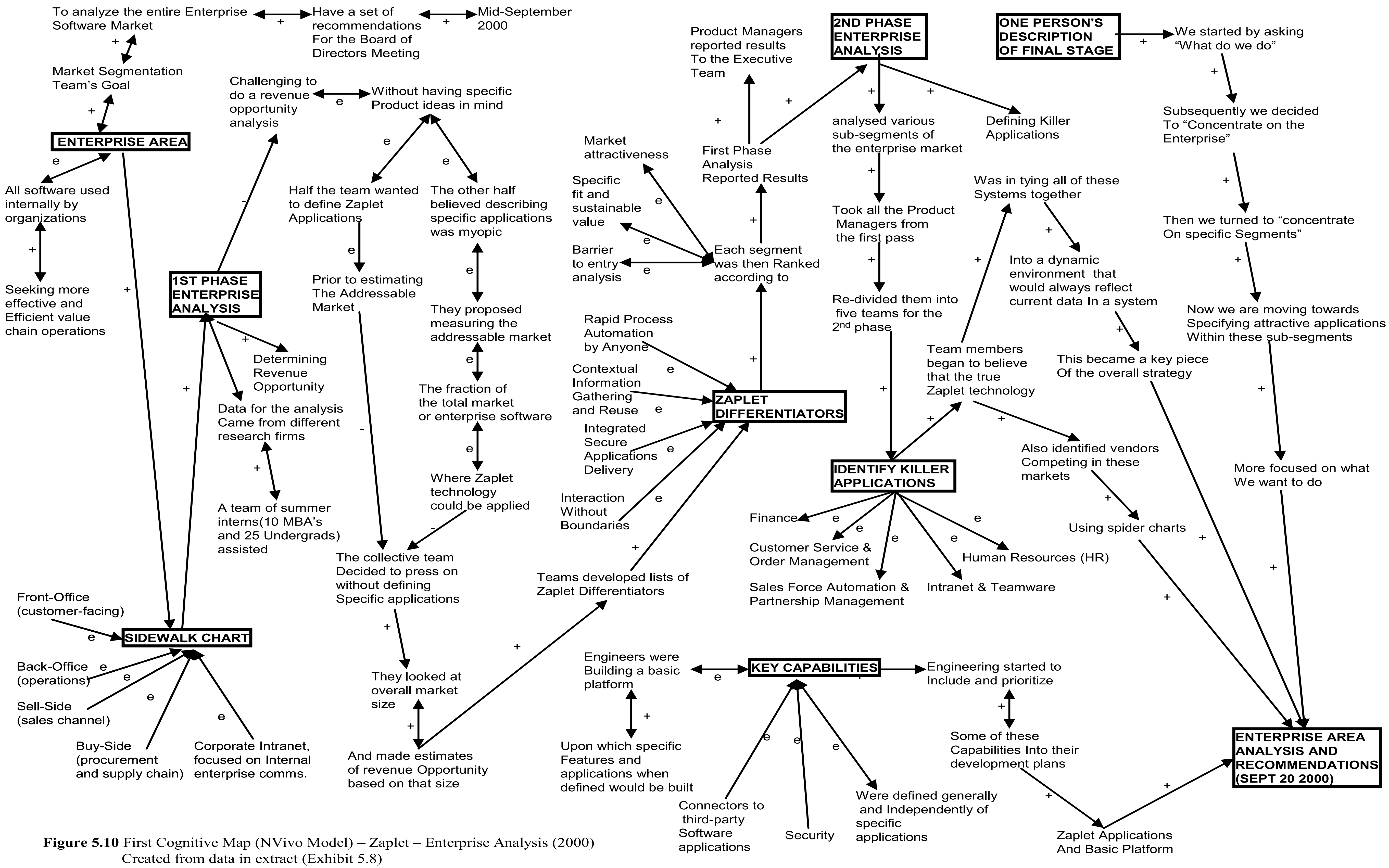


Figure 5.10 First Cognitive Map (NVivo Model) – Zaplet – Enterprise Analysis (2000)
Created from data in extract (Exhibit 5.8)

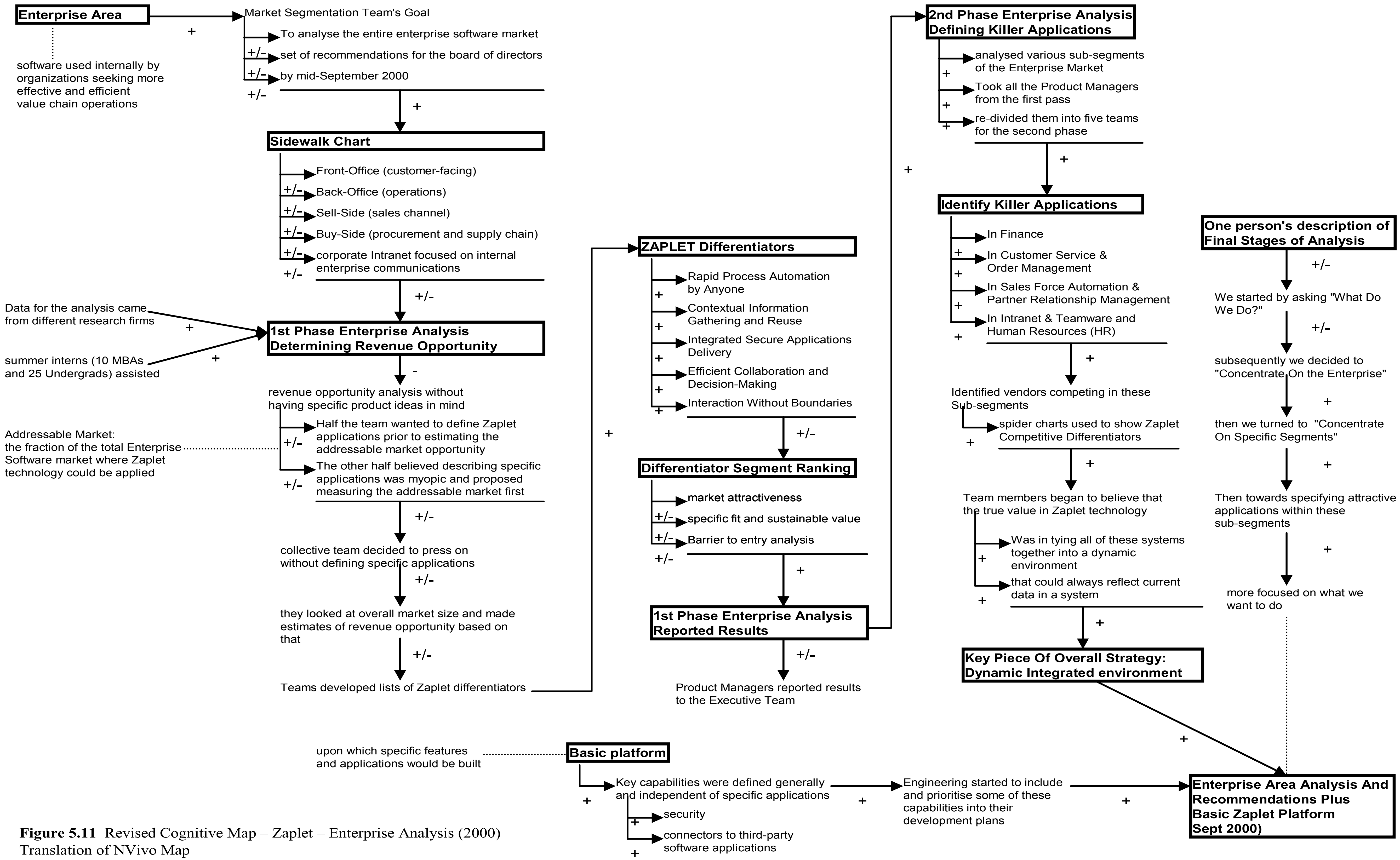


Figure 5.11 Revised Cognitive Map – Zaplet – Enterprise Analysis (2000)
Translation of NVivo Map
Note. This revision was shown to participant in emic 2 stage for his (Robert's) interpretation.

The actual enterprise market was the last market segment to be formally analysed after the other three market segments targeted by Zaplet, that is, consumer, commerce, and content.

The enterprise segment was subjected to the most extensive and formally rigorous analysis out of the four segments. The enterprise market team's goal was to complete an entire segment analysis including a set of recommendations for Zaplet's board of directors by mid-September 2000.

At one of the early team meetings, Samir Mitra developed a chart with five key enterprise activities that is, front-office, back-office, sell-side, buy-side, and corporate intranet.

This chart became known as the "Sidewalk Chart" and acted as a roadmap for further analysis. At that stage there was no assessment of Zaplet's capabilities or competitive position in the enterprise market as expressed by these key enterprise activities.

The first-phase of enterprise market analysis focused on determining revenue opportunity. Data was obtained from several different research firms. MBA and undergraduate students serving summer internships assisted with collecting and collating market research data.

There was an immediate dilemma in determining revenue opportunities, as the team was split between defining Zaplet applications before estimating market opportunities, and starting with defining market sizes before defining Zaplet applications.

The former position was a logical progression of the way Zaplet had evolved as a start-up company through 1999-2000, brainstorming new application ideas and then attempting to develop prototypes of them.

The latter position represented a significant departure in the philosophy of Zaplet toward understanding market potential for its technology.

The team chose to estimate an overall enterprise market revenue figure. These estimates may well been "best-guess" team estimates, as it was unclear how a figure could be developed without some speculation regarding potential Zaplet applications.

There seems to have been a quick switch in direction for the team away from focusing on the enterprise back to identifying Zaplet technology differentiators that could be applied to all of the researched market segments. One reason for such a switch could have been due to ongoing problems over defining the market versus defining applications.

Five differentiators were identified which were regarded as positive for Zaplet. These differentiators were then applied to the different market segments (consumer, content, commerce, and enterprise) and then ranked according to market attractiveness, specific fit/sustainable value, and barrier to entry analysis.

The overall results from the first-phase enterprise analysis confirmed the enterprise market as the most attractive market segment for Zaplet in terms of both potential revenue opportunity and Zaplet's differentiators.

The second-phase enterprise analysis focused on identifying "Killer Applications" in the enterprise market segment. Product Managers used in the first-phase analysis were redivided into five teams to define these applications.

Four "Killer Application Areas" were identified, that is, finance, customer service & order management, sales force automation & partner relationship management, and intranet & teamware and human resources (HR).

Each of these "Killer Application Areas" (sub-segments) was further analysed, identifying essential competitive vendors and to assess Zaplet's competitive position against them. Spider-charts using Zaplet's differentiators as axes were created as part of this analysis.

Team members realised that tying all the killer applications sub-segments together prompted them toward conceptualising the collection as a complete system. This insight became a key piece of Zaplet's overall strategy. However, this insight would also put Zaplet directly in the competitive path of the major Enterprise Resource Planning (ERP) vendors.

Although Zaplet could not hope to directly compete against ERP vendors through offering a standard integrated Enterprise application suite, it believed that there was a small window of opportunity to develop a set of Internet-based applications (an e-commerce enterprise suite) as none of the major ERP vendors had converted their applications for the Internet by 2000.

The main alternative to integration was to develop killer applications in niche areas that were "must-haves" for enterprises. The theme of balancing between a comprehensive enterprise-wide e-commerce application suite, and focusing on specific applications, is important for not only the Zaplet case study but also the Kana and Trilogy cases in this research project.

The results of the full enterprise analysis were presented to Zaplet's Board of Directors in September 2000.

Further insights were drawn from the second-phase of enterprise analysis. One person's description of the final stages of the analysis indicated a staged focusing down to what Zaplet needed to do in the enterprise market.

The Engineering Group also started to add capabilities to the Zaplet technology platform that could support the development of Zaplet applications that would come out of the enterprise analysis.

The enterprise market segment market analysis findings and recommendations were key inputs for Baratz as the newly appointed CEO of Zaplet to change the direction and focus of the company in late 2000. Organisational restructuring and application development in Zaplet for 2001-02 can be traced directly back to these findings and recommendations.

5.9.2.1 Updated Cognitive Map 2 – Enterprise Analysis (June to September 2000)

An updated cognitive map for enterprise analysis was developed after the interview stage and is presented in Figure 5.12. Most sections were validated as unchanged, although some significant additional insights and comments were incorporated into the map.

The enterprise area (box 1) was renamed the enterprise market, plus it was noted that Baratz initiated the enterprise analysis, probably at about the same time he was appointed CEO.

The "Sidewalk Chart" and sections covering the first-phase of analysis (boxes 2, 3, 4, 5, and 6) were unchanged apart from some sectional formatting.

There was confirmation of some confusion as the team worked through defining the enterprise market revenue opportunity versus defining new Zaplet applications.

There was no confirmation of any potential revenue estimates and the actual reduction of the total revenue opportunity to an addressable market.

Defining Zaplet Differentiators and then drawing back to a cross-segment analysis seemed to reinforce some of the tension and confusion between a technology and a market focus with the team.

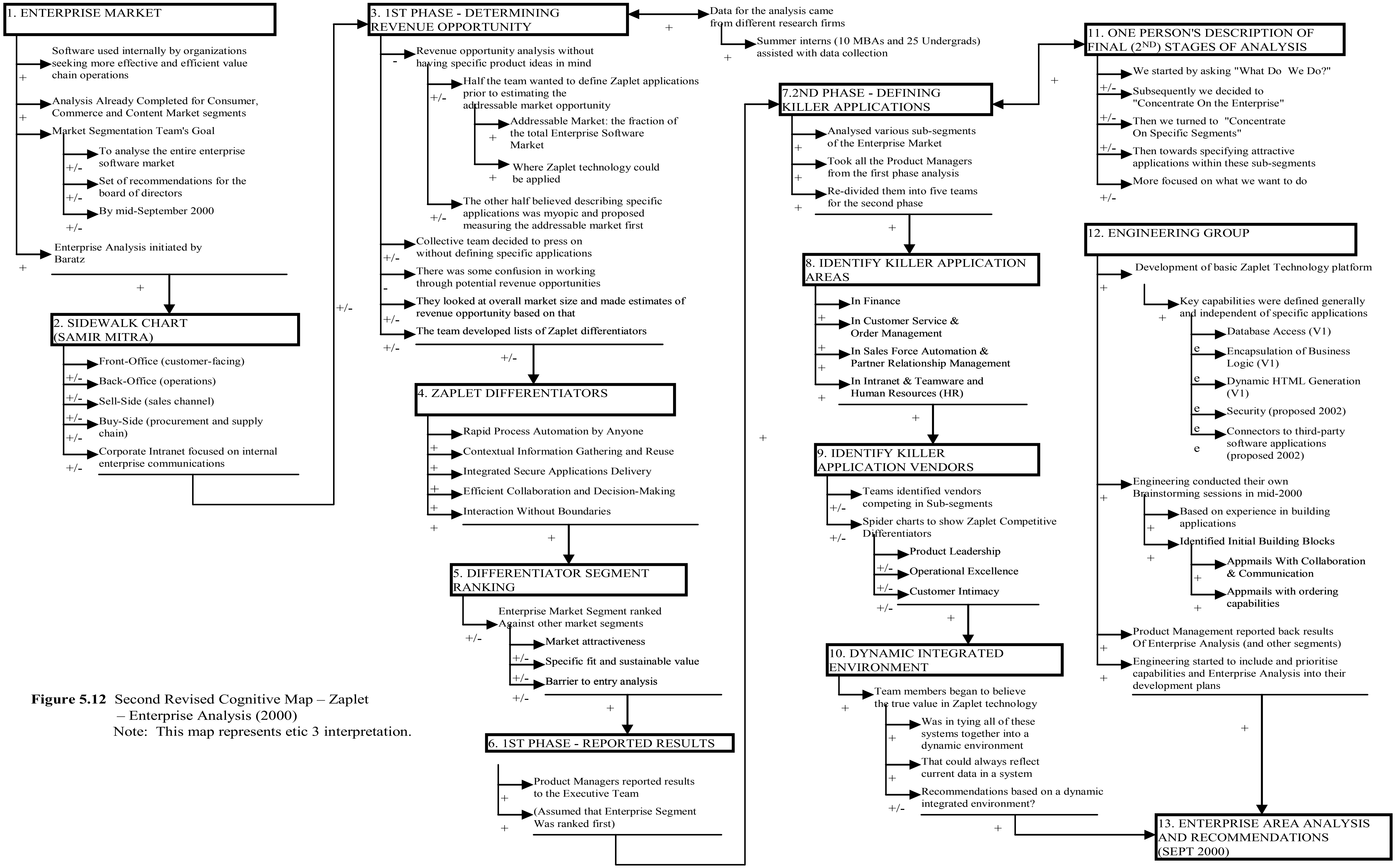


Figure 5.12 Second Revised Cognitive Map – Zaplet – Enterprise Analysis (2000)
Note: This map represents etic 3 interpretation.

There was no detail of the reported results to the executive team but it is a reasonable assumption is that the results would have favoured further analysis of the enterprise market segment ahead of the other market segments.

The second-phase enterprise analysis (boxes 7, 8, 9, 10) was reformatted to highlight distinct sections for Identification of Killer Application Vendors (box 9) and a “Dynamic Integrated Environment” (box 10).

Although the spider charts on competition may have addressed the Zaplet differentiators, a more standard spider chart approach was used in marketing strategy using dimensions of “Product Leadership”, “Operational Excellence”, and “Customer Intimacy” was applied in this analysis (Treacey & Wiersma 1996).

Roberts (2002) and Chandra (2001) confirmed that the team realised that viewing killer applications as a total dynamic system could be a key Zaplet differentiator (box 10). However, this differentiator had not been actually applied by Zaplet to its application development in 2001-02 – at least not in Zaplet V2.0 and V2.5.

The perspectives presented on the final stages of analysis (box 11) were attributed to a known person, but it was agreed that the identity remain confidential.

There was significant additional commentary on Engineering Group activities (box 12). Key capabilities to be incorporated into the Zaplet technology platform were expanded and defined by Zaplet Versions.

Product Managers reported back to the Engineering Group on findings from the enterprise analysis and these were incorporated in Engineering’s future development plans. The Engineering Group also conducted brainstorming sessions on developing new Zaplet Building Blocks in mid-2000.

A stronger alignment between the market analysis and Zaplet technology development was addressed through the re-organisation of resources around one enterprise business unit through late 2000 and into 2001. However, this re-organisation was outside the scope of the cognitive maps created for the case study

The updated cognitive map highlights decision-making associated with running a multi-stage market segmentation analysis for an application technology that could be used in a very wide range of potential applications. There was substantial tension related to bounding potential application uses down to distinct market segments, which may have contributed to a horizontal rather vertical approach to market segmentation.

The shift toward an enterprise focus was reflected in strategic, organisational and application development decisions and outcomes through 2001-02.

5.9.3 Cognitive Map 3 – Prototype-to-Product (1999-2000)

Development details for the third cognitive map include:

- Extract section, size and exhibit:
    - "From Prototype-To-Product" Section, DeLacey & Leonard (2001. pp. 6-7); 1½ pages; Exhibit 5.9
- TACT word frequency lists and comments on key words:
    - First pass: 56 words with frequency > 2
    - Second & final pass: 22 words with frequency > 5
    - Final ranked word list: Table 5.2
    - Apart from the words "Zaplet", "business", and "company", Vinod Khosla's role in developing ideas for application prototypes was highlighted through his name being prominent on the word list. Several words scoring frequencies of 5 and 4 were related to building and developing applications and prototypes.
- NVivo ranked word, sentence and phrase node count = 62
- Association of concepts coding used: +, -. +/-, =, "e" ("e" dropped in translated map)
- NVivo PowerPoint cognitive map: Figure 5.13
- Reformatted PowerPoint cognitive map: Figure 5.14

The map highlights a flow of idea creation and then managing those ideas into a set of achievable applications by the Engineering Group. There is a parallel flow describing a re-organisation of Zaplet into Business Units up to the public launch of Zaplet as a company.

As highlighted in the events chronology map and the first cognitive map, Axe and Roberts had developed early prototype applications in Event Planning and Group Purchasing, plus a photo application.

Nevertheless, in during their meeting with Vinod Khosla at KPC&B in July 1999, they realised that the current prototypes would have to be thrown away, and the new technology redirected toward the development of new applications.

Exhibit 5.9 Text Extract: Zaplet – Prototype-To-Product (1999-2000)

(See Exhibit 5.6 for a description of the building blocks.)

From Prototype to Product
The distance between Axe and Roberts' prototype and a final product had been clear to Khosla from the start. He knew the technology they had developed had to be thrown away, "It was very clear to me from day one there was nothing there that was usable. So most of what I did through December was ask questions about recruiting. I didn't care about anything else."

One of the great challenges with this novel Zaplet technology was how to imagine everything they could do with it. Khosla believed because good entrepreneurs are focused on execution "it's really hard for them to be truly dreamers." Khosla said, "Too much dreaming is not good and too much heads-down is not good. The right approach is a little bit of both. I try to emotionally bounce entrepreneurs between these two walls of practicality and dreaming. Sometimes they think I'm jerking them around, but I really do want them to take both views."

Moreover, based on long experience, Khosla believed there were two alternative models for managing in companies, the sergeant and the shepherd.
In Khosla's view, the more authoritarian sergeant model was appropriate when everything is defined:
Everybody marches in lock step towards a defined goal. "We're getting to X. OK? That's where we go." And everybody's taking steps and coordinating, just like in a marching army. The shepherd is much more appropriate for start-ups, where it is a process of discovery. A start-up is like herding sheep, where there are plenty of sheep in a herd that are going the wrong way at any given time, but the herd as a whole is moving in a general direction that is not rigidly specified. The shepherd allows for enough experimentation.

In September 1999, Axe and Roberts held the title of co-president, dividing projects between themselves, sharing responsibility for decisions and for acting on behalf of the company. Sharing the top role proved problematic. When Khosla sent them both e-mail, it was not clear who should act; on occasion each assumed the other would. After a few tasks dropped through the cracks, they decided to split the responsibilities of CEO and CTO, although they did not assume those titles officially. Both had strong technology backgrounds. As Axe was working on product-related activities at the time, he took on the role of CTO. Roberts assumed the role of CEO, since his background and activities were management-related.
By October 1999, a growing engineering team was working under the leadership of Chandra.

The company moved into new offices with about 25,000 square feet of office space. They had enough space for about 100 people, with only 36 people on staff at the time. The empty space left plenty of room for Nerf gun wars, Frisbee, a miniature golf course, and other stress-relieving activities. It also served as a constant reminder of the explosive growth being planned.

The pace was frenetic. Lunches were provided to employees on a daily basis, both as a perk and to increase productivity. The kitchen was generously stocked with plenty of drinks and snacks.

Exhibit 5.9 Text Extract: Zaplet – Prototype-To-Product (1999-2000) (Cont'd)

Four MBA students from Stanford Business School were brought in under the direction of Axe to help establish intellectual property for the company. The students learned about the technology and spent time developing ideas for Zaplet applications. Their objective was to create as many different examples of Zaplet applications as they could imagine. This effort contributed directly to the process of filing for patent protection.

The Engineering team derived another benefit from this activity: Chandra's team spent a month implementing prototypes based on some of these Zaplet ideas. According to Chandra:

We looked for patterns in the prototypes. Lo and behold two distinct patterns emerged. First, there were a number of services. Second, the engineering team identified a group of building blocks of functionality for building Zaplets in conjunction with the services. With these, you could create a lot of different Zaplet applications by simply combining these building blocks and turning on whatever services and features you wanted. That was a fabulous discovery because we found that with about 30 building blocks we could accommodate close to 200 of these Zaplet ideas that the B-school students had come to us with.

In December 1999, Roberts and Axe began inviting more of their staff to weekly meetings with Khosla, eventually including all the top managers. In order to learn as much as possible and explore all the different application areas identified, Khosla suggested they set up business units (also referred to as Areas) around different customer opportunities. This structure, with area managers reporting directly to the CEO, enabled the fledgling company to attract top tier talent and was an important recruiting advantage. An area manager led the business unit and was responsible for creation of the business plan. Each business unit had its own dedicated product managers and business development people as well as leadership. When Alfonsi joined he had never seen anything like it, a small company with many individual business units. Still, Alfonsi was convinced the experimentation would pay off:

We thought there was enough opportunity for every one of these [areas] so that each could be a viable business. I was heading up Content. We developed the first two customers Zaplet ever had-ZDNet and USA Today. The Republican National Committee was another. It was exciting to ship product and get it up and running on the customer's website.

On March 13, 2000, Zaplet emerged from stealth mode with a public launch of their website at Zaplet.com and product plans.

Extracted from DeLacey & Leonard (2001, pp. 6-7).

Table 5.3 Zaplet – Prototype-to-Product (1999-2000) – Ranked Word List

Anchor Concepts	**Frequency**
Zaplet	10
business	8
Khosla	7
company	6
area	5
axe	5
both	5
building	5
manage	5
product	5
activity	4
application	4
blocks	4
develop	4
different	4
prototype	4
Roberts	4
team	4
technology	4
time	4
two	4
unit	4

Source: data analysis performed using TACT (1997)

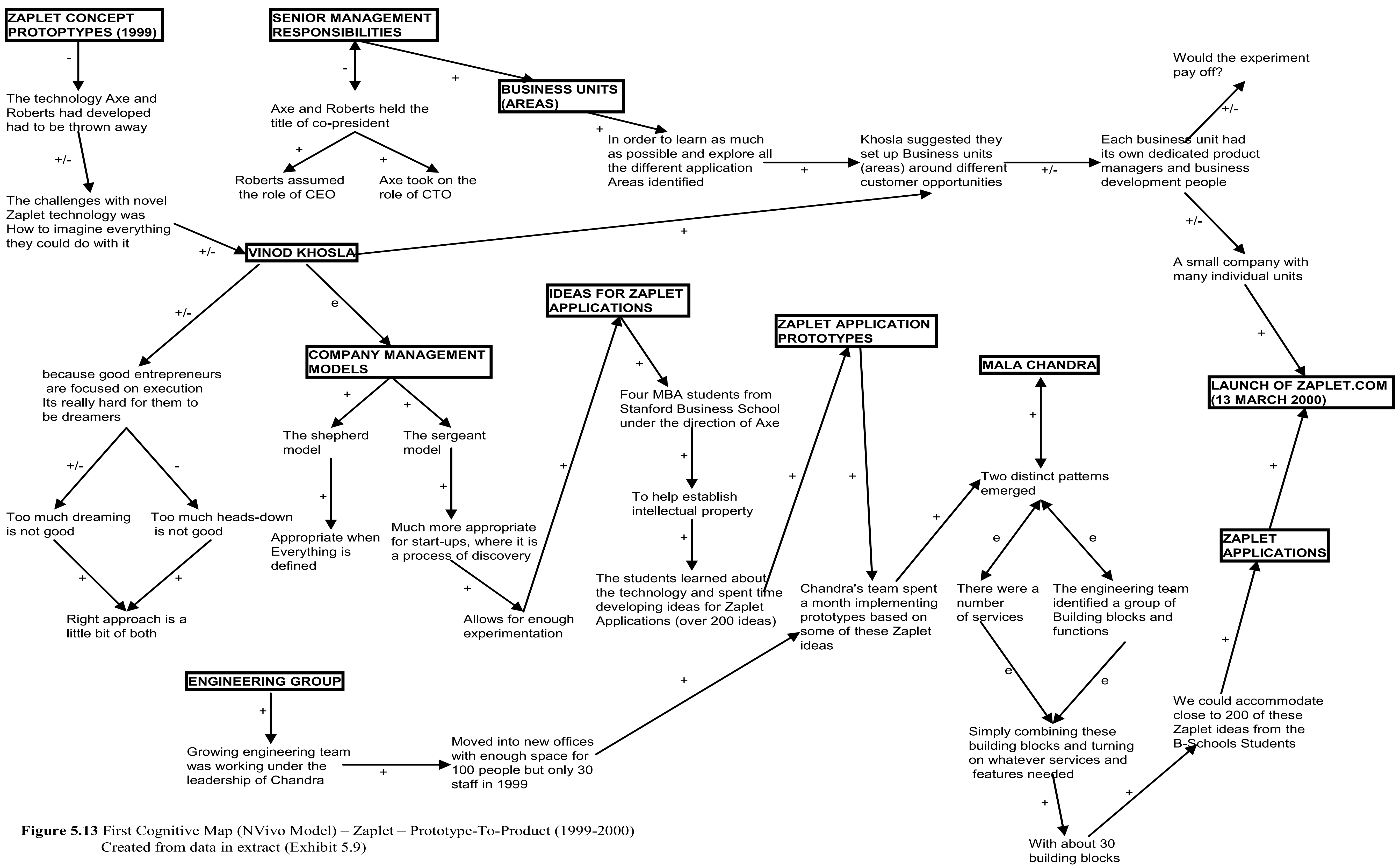


Figure 5.13 First Cognitive Map (NVivo Model) – Zaplet – Prototype-To-Product (1999-2000)
Created from data in extract (Exhibit 5.9)

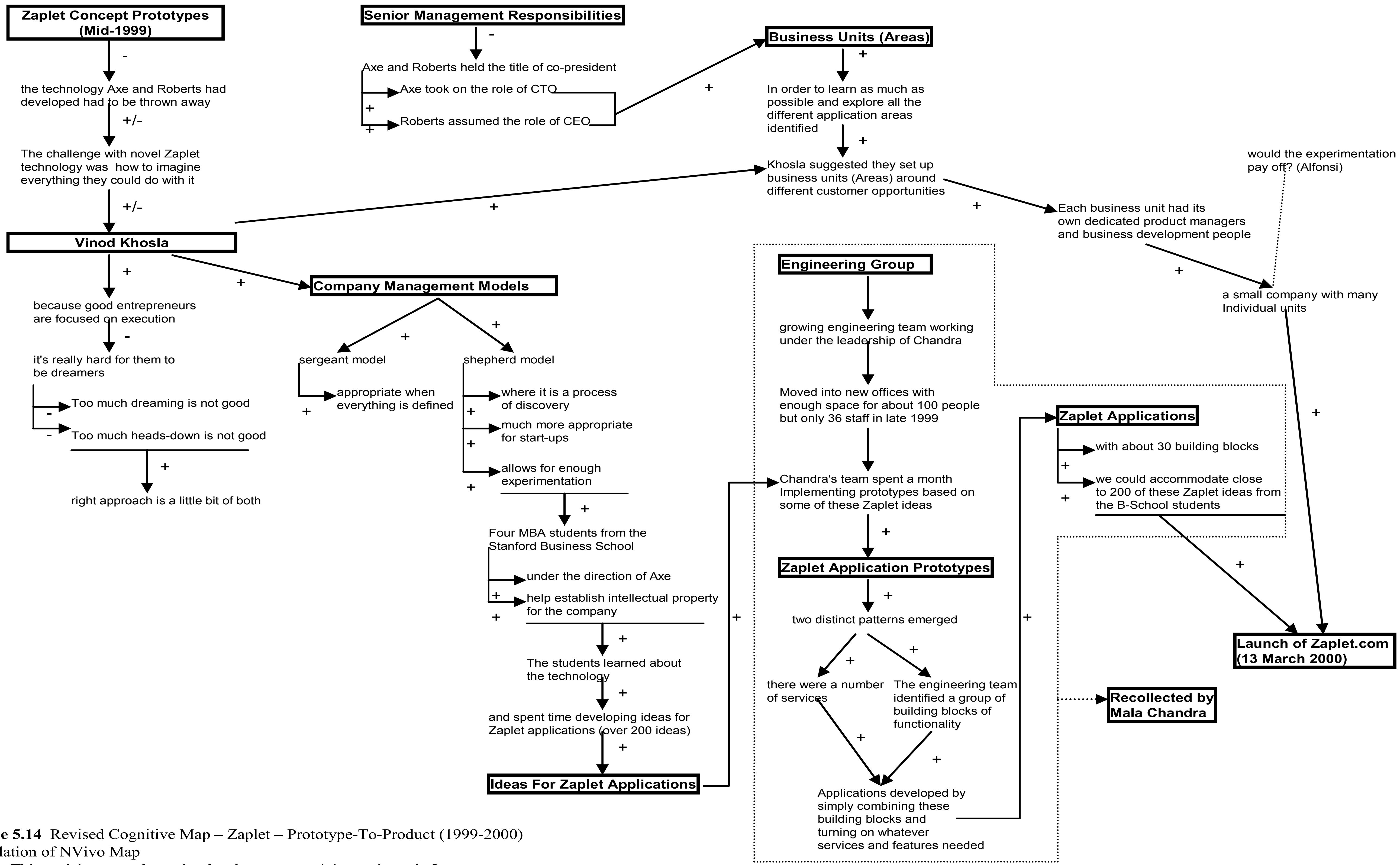


Figure 5.14 Revised Cognitive Map – Zaplet – Prototype-To-Product (1999-2000)
Translation of NVivo Map
Note: This revision was planned to be shown to participants in emic 2 stage

Vinod Khosla's insights into balancing the development of new ideas and actually developing applications (or products or technologies) assist in understanding development of new Internet-based applications in a fast – moving high-technology business environment.

Khosla noted that entrepreneurs are strongly focused on execution but need to dream or imagine new uses for their technologies. Khosla extended his view to defining two company management models for development of new technologies – the sergeant and shepherd models.

The sergeant model is appropriate when a technology and its potential applications are well defined. The shepherd model is more appropriate where the technology is new or novel and where there is a potentially large range of uses or applications. Khosla viewed the shepherd model as appropriate for Zaplet.

These models could be seen in themselves as "decision-making" models. They set a decision-making infrastructure or platform within a start-up company, to work with the underlying decision-making processes of the founders of the company. An interesting issue is the point or period at which a company transitions from a shepherd to a sergeant model.

Khosla encouraged Zaplet to imagine possible applications for the Zaplet technology. Actual execution of this exercise was undertaken by a group that was not actually a permanent part of the organisation, but had a fresh and independent view of Zaplet's products and services.

Four Stanford MBA students were working with Zaplet to learn about the Zaplet technology and to establish intellectual property for the company, and Brian Axe decided to redirect their work efforts toward developing ideas for Zaplet applications. The students developed a list of over 200 ideas for applications.

At about the same time, Axe and Roberts were co-presidents but decided to change organisational roles. Axe moved to the role of Chief Technology Officer (CTO) and Axe Chief Executive Officer (CEO).

Zaplet restructured below the Executive Team, following a suggestion from Khosla to set up business units (or areas) around different customer opportunities. Each business unit had its own product and business development managers.

Richard Alfonsi, who was appointed the General Manager for the content business area, was not convinced that a small company with many business units would pay off.

Nevertheless, the business units were in place by the time Zaplet was publicly launched on 13 March 2000.

At the same time as Zaplet's restructuring activities, the Engineering Group's software development team recruited large numbers of software engineers. In the context of the Zaplet case, the term "software developer" is interchangeable with "software engineer".

Most of the funding gained by Zaplet through working with KPC&B was ploughed into recruiting a strong software development team. Mala Chandra came across to Zaplet from Sun Microsystems with a very strong reputation in the management of software development.

By late 1999, Zaplet employed 36 people, mainly in Engineering, and moved to premises at Redwood Shores which had space for up to about 100 employees, mainly software developers.

The ideas list generated by the Stanford students was passed on to the Engineering Group of which spent a month creating prototypes based on some of them. According to Chandra, the Engineering Group made a major breakthrough when it realised that application development could broken down to two distinct patterns.

Subsequent to identification of services (or potential application opportunities), the Engineering Group could either build or customise a set of basic applications (or "Building Blocks") for those services.

Chandra believed that around 30 Zaplet Building Blocks could be developed, and then customised to address most of the 200 ideas in the students' list. This insight seemed to have potential to greatly improve the Engineering Group's efficiency in developing several new Zaplet applications. Zaplet's strong development capability would be a core competency for the company strength at its launch in March 2000.

Overall, the cognitive map describes a small start-up thinking big about the potential of its application technology – "big" in terms of potential application opportunities, software development capability, and organisational structure. This approach is not surprising, given that the company was at an embryonic stage with its venture capital company judging its technology capable producing "killer applications".

Assembly of a very strong application development reinforced perceptions that "killer applications" could be rapidly developed and delivered to market first.

Alfonsi's perceptions of a small company with many business units were a good counter-balance to the prevailing approach of aggressive expansion of as many ideas as practically possible in late 1999.

5.9.3.1 Updated Cognitive Map 3 – Prototype-to-Product (1999-2000)

Following the interview stage, an updated cognitive map was created incorporating significant changes in formatting, new sections, new links, and new insights (see Figure 5.15).

The "Zaplet Concept Prototypes" of mid-1999 (box 1) were at a less developed stage than suggested or implied in the DeLacey & Leonard (2001) case study.

The "prototypes" were some software programs that could demonstrate the possibility of Event Planning or Group Purchasing or managing photos – but they were not actual software applications.

Vinod Khosla of KPC&B was convinced that the basic Zaplet technology could be applied to many opportunities. However, a lot of investment into setting up an effective application development team was required before any real applications could be developed.

Khosla's views on entrepreneurship and company models (boxes 2 and 3) were unchanged. Development of ideas for Zaplet Applications (box 6) was also largely unchanged.

However, Roberts (2002) noted that most of the Zaplet application ideas developed by the students were consumer-oriented, whereas twelve months later, most of the actual applications relevant to Zaplet's future were actually enterprise-based.

A significant set of decisions relating to application development during early 2000, but not addressed in DeLacey & Leonard's (2001) case, was the selection of Operating System platforms for application development (box 4).

Although strictly speaking, Java is supposed to be operating-system-independent, early Zaplet development was undertaken on Sun Microsystems servers and workstations using Solaris (Sun's version of UNIX). The rapidly expanding Engineering Group was mainly experienced in programming Java on Solaris.

Linux and Microsoft's Windows-NT and were rapidly gaining market share as server operating systems during 2000. Zaplet technology was ported to both Solaris and Windows-2000 (a compatible successor to Windows NT) during early 2000.

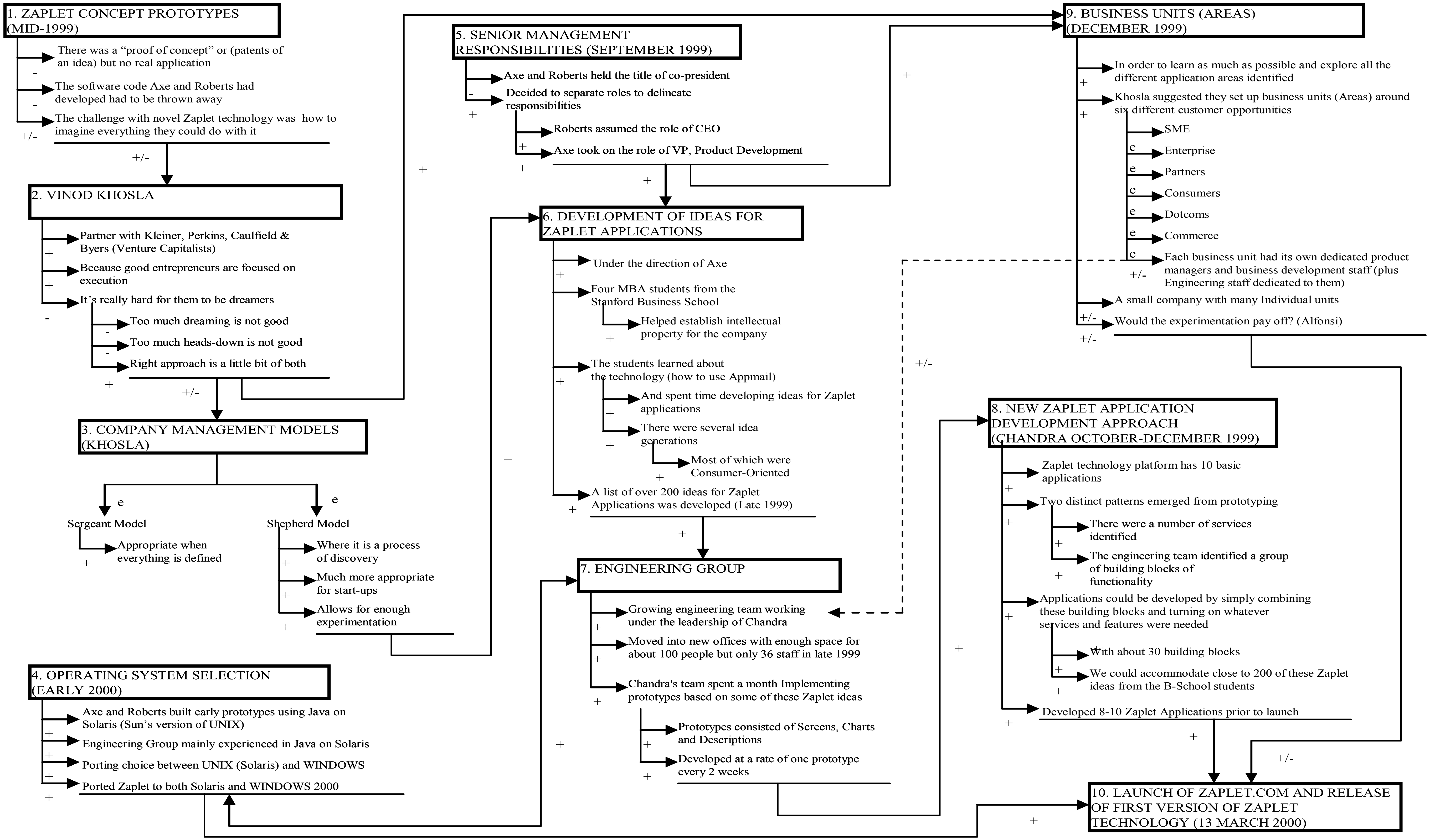


Figure 5.15 Second Revised Cognitive Map – Zaplet – Prototype-To-Product (1999-2000)
Note. This map represents etic 3 interpretation

Changes in Senior Management responsibilities (box 5) were corrected to show that Axe became Vice-President of Product Development and drove the ideas generation project undertaken by the Stanford MBA students.

The sections in the cognitive map related to the Engineering Group were re-organised into two sections. The first section on the Engineering Group (box 7) covers the rapid growth of the Group, but offers some additional insights development of some of the Zaplet application prototypes. The prototypes consisted of screens, charts and descriptions and were developed at a rate of one prototype every two weeks.

The second section covered the new Zaplet Application Development Approach (box 8), combining the Zaplet Application Prototypes and Zaplet Application sections. Most of the details remained unchanged, except that approximately 8-10 Zaplet applications were developed (at least in prototype form) prior to Zaplet's launch in March 2000.

The Business Units (Areas) section (box 9) was expanded to indicate the actual units, that is, SME, enterprise, partners, consumers, dotcoms and commerce.

Richard Alfonsi's question as to whether the experimentation with business units would pay off was justified with Zaplet's restructuring to focus on the enterprise with emphasis on integrated solutions in late 2000-01.

The list of Zaplet ideas was refined through 2000 to focus more on ERP and CRM applications which were more in alignment with the enterprise analysis outlined in the second cognitive map.

The final box (10) was updated to indicate that at least unofficially the first version of Zaplet technology was released on the launch of Zaplet.com.

The updated cognitive map adds insights into differences between the development of an application prototype and the actual application. Over the seven-eight months between meeting the venture capital company and launching the company several ideas for applications were explored, but 8 to 10 applications were actually developed.

How far the applications were developed was open to debate. According to the suggested development rate for prototypes, there were 8 to 10 prototype applications available at Zaplet's launch and these were probably released between March and June 2000.

It is likely that a substantial set of applications prototyped in the first half of 2000 would have some relation to ERP-and-CRM applications suites. The Engineering Group was working on ERP and CRM related applications before Zaplet's restructure toward an Enterprise focus from September 2000.

The cognitive map is a reminder of the often frenetic efforts undertaken to rapidly transform ideas to application prototypes and releases in tandem with creating a new company.

Hundreds of start-up companies with new Internet-based software applications undertook this journey from around 1994 through to the technology stocks correction in 2000. Developers with new software technologies for the Internet continue to go through the process of commercialising applications and setting up new companies to this day.

5.10 Extension Questions

Zaplet interviewees were asked extension questions covering Dominant Logic, Shared Vision, Leverage Points, and Strategic Marketing Principles.

5.10.1 Dominant Logic

During the interview stage of the project, various executives at different times were identified as dominating decision-making, or providing dominant logic.

Brian Axe initially articulated an unmet community opportunity into the Zaplet concept, and provided dominant logic early application development. David Roberts joined Brian Axe to produce a combined dominant logic that drove development of the first business plan for the Zaplet concept.

However, after Axe and Roberts linked up with the Venture Capital firm Kleiner, Perkins, Caulfield & Byers (KPC&B) in July 1999, the interviewees agreed that Vinod Khosla became the dominant logic for the establishment of Zaplet as a start-up company.

Khosla coached the founders (particularly Axe) to explore ideas for new Zaplet applications and to set up an organisation structure and philosophy that encouraged the development of new "killer applications".

Khosla also drove the recruitment of best-in-class Java application developers, and was quite evangelistic about Zaplet's new technology and business prospects in software and venture capital circles.

A by-product of Khosla's recruitment of a strong application development was an increasing contribution to dominant logic from the Engineering Group in 1999-2000.

Through early 2000, according to Roberts (2002), there was a strong collective sense of shared vision throughout the company with dominant logic dispersing across it.

The founders of Zaplet were supportive of dominant logic dispersing across the company as they were both saw themselves as entrepreneurs more suited to set up a company. They preferred to then hire people better than themselves to run and develop the company, while the founders would then move on to other projects in the company or to other ventures.

Following this logic, Roberts saw his role as CEO from mid-1999 though to mid-2000 as a transitional role until a new and different CEO was appointed.

The appointment of Alan Baratz as the new CEO in July 2000 heralded a shift in dominant logic toward him and his executive team. At about this time Khosla's contribution to dominant logic diminished as he moved onto new ideas and new company ventures.

Chandra (2001) believed that through 2000-01, Zaplet technology and application customers contributed significantly to dominant logic for Zaplet application development via the Product Marketing Group. "Customers" here include ISVs, System Integrators and Consulting Houses.

5.10.2 Shared Vision

Interviewees agreed that there was a strong sense of shared vision across the whole company during 1999 and 2000. Once a week, everyone at Zaplet would meet over a coffee session at the top of the stairs in the office building at Redwood Shores (there is a long straight set of stairs leading to the second floor with a large open area surrounding it).

Sharing vision through these coffee sessions was relatively easy when the company was still small (around 30-50 employees). As the company grew and specialised groups emerged, vision tended to be taken over mainly by Product Marketing and the Engineering Group.

There was no comment with regard to specific shared vision across third-parties, although Chandra (2001) indicated that for 2001, ISVs, System Integrators and Consulting Houses were sharing ideas with the Product Marketing and Engineering Groups, pointing toward Zaplet sharing its vision with third-parties.

During the interview stage of the project (late 2001 – early 2002), Zaplet was ramping up definition and implementation of Alliance Programs. Alliance Program elements considered as priorities for Zaplet include providing training programs, SDKs, Application Programming Interfaces (APIs), and other third-party application connector applications.

Zaplet also placed high priority on strengthening customer relationships (at least from a business-to-business perspective) and developing shared vision beyond Zaplet. Programs were created to encourage active "customer" participation in beta testing of new Zaplet applications.

5.10.3 Leverage Points

Several leveraged points were identified during the interviews. These points are outlined in chronological order.

5.10.3.1 Development of Zaplet Technology Prototypes Using Java Technology at Reactivity (1998 to mid-1999)

5.10.3.2 Roberts' and Axe's Presentation to Vinod Khosla at KPC&B (23 July 1999)

5.10.3.3 Recruitment of Key Managers and Developers (Second-Half 1999)

5.10.3.4 Realisation That a Set of Zaplet Building Blocks Could be Used to Create Several Applications (Late 1999)

5.10.3.5 Public Launch of Zaplet.com (13 March 2000)

5.10.3.6 Appointment of Alan Baratz to Zaplet Board of Directors (Early 2000)

5.10.3.7 Focus on Enterprise Market Segment (September 2000)

5.10.3.8 First Major Zaplet Restructuring (March 2001)

5.10.3.1 Development of Zaplet Technology Prototypes Using Java Technology at Reactivity (1998 to mid-1999)

Selection of Java technology for the application while Axe was developing the Zaplet concept in 1998-1999, seemed to be a minor decision in 1998-999, but it contributed to several major developments over the following two years.

Vinod Khosla, who listened and supported Axe and Roberts' presentation to KPC&B in July 1999, was an ex-Sun Microsystems executive who possessed very strong software development contacts in Sun Microsystems, and in particular in the Java technology area.

Khosla could see potential for the Zaplet technology to be a "killer application", in the emerging Internet world. He was also able to recruit some of the top Java developers directly out of Sun into the new Zaplet development team.

Java technology also contained sufficient features for Zaplet programmers to develop several Zaplet applications without having to incorporate many other third-party applications.

5.10.3.2 Roberts' and Axe's Presentation to Vinod Khosla at KPC&B (23 July 1999)

Roberts and Axes presentation to KPC&B in July 1999 represented a leverage point between Zaplet as an idea, and Zaplet as a start-up company.

After the presentation, the founders' focus switched from developing a concept to gaining funding for developing both a company and a set of software applications. Khosla became actively involved in encouraging the identification of ideas for Zaplet applications and establishment of a complementary company structure.

5.10.3.3 Recruitment of Key Managers and Developers (Second-Half 1999)

Roberts (2002) identified the recruitment of Mala Chandra, Samir Mitra and Steve Evans as key leverage points for the development of Zaplet as a company during the second half 1999.

Zaplet's appeal to leading-edge application developers to join them was strongly enhanced through recruitment of highly regarded application developers. Recruitment of developers with established reputations is an important factor in other case-studies in the project (for example, David Franke for Trilogy, Yarden Malka for NetDynamics).

5.10.3.4 Realisation that a set of Zaplet Building Blocks Could Be Used to Create Several Applications (Late 1999)

Chandra (2001) identified a major leverage point in late 1998, based on the realisation that the Engineering Group could develop around 30 basic Zaplet applications (or Zaplet Building Blocks) that could be then be customised for hundreds of potential applications.

It was thought at the time that a basic building-block approach could greatly increase application development efficiency and reduce the time taken to convert an idea to a prototype and then to a full application.

5.10.3.5 Public Launch of Zaplet.com (13 March 2000)

Zaplet as an idea and a company entity was disguised or placed in "stealth" mode, under the name "FireDrop", from July 1999 to March 2000. Roberts believed this veil of secrecy enabled the company to go about developing application prototypes and company infrastructure without losing ground to competition.

Subsequently, when Zaplet was launched in March 2000, the Zaplet concept was regarded as a very new technology in the marketplace, giving the organisation at least a perception of competitive advantage at its public launch.

5.10.3.6 Appointment of Alan Baratz to Zaplet Board of Directors (Early 2000)

Initial reviews of DeLacey & Leonard (2001) may imply a leverage point for the company when Baratz was appointed CEO of Zaplet in July 2000.

However, Roberts (2002) and Chandra (2001) suggested that Baratz had initiated significant changes from the time he joined Zaplet's Board of Directors early in 2000. Baratz initiated the market segment analysis projects leading to Zaplet refocusing on the enterprise market Segment in late 2000 and 2001.

5.10.3.7 Focus on Enterprise Market Segment (September 2000)

The Zaplet Board of Directors received the findings from the enterprise market segment project in September 2000, and decided that Zaplet should focus on that market. Subsequent application development focus and company restructuring in 2000-01, was largely a flow-on from this decision.

5.10.3.8 First major Zaplet Restructuring (March 2001)

Zaplet significantly cut its employee numbers in March 2001 (after reaching over 100 employees in late 2000), to focus on enterprise market opportunities, and also reflecting the downturn in e-commerce activity in 2001.

The restructure in March 2001 was the first time that the company had switched from growth to consolidation mode and was regarded by Roberts (2002) as a major leverage point.

There was a secondary consolidation in September 2001, but it was not regarded as a leverage point for Zaplet.

5.10.4 Strategic Marketing Issues

There were two distinct perspectives on strategic marketing issues that emerged from the interviews. One was a relatively technical perspective, while the other was more philosophical. Six strategic marketing issues related to these perspectives were identified during the interview stage:

5.10.4.1 Understanding User Requirements Specifications
5.10.4.2 Consolidation to a Specific Market Focus While Creating New "Killer Applications"
5.10.4.3 Product Managers and Product Marketers Seen as Marketers
5.10.4.4 Development of Customer Networks or "Ecosystems"
5.10.4.5 Application Adoption Issues
5.10.4.6 "Viral" Diffusion of Applications

5.10.4.1 Understanding User Requirements Specifications

Chandra (2001) emphasised understanding user requirements specifications as a strategic marketing issue. As Zaplet was still a young company (less than two years old), balancing customer requirements for both Zaplet technology platform and Zaplet application enhancements with Zaplet's own business vision and needs was a major strategic marketing challenge for the company.

5.10.4.2 Consolidation to a Specific Market Focus While Creating New "Killer Applications"

Roberts (2002) and Chandra (2001) noted challenges associated with undertaking market segmentation analysis while at the same time attempting to create "killer applications" drawn from an earlier idea-generation process.

The two stage market segmentation process was very difficult for Zaplet to go through immediately after over 200 potential applications were identified and the Engineering Group had identified 30 building blocks could be used to create all the applications.

Reduction of potential applications down to a set of enterprise-based "Killer Applications" was logical, given Zaplet's development resources, but represented a significant shift in mindset for the company.

5.10.4.3 Product Managers and Product Marketers Seen as Marketers

For high technology companies such as internet software houses, marketing management may be found in both product (applications) management and marketing roles.

Product managers need to possess very high levels of software knowledge – and be able to work with ISVs and other developers, listening to their application requirements and to communicate those requirements to the Engineering Group. Such relationship management activity would be regarded as marketing activity in business-to-business marketing contexts.

Product managers may have management responsibility for pricing applications and inputs for distribution channels. Product marketers may be responsible for validating pricing and promoting the application. Collectively, product management and product marketing working roles can be viewed, at least in the context of a start-up software house, as a combined marketing role.

Workman's (1993) research on marketers in high-technology firms requiring substantial technical knowledge to participate in product development activities is relevant for understanding the skills and knowledge that marketers may need in a company such as Zaplet.

5.10.4.4 Development of Customer Networks or "Ecosystems"

A more philosophical strategic marketing perspective was suggested by Roberts (2002), where networks or "ecosystems" of customers should be sought and developed.

Such an ecosystem might comprise customers with similar attributes who may work together on enterprise tasks (for example collective purchasing, and maintenance). Customers with such an ecosystem would rely on each other for various functions, referencing each other (at least in terms of technology and applications) and grow together.

5.10.4.5 Application Adoption Issues

Roberts (2002) also referred to adoption of software as a strategic marketing issue. Some customers thoroughly evaluate software applications and then adopt one or a number of preferred applications from that evaluation process. Governments were seen as good examples of customers selecting applications from such evaluation processes.

Some customers have individuals who try out new software and, if it works, usage may spread in an almost "viral' but stealthy fashion throughout the company without such an explicit evaluation process.

5.10.4.6 "Viral" Diffusion of Applications

"Viral" diffusion of software throughout companies is a strong insight – and an interesting extension of the Zaplet perspective on gaining acceptance for its application in the Consumer Market Segment in early 2000.

The concept of "viral" marketing, that is, rapid acceptance of Internet-based applications through either word-of-mouth or electronically passing on applications, was regarded by Roberts (2002) as a critical success factor in the Consumer Market Segment.

The viral marketing concept may be relevant in business-to-business environments – especially at the "chasm" and "tornado" stages of Moore's (1995) high-technology adoption cycle. Viral marketing may be relevant to acceptance of Zaplet's enterprise applications, as well as applications for the consumer segment.

5.11 Summary

The Zaplet case study provides insights into decision-making associated with the creation and commercialisation of an application technology in tandem with creating a business.

Decision-making issues associated with the creation of the Zaplet application technology, development of applications using the new technology and the launch of the Zaplet company from 1995 through to 2001, were captured and mapped in a package of DSA models, events chronology maps and cognitive maps and extension questions.

The interview process validated the package, but enabled extension of the case study timeframe to analyse, in substantial detail, a revised application development cycle for Zaplet in 2001.

Key insights from the case study include:

5.11.1 Commercialisation of an Application in Conjunction with Creating a New Business

5.11.2 Role of Venture Capitalists in Providing Leadership in Development of Ideas for Potential Applications and Providing Application Development Resources

5.11.3 Decision-making Associated with Transition from Broad Idea Generation to Highly Focused Enterprise Development

5.11.4 Searching for an Integrative Application to Support an Enterprise "Ecosystem"

5.11.1 Commercialisation of an Application in Conjunction with Creating a New Business

Decision-making was explored and mapped right through from articulation and prototyping of a technology within an incubator environment. Decisions associated with taking the technology into a "pre-company" vehicle and to commercialising the technology into various applications were mapped, including events related to Zaplet's founders pitching their technology concept to a prospective venture capital group.

Nevertheless, at the incorporation of Zaplet as a company in March 2000, Zaplet offered a technology but still had not produced any full commercial applications. Zaplet was effectively still in a pre-commercialisation state up to its release of robust enterprise-based applications in 2001.

5.11.2 Role of Venture Capitalists in Providing Leadership in Development of Ideas for Potential Applications and Providing Application Development Resources

The venture capital firm, KBC&B, and, in particular, Vinod Khosla, "bought" the concept of Zaplet's application technology, rather than the actual application in the state that it was, in July 1999. Khosla provided mentoring and encouragement for Zaplet to explore as many ideas for potential applications that could be developed using Zaplet technology concept. This approach, classified by Khosla as the "shepherd" model, was appropriate for Zaplet.

If Zaplet's technology had been sufficiently commercialised and applications defined (the "sergeant" model), then contribution from the venture capitalist may have been focused more on specific funding and management support.

Zaplet was able to develop a list of over 200 potential applications which was then submitted to its Engineering Group for prioritisation into developing application prototypes.

The Zaplet case study also highlights the role of venture capitalists providing both funding and human resources for application development. Vinod Khosla's connections with Sun Microsystems enable KPC&B to recruit high-quality application developers – delivering rapid developer capability plus significantly increased visibility and reputation for Zaplet.

Zaplet's new Engineering Group was able to articulate 30 Building Blocks from which most of the 200 potential applications could be developed.

5.11.3 Decision-making Associated with Transition from Broad Idea Generation to Highly Focused Enterprise Development

Despite Zaplet's Engineering Group possessing the capability to develop literally almost all potential applications, Zaplet's Executive Team were moving toward a strict focus on enterprise-based applications. Decision-making associated with conducting market segmentation analysis down to a set of killer applications in an enterprise segment is captured in a cognitive map in the package.

5.11.4 Searching for an Integrative Application to Support an Enterprise "Ecosystem"

Following on from creating an enterprise focus, Zaplet zoomed in on integration of online enterprise wide collaboration and communication. Actually, the discovery and articulation of collaboration and communication may be greatly assisted by mapping underlying intra – and inter – enterprise "ecosystems". The defining and development of applications using an "ecosystem" philosophy may be an area for research emerging from the case study.

5.12 Conclusions

The Zaplet case study highlights issues associated with development of technologies – and new software houses – to be used with the Internet to create Internet-based applications.

Conclusions are presented in the sections below:

5.12.1 Theoretical Issues Addressed in the Zaplet Case Study

The Zaplet case study addressed mapping strategic thought associated with application conceptualisation, development and delivery.

The extended storytelling method used for analysis in the case study, advanced hermeneutic analysis and extended DSA were effective at identifying key decision-making issues, events, and linkages.

The methodology provided mapping and validating of initial accounts of decision-making, followed up with subsequent revisions of those accounts using a systematic emic/etic representation system.

“Application conceptualisation” proved to be much more than just creating a software technology to address online group communication. That aspect of “development” was completed in the Reactivity incubator and Zaplet was able to provide a commercial version of its own technology to ISVs at its launch in March 2000.

However, further commercialisation of the technology into usable applications required mapping of the decision-making of Zaplet’s founders working with their venture capitalist to identify ideas for potential application development.

Actual application “development” initially was based on Zaplet’s Engineering Group making decisions about setting up application building blocks and systematically creating prototypes for those building blocks. Actual enterprise applications based on the Zaplet application technology were released at about the time Zaplet consolidated into an enterprise-focused company in 2001. This “development” was also captured in cognitive maps.

5.12.2 Zaplet Case Study Limitations

The Zaplet case study was limited to two interviews with two key actors in DeLacey & Leonard’s (2001) case study. However, full review, validation and updates were provided by the interviewees. Additional interviews and interviewees may have assisted in illuminating some insights.

Attempts were made to gain interviews with representatives from Zaplet’s venture capital company, which may have provided additional insights into the venture capital company’s (KPC&B) perceptions of Zaplet’s technology and KPC&B’s actual contributions to Zaplet.

5.12.3 Contributions from the Zaplet Case Study To Strategic Sensemaking

The Zaplet case study explores Decision Systems Analysis (DSA) through mapping the decision-making of changes over time in shared vision, and application technology development including applications developed using the technology. Sensemaking is further enhanced and illuminated through incorporation of extended DSA into a powerful five-level hermeneutic analysis.

The case study initially mapped Zaplet’s decision-making associated with commercialising a technology – at least to an early prototype stage. Mapping then shifted toward creating as many ideas for potential applications built on the Zaplet application technology as possible. The Engineering Group continued to make decisions facilitating the development of prototypes for most of these ideas.

At the Zaplet Executive Team level, decision-making then shifted toward choosing a viable market segment for Zaplet as a real company. Inputs from the interview process enabled more detailed mapping of Zaplet's decision-making processes for an organisation focused on enterprise applications.

5.12.4 Contributions from the Zaplet Case Study to Practice

The Zaplet case study provides valuable insights into challenges for entrepreneurial application developers seeking to create, commercialise and deliver software applications with viable potential market demand.

Prospective developers may find incubators useful to garner resources to create software applications, but they most likely will still have to pitch their concepts to investors for funding or resources to commercialise their applications.

The study highlights important contributions that venture capitalists can offer to emergent technology companies beyond funding. While funding was critical for Zaplet's growing into a viable company, their venture capital company provided valuable support and assistance with management skills, encouragement to brainstorm potential application ideas, and assistance with the recruitment of very high quality application developers.

The case study also highlights how contributions from venture capitalists may change over time and as a company restructures its business and market focus.

5.12.5 Zaplet Case Study Endnote

Zaplet restructured in mid-2002, extending its focus beyond enterprise applications to security and defence related applications, opening a business office in Washington DC.

Zaplet 3 was launched on 1 October 2003. Zaplet's revised focus and mission in 2003 (as stated with the launch of Zaplet 3) is:

"Zaplet^{tm}, Inc. is a privately held enterprise software company and creator of Zaplet 3, business process management software that brings application functionality directly to a user's e-mail inbox to complete business processes. Zaplets are task-based applications that can be built or modified by snapping together reusable components and applying rules to define and manage process flow. Zaplet 3 offers visibility and audit ability into mission-critical processes such as in intelligence gathering and dissemination, customer relationship management, product lifecycle management and HR" (Zaplet 2003).

Chapter 6: Case Study – NetDynamics

Abstract

The NetDynamics case study offers an account of rapid application development using two storytelling methods for analysis – an **advanced hermeneutic framework**, and an **extended form of decision systems analysis (DSA)** incorporating **cognitive mapping**.

The account explores decision-making for a start-up software house with a concept for application possessing potential to become a fundamental component of an emerging computer-mediated environment (CME – in the case study – the World Wide Web) - and a foundation for future enterprise e-commerce applications.

The case study provides insights into a group of three application developers working, testing and updating each other's applications, with a common vision to rapidly produce an advanced multimedia CME.

Key findings from the case study highlight issues with rapid application development across a number of new application technologies; variations in actual relationships between developers working with each other's applications; roles for System Integrators as extended developers and marketers for small software houses; challenges with managing rapidly growing enterprise customer bases; and whether start-up software houses with enterprise-based applications can grow to a critical mass to provide enterprise customer support, or end up being acquired by larger vendors with established enterprise customer support infrastructure.

6.1 Background

While at the NCSA at the University of Illinois, Marc Andreessen and other students developed the first application capable of collecting and reading multimedia information from the World-Wide Web in 1993.

Jim Clark, an ex-CEO of Silicon Graphics Inc., recognised the huge potential for the "NCSA Mosaic" browser to become standard for accessing the World-Wide Web and on April 4, 1994, with Andreessen, he founded Netscape.

Netscape's browser gained rapid user acceptance during 1994-95, spawning a massive need for applications to enhance the capabilities of the browser. However, the window of opportunity was perceived at the time to be very short, with developers rushing to release brand new applications with 60 to 90 day cycles.

The NetDynamics case study chapter explores the development and delivery of NetDynamics, which developed an application to design web pages with links to company databases during 1995-96. The timeframe parallels the incorporation of Java technology into the Netscape browser, thus providing a platform for the development of several new applications to improve browser features and capability, including NetDynamics' application.

NetDynamics was included in the final case study list for the project as it met the initial criteria:

- Software has to be "beta"-tested and commercially introduced
  - NetDynamics 1.0 was already commercially available, although the case study focused on the development of NetDynamics 2.0 through to release and was considered to have met this criteria
- Case-studies were written by well-known authors
  - The original case study is a Harvard Business School Case study written by Alan MacCormack (see Red Hat case study – a professor for that case study but a doctoral student for the NetDynamics case study) and professor Marco Iansiti, both recognised as experts in new product development – especially in the area of internet-based software applications
- The application development process studied must be over a timeframe of more than 12 months
  - the original case study covers a timeframe of just over 12 months from conception of application to delivery in 1995-96
- Names of case study companies and participants can be disguised if requested
  - Permission was gained from Zack Rinat and Kenneth Ebbs, both for a combination of face-to-face and digitally recorded telephone interviews and use of that information in the research project, without requirement for disguise
- Applications must be Internet-based

- NetDynamics 2.0 represents one of the most fundamental Internet-based applications ever developed. It enabled web-pages to be used to inquire and use information from databases on local and remote servers – paving the way for a critical component of enterprise e-commerce activity

Additional case study selection criteria included:

- Size: Small
- Success: Was acquired and further developed by Sun Microsystems iPlanet Group; became part of Sun One's Web Services Offering
- Disruptive Technology: No, linked to Java, Web Browsers and ultimately Web Services – which may an emerging disruptive technology
- Harvard Business School Case: "Living On Internet Time: Product Development at Netscape, Yahoo!, Net Dynamics and Microsoft", Iansiti & MacCormack (1996)

The HBS case study represents the best and most detailed description of the actual development of a set of Internet-based applications out of all the Harvard Business School Case-studies reviewed for the project. Variations on the HBS NetDynamics case study can be found in Iansiti (1995), Iansiti (1998) and Iansiti & MacCormack (1998).

A package was created based on Iansiti & MacCormack's (1996) case study, made up of one DSA model, one events chronology map and three cognitive maps.

The DSA model and maps in this case were updated through multiple interviews with one of the founders of NetDynamics, and one of the main developers of NetDynamics 2.0 (see Rinat 2002a & b; Ebbs 2002a & b). Extension questions were addressed during the follow-up interviews.

Although Iansiti & MacCormack's (1996) case study focuses on the 1995-96 timeframe, NetDynamics continued to develop new versions of its application through 1996, 1997 and 1998.

The company was acquired by Sun Microsystems in July 1998, but continued development on its own applications through to March 1999, when it released NetDynamics 6.0.

NetDynamics was absorbed into Sun's iPlanet Group in March 1999, and the iPlanet Group (with other groups) was superseded by Sun One1 in March 2002.

Developments from 1997 though to 1999 were incorporated from comments and insights obtained from interviews and were noted in an updated events chronology map.

The outline and chapter structure for the NetDynamics case study are presented in Figure 6.1.

6.2 Theoretical Issues (or Questions) for the NetDynamics Case Study

The main issue for the NetDynamics case study is mapping the decision-making associated development of a potential "killer application" at a very early stage of emergence of a new computer-mediated environment that is, the World-Wide Web.

During 1995-1996, rapid advances in Web browser technology opened up a small window for some application developers to create and integrate various applications that would become fundamental components of that technology.

The case study traces and maps NetDynamics' decision-making for successfully developing an idea right through to a fundamental Web Technology component in a timeframe denoted by Iansiti & MacCormack (1996) as "Internet-Time" that is, a very rapid development cycle of six months.

As with several other cases in the project, not only are developers attempting to create an application in "Internet Time" but they are also attempting to create a business in the same rapid timeframe.

An extended perspective of the NetDynamics case study out to 1999 confronts issues beyond the initial application and new business development and into sustainability and attaining critical mass to a base of rapidly expanding large enterprise customers.

The storytelling methodology used for analysis in the case study – an advanced hermeneutic analysis, including an extended form of **decision systems analysis** (DSA) incorporating **cognitive mapping** – were effective at identifying key decision-making issues, events, and linkages.

The methodology provided mapping and validating of initial accounts of decision-making, followed up with subsequent revisions of those accounts using a systematic emic/etic representation system.

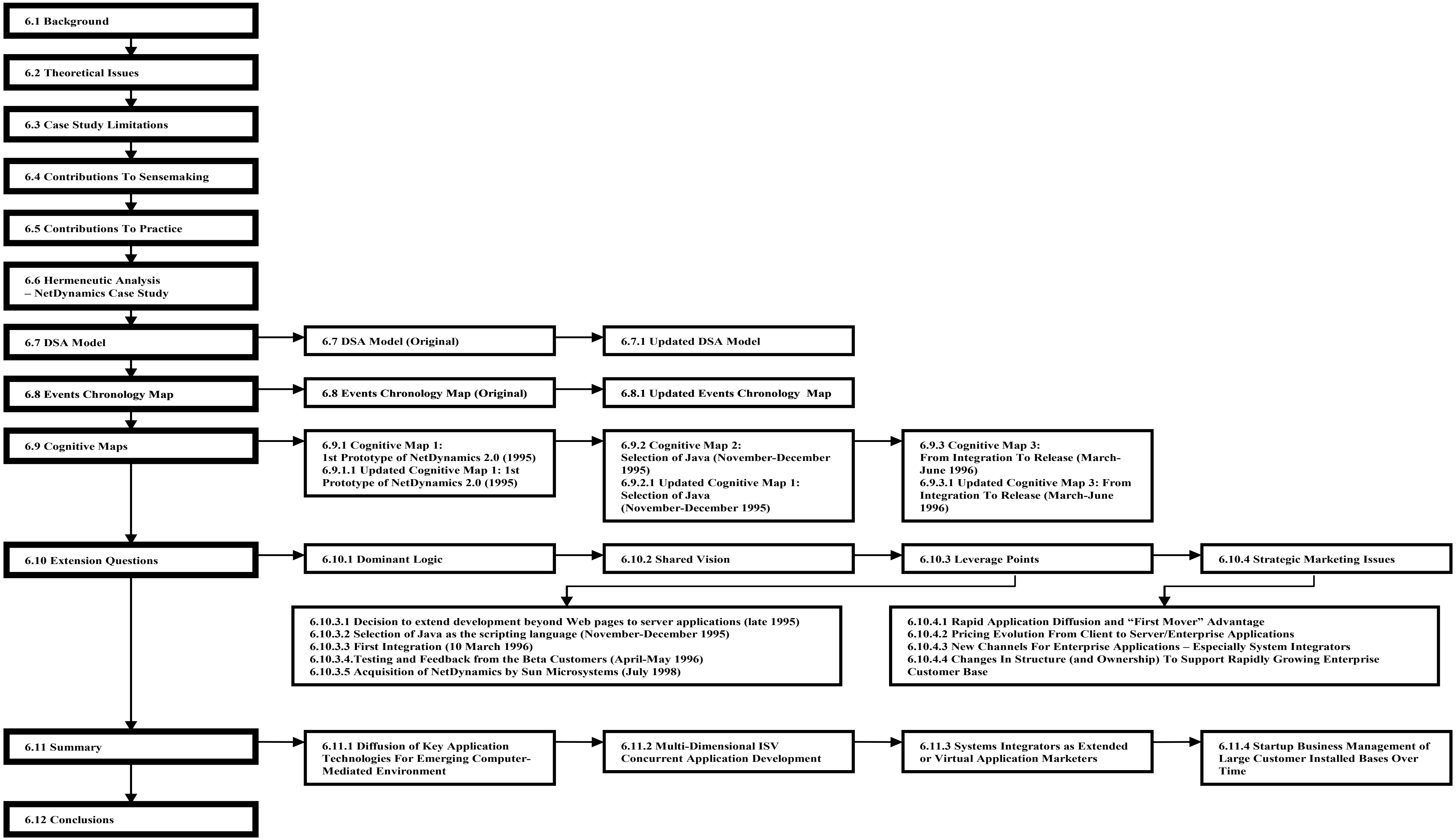


Figure 6.1 NetDynamics Case Study – Chapter Structure

Hermeneutic analysis for the NetDynamics case study extended one more level further than the other five case-studies, as a selection of DSA models and cognitive maps from the case was sent to a panel of experts to evaluate the overall DSA modelling and mapping methodology from a humanistic inquiry perspective (see Hirschmann 1986).

6.3 NetDynamics Case Study Limitations

The NetDynamics case study contains a complete set of multiple interviews with two respondents who were directly involved with the development of the NetDynamics application in 1995-96.

The NetDynamics case study would have benefited from additional interviews, including the NetDynamics Chief Scientist who was a key decision-maker and hands-on developer of the NetDynamics application.

The study may have also benefited from interviews with developers from Sun Microsystems and Netscape who were all developing applications with each other and NetDynamics.

6.4 Contributions from the NetDynamics Case Study to Strategic Sensemaking

The NetDynamics case study explored decision-making issues, events, and linkages using the storytelling method developed for the project. An advanced hermeneutic analysis provided several levels of emic/etic representation of decision-making, highlighting key case study issues.

The Hermeneutic framework incorporated an extended form of Decision Systems Analysis (DSA) highlights specific subsets of decision-making, through mapping of decision-making from original accounts with revisions, through subsequent contact and discussions with original decision-makers. The extended DSA also explored changes over time in shared vision and new application development.

The NetDynamics case study contributes to strategic sensemaking through presentation of mapping DSA in three dimensions. A first dimension addresses very rapid product development (“product development in Internet Time”).

A second dimension focuses on mapping DSA for transition from an embryonic business to acquisition by a large high-technology company. A third-party dimension covers the mapping of DSA issues associated with managing a rapidly growing base of enterprise business accounts.

6.5 Contributions from the NetDynamics Case Study to Practice

The NetDynamics case study provides insights for application developers seeking to create new businesses out of the development of new products for revolutionary or discontinuous infrastructure.

The case study also provides insights for application developers and marketers managing product development within a multiple ISV strategic alliance arrangements. The case provides distinct insights in a scenario where three application developers effectively act as “beta-testers” for each other’s new and emerging applications.

6.6 Hermeneutic Interpretation – NetDynamics Case study

The NetDynamics case study offers an application of the proposals described in Chapter 1 for advancing hermeneutic research in B2B decision-making.

Six levels of hermeneutic analysis were developed to address decision-making issues, events, and linkages in the NetDynamics case study. These levels are presented in Exhibits 6.1 to 6.5 and 6.7.

The original case study by Iansiti & MacCormack (1996) serves as the etic 1 report of the case study – although there are elements of emic 1 reporting in the form of direct quotes from decision-makers in the account. Quotes and perspectives from decision-makers were encapsulated the Level 1 emic analysis in Exhibit 6.1.

Exhibit 6.1 Level I Analysis of NetDynamics Case

Emic 1 view:
•Development of a key Web-based application technology in "Internet-Time"
•"Proof-of-concept" sufficient for analysts and investors to support NetDynamics
• Java technology was not the final choice for development language when project commenced
• NetDynamics noted that Java was likely to be incorporated into next Netscape Browser Release and took a risk on developing NetDynamics to work with both Java and Netscape technology and applications
•Project was successfully completed within 90 days

↓

1995-1996. Initial mental model:
•Identification of Potential Critical Internet-based business application (Web Page access to Database records)
•Extension of application from UNIX into cross-platform environment
•Flexible about application development language
•Must deliver as part of and by the next release of Netscape Communications Web Browser
•Must have "proof-of concept" for investors in 30 days
•Must be prepared to completely redesign the application with different languages if necessary

Decisions-actions:
•RDC as Spider UNIX (1995) –Renamed to NetDynamics V1 mid-1995; Plan to redevelop for both UNIX and Windows; Plan revised to make new application "cross-platform" and operating system independent; Software Engineers recruited mainly with Windows experience; Engineers "learn' Java sufficiently quickly to refocus Project toward that technology; "Proof-of-Concept" demonstrated successfully (Mar 1996); application revised daily in development phase; NetDynamics released June-July 1996

Exhibit 6.2 Level II Analysis of NetDynamics Case

Etic 1 view:
•Development of applications in "Internet-time"
•Daily Builds of application with internal and 3rd parties inputs
•Commenced actual development before choosing application language
•Rush to incorporate key new application into upcoming release of industry standard Web-Browser
•Importance of delivering an effective proof-of-concept to analysts and supporters
•New approach to application development (one case within a group to demonstrate "Internet-Time application development)

Emic 1 view:
•Development of a key Web-based application technology in "Internet-Time"
•"Proof-of-Concept" sufficient for analysts and investors to support NetDynamics
• Java Technology was not final choice for development language when project commenced
• NetDynamics noted that Java was likely to be incorporated into next Netscape Browser Release and took a risk on developing NetDynamics to work with both Java and Netscape technology and applications
•Project was successfully completed within 90 days

1995-1996. Initial mental model:
•Identification of Potential Critical Internet-Based Business Application (Web Page access to Database records)
•Extension of Application from Unix into cross-platform environment
•Flexible about application development language
•Must deliver as part of and by the next release of Netscape Communications Web Browser
•Must have "proof-of concept" for investors in 30 days
•Must be prepared to completely redesign the application with different languages if necessary

Decisions-actions::
•RDC as SPIDER UNIX (1995) –Renamed to NetDynamics V1 mid-1995;
•; Plan to redevelop for both UNIX and WINDOWS; Plan revised to make new application "cross-platform" and operating system independent; Software Engineers recruited mainly with Windows experience; Engineers "learn' Java sufficiently quickly to refocus Project toward that technology; "Proof-of-Concept" demonstrated successfully (Mar 1996); application revised daily in development phase; NetDynamics released June/July 1996

Key case study issues as articulated by Iansiti & MacCormack (1996) were added to the emic 1 data in the Level II analysis and presented as etic 1 data in Exhibit 6.2. These issues included:

- Development of applications in "Internet-time"
- Daily Builds of application with internal and third-party inputs
- Commenced actual development before choosing application language

- Rush to incorporate key new application into upcoming release of industry standard Web-Browser
- Importance of delivering an effective proof-of-concept to analysts and supporters
- New approach to application development (one case within a group to demonstrate "Internet-Time" application development).

The Level II analysis also explored mental models and decisions/actions associated with these issues.

The third level of analysis, as presented in Exhibit 6.3, is an etic 2 representation based on the current researcher's summarisation of event milestones and the emic 1 sensemaking views identified in the data in the Iansiti & MacCormack (1996) case study. Subsequently to developing Exhibit 6.3, emic 2 and etic 2 interpretations along with DSA and event maps provide revisions to Exhibit 6.3.

Iansiti & MacCormack's (1996) coverage of NetDynamics' selection of application development technology (essentially the Java scripting language), and detailed description of development of the NetDynamics 2.0 application, was more than adequate for the development of an initial (etic 2) representative DSA model and to undertake further analysis on application development and delivery. The initial DSA, event chronology, and cognitive maps reported below offer details supporting Exhibit 6.3. Etic 2 perspectives are mainly based from the initial DSA model, initial events chronology map, and initial cognitive maps.

Key etic 2 issues included:

- NetDynamics application is a "killer application" and a fundamental building block in the development of e-commerce application suites
- NetDynamics application was a foundation for the development of Web Services
- Was NetDynamics really as flexible about application language selection as claimed in the case?
- Had NetDynamics actually already decided about Java before starting the project?
- Triple ISV beta-testing and application development

Exhibit 6.3 Level III Analysis of NetDynamics Case

Etic 1 view:
•Development of applications in "Internet-time"
•Daily Builds of application with internal and 3rd parties inputs
•Commenced actual development before choosing application language
•Rush to incorporate key new application into upcoming release of industry standard Web-Browser
•Importance of delivering an effective proof-of-concept to analysts and supporters
•New approach to application development (one case within a group to demonstrate "Internet-Time application development)

Etic 2 view:
•NetDynamics application is a "killer application" and a fundamental building block in development of e-commerce application suites
•NetDynamics applications was a foundation for the development of Web Services
•Were NetDynamics really as flexible about application language selection as claimed in the case?
•Had NetDynamics actually already decided about Java before starting the project?
•Triple ISV Beta-Testing and Application Development

Emic 1 view:
•Development of a key Web-based application technology in "Internet-Time"
•"Proof-of-Concept" sufficient for analysts and investors to support NetDynamics
• Java Technology was not final choice for development language when project commenced
• NetDynamics noted that Java was likely to be incorporated into next Netscape Browser Release and took a risk on developing NetDynamics to work with both Java and Netscape technology and applications
•Project was successfully completed within 90 days

1995-1996. Initial mental model:
•Identification of Potential Critical Internet-Based Business Application (Web Page access to Database records)
•Extension of Application from Unix into cross-platform environment
•Flexible about application development language
•Must deliver as part of and by the next release of Netscape Communications Web Browser
•Must have "proof-of concept" for investors in 30 days
•Must be prepared to completely redesign the application with different languages if necessary

Decisions-actions:
•RDC as SPIDER UNIX (1995); –Renamed to NetDynamics V1 mid-1995; Plan to redevelop for both UNIX and WINDOWS; Plan revised to make new application "cross-platform" and operating system independent; Software Engineers recruited mainly with Windows experience; Engineers "learn' Java sufficiently quickly to refocus Project toward that technology; "Proof-of-Concept" demonstrated successfully (Mar 1996); application revised daily in development phase; NetDynamics released June/July 1996

Exhibit 6.4 Level IV Analysis of NetDynamics Case

Etic 1 view:
- Development of applications in "Internet-time"
- Daily Builds of application with internal and 3rd parties inputs
- Commenced actual development before choosing application language
- Rush to incorporate key new application into upcoming release of industry standard Web-Browser
- Importance of delivering an effective proof-of-concept to analysts and supporters
- New approach to application development (one case within a group to demonstrate "Internet-Time application development)

Emic 2 view:
- Reaffirmed all of Emic 1 view with extensions for details on:
  - Initial RDC of SPIDER
  - Selection of Software Engineers
  - Details of "Proof-Of Concept"
- High risks of using Java in Beta-Testing stages
- High uncertainties associated with determination and incorporation of features and applications in new Netscape Browser
- Go with incorporation of NetDynamics into new Browser in 60-90 days or next version in 12 months time?
- Huge challenges for small start-ups to manage rapidly growing enterprise installed base
- How to price an emerging operating system independent enterprise application

Emic 1 view:
- Development of a key Web-based application technology in "Internet-Time"
- "Proof-of-Concept" sufficient for analysts and investors to support NetDynamics
- Java Technology was not final choice for development language when project commenced
- NetDynamics noted that Java was likely to be incorporated into next Netscape Browser Release and took a risk on developing NetDynamics to work with both Java and Netscape technology and applications
- Project was successfully completed within 90 days

Etic 2 view:
- NetDynamics application is a "killer application" and a fundamental building block in development of e-commerce application suites
- NetDynamics applications was a foundation for the development of Web Services
- Were NetDynamics really as flexible about application language selection as claimed in the case?
- Had NetDynamics actually already decided about Java before starting the project?
- Triple ISV Beta-Testing and Application Development

1995-1996. Initial mental model:
- Identification of Potential Critical Internet-Based Business Application (Web Page access to Database records)
- Extension of Application from Unix into cross-platform environment
- Flexible about application development language
- Must deliver as part of and by the next release of Netscape Communications Web Browser
- Must have "proof-of concept" for investors in 30 days
- Must be prepared to completely redesign the application with different languages if necessary

1996-1999. Extended mental model:
- Expansion toward a key enterprise e-commerce application
- Funding and Enterprise installed based management issues prompt support from Sun Microsystems

Decisions-actions:
- RDC as SPIDER UNIX (1995) –Renamed to NetDynamics V1 mid-1995; Plan to redevelop for both UNIX and WINDOWS; Plan revised to make new application "cross-platform" and operating system independent; Software Engineers recruited mainly with Windows experience; Engineers "learn' Java sufficiently quickly to refocus Project toward that technology; "Proof-of-Concept" demonstrated successfully (Mar 1996); application revised daily in development phase; NetDynamics 2.0 released June/July 1996; NetDynmaics V3.0 released Oct 1996; NetDynamics V4.0 released Dec 1997; Sun Acquires NetDynmaics Jul 1998; NetDynamics V5.0 released Mar 1999 – becomes I-Planet IAS Application

The etic 2 DSA model and maps for the case were updated following extensive questioning of the accuracy and completeness of the original Iansiti & MacCormack's (1996) case – the collected data represents emic 2 material and was added during the fourth level of analysis (See Exhibit 6.4).

Thus, additional (emic 2) data were collected for etic 3 description and interpretation of the NetDynamics decision-making process as reported in the original case study and for a period of over three years beyond that reported by Iansiti & MacCormack. Emic 2 data consists of responses from interviews with the founder of NetDynamics and one of the chief developers of the NetDynamics 2.0 application (see Rinat 2002a & 2002b; Ebbs 2002a & 2002b).

Exhibit 6.5 is a fifth level of analysis with the addition of etic 3 data summarising emic 2 interpretations of mental models and events covered in the original case study as well as the work completed for the etic 2 interpretation – including the DSA, event chronology, and cognitive maps developed for the etic 2 interpretation.

The revised DSA, event chronology, and cognitive maps presented in subsequent sections below follow from the emic 2 interpretations and these maps are part of the etic 3 interpretation.

The new data from the emic 2 and etic 3 rounds of interpretation validated etic 2 data and highlighted new insights related to:

- Concurrent ISV application development
- Diffusion of key application technologies for emerging computer mediated environments (CMEs)
- Systems integrators as extended virtual application marketers
- Paths and transitions for Start-ups with enterprise applications
- A great account of genesis of Web Services

The NetDynamics case study was selected for extraction of a package of cognitive maps for expert validation of the development of the maps. A subset of cognitive maps were packaged with an Expert Auditors Evaluation Form (EAEF), based on a humanistic inquiry evaluation framework.

Exhibit 6.5 Level V Analysis of NetDynamics Case

Etic 1 view:
•Development of applications in "Internet-time"
•Daily Builds of application with internal and 3rd parties inputs
•Commenced actual development before choosing application language
•Rush to incorporate key new application into upcoming release of industry standard Web-Browser
•Importance of delivering an effective proof-of-concept to analysts and supporters
•New approach to application development (one case within a group to demonstrate "Internet-Time application development)

Emic 2 view:
•Reaffirmed all of Emic 1 view with extensions for details on:
- Initial RDC of SPIDER
- Selection of Software Engineers
- Details of "Proof-Of Concept"

•High risks of using Java in Beta-Testing stages
•High uncertainties associated with determination and incorporation of features and applications in new Netscape Browser
•Go with incorporation of NetDynamics into new Browser in 60-90 days or next version in 12 months time?
•Huge challenges for small start-ups to manage rapidly growing enterprise installed base
•How to price an emerging operating system independent enterprise
•application

Emic 1 view:
•Development of a key Web-based application technology in "Internet-Time"
•"Proof-of-Concept" sufficient for analysts and investors to support NetDynamics
• Java Technology was not final choice for development language when project commenced
• NetDynamics noted that Java was likely to be incorporated into next Netscape Browser Release and took a risk on developing NetDynamics to work with both Java and Netscape technology and applications
•Project was successfully completed within 90 days

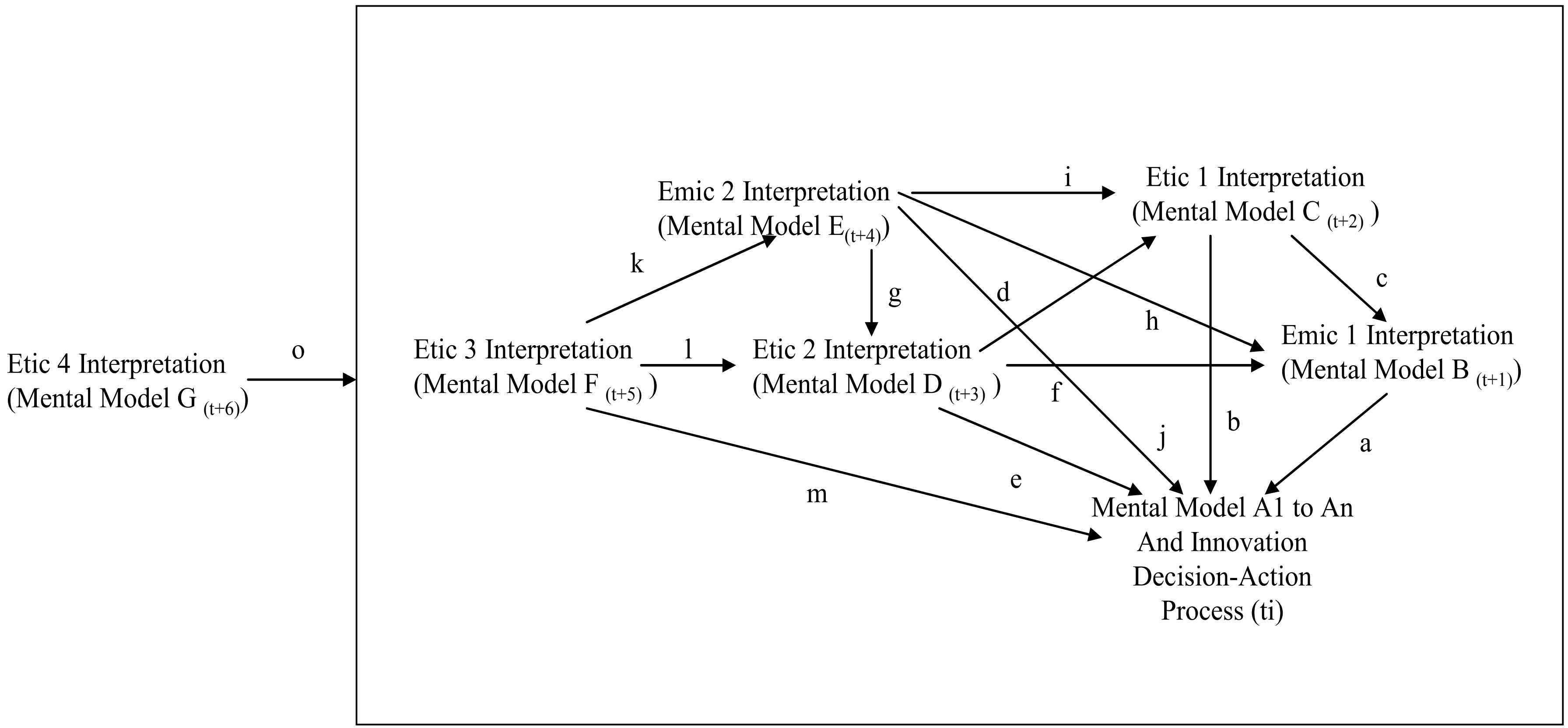
Etic 3 view:
•Etic 2 views validated plus:
- Concurrent ISV Application Development
- Diffusion of key Application Technologies for emerging computer mediated environments (CME's)
- Systems Integrators As Extended Virtual Application Marketers
- Paths and transitions for Start-ups with enterprise Applications

•A great account of genesis of Web Services

Etic 2 view:
•NetDynamics application is a "killer application" and a fundamental building block in development of e-commerce application suites
•NetDynamics applications was a foundation for the development of Web Services
•Were NetDynamics really as flexible about application language selection as claimed in the case?
•Had NetDynamics actually already decided about Java before starting the project?
•Triple ISV Beta-Testing and Application Development

1995-1996. Initial mental model:
•Identification of Potential Critical Internet-Based Business Application (Web Page access to Database records)
•Extension of Application from Unix into cross-platform environment
•Flexible about application development language
•Must deliver as part of and by the next release of Netscape Communications Web Browser
•Must have "proof-of concept" for investors in 30 days
•Must be prepared to completely redesign the application with different languages if necessary
1996-2001. Extended mental model:
•Expansion toward a key enterprise e-commerce application
•Funding and Enterprise installed based management issues prompt support from Sun Microsystems
Decisions-actions:
•RDC as SPIDER UNIX (1995) –Renamed to NetDynamics V1 mid-1995;
Plan to redevelop for both UNIX and WINDOWS; Plan revised to make new application "cross-platform" and operating system independent; Software Engineers recruited mainly with Windows experience; Engineers "learn' Java sufficiently quickly to refocus Project toward that technology; "Proof-of-Concept" demonstrated successfully (Mar 1996); application revised daily in development phase; NetDynamics 2.0 released June/July 1996; NetDynmaics V3.0 released Oct 1996; NetDynamics V4.0 released Dec 1997; Sun Acquires NetDynmaics Jul 1998; NetDynamicsV5.0 released Mar 1999 – becomes I-Planet IAS Application becomes part if SUN ONE Web Services (2001)

Exhibit 6.6 Expert Validation of DSA Approach (Selected DSA Instruments)



Etic 4 view:
•Ethnographic Validation of Cognitive Maps Supporting Etic 2 analysis– by expert "judges"
•Generally validated the cognitive mapping process
•Judges looked forward to full Etic 3 conclusions and insights drawn from further analysis of Emic 2 views and inputs – beyond models, maps and early notes of findings

Exhibit 6.7 Expert Validation of DSA Approach (Selected DSA Instruments) - NetDynamics Case Analysis

Etic 1 view:
•Development of applications in "Internet-time"
•Daily Builds of application with internal and 3rd parties inputs
•Commenced actual development before choosing application language
•Rush to incorporate key new application into upcoming release of industry standard Web-Browser
•Importance of delivering an effective proof-of-concept to analysts and supporters
•New approach to application development (one case within a group to demonstrate "Internet-Time application development)

Emic 2 view:
•Reaffirmed all of Emic 1 view with extensions for details on:
- Initial RDC of SPIDER
- Selection of Software Engineers
- Details of "Proof-Of Concept"

•High risks of using Java in Beta-Testing stages
•High uncertainties associated with determination and incorporation of features and applications in new Netscape Browser
•Go with incorporation of NetDynamics into new Browser in 60-90 days or next version in 12 months time?
•Huge challenges for small start-ups to manage rapidly growing enterprise installed base
•How to price an emerging operating system independent enterprise
•application

Emic 1 view:
•Development of a key Web-based application technology in "Internet-Time"
•"Proof-of-Concept" sufficient for analysts and investors to support NetDynamics
• Java Technology was not final choice for development language when project commenced
• NetDynamics noted that Java was likely to be incorporated into next Netscape Browser Release and took a risk on developing NetDynamics to work with both Java and Netscape technology and applications
•Project was successfully completed within 90 days

Etic 3 view:
•Etic 2 views validated plus:
- Concurrent ISV Application Development
- Diffusion of key Application Technologies for emerging computer mediated environments (CME's)
- Systems Integrators As Extended Virtual Application Marketers
- Paths and transitions for Start-ups with enterprise Applications

•A great account of genesis of Web Services

Etic 2 view:
•NetDynamics application is a "killer application" and a fundamental building block in development of e-commerce application suites
•NetDynamics applications was a foundation for the development of Web Services
•Were NetDynamics really as flexible about application language selection as claimed in the case?
•Had NetDynamics actually already decided about Java before starting the project?
•Triple ISV Beta-Testing and Application Development

1995-1996. Initial mental model:
•Identification of Potential Critical Internet-Based Business Application (Web Page access to Database records)
•Extension of Application from Unix into cross-platform environment
•Flexible about application development language
•Must deliver as part of and by the next release of Netscape Communications Web Browser
•Must have "proof-of concept" for investors in 30 days
•Must be prepared to completely redesign the application with different languages if necessary

1996-2001. Extended mental model:
•Expansion toward a key enterprise e-commerce application
•Funding and Enterprise installed based management issues prompt support from Sun Microsystems

Decisions-actions:
•RDC as SPIDER UNIX (1995) –Renamed to NetDynamics V1 mid-1995; Plan to redevelop for both UNIX and WINDOWS; Plan revised to make new application "cross-platform" and operating system independent; Software Engineers recruited mainly with Windows experience; Engineers "learn' Java sufficiently quickly to refocus Project toward that technology; "Proof-of-Concept" demonstrated successfully (Mar 1996); application revised daily in development phase; NetDynamics 2.0 released June/July 1996; NetDynmaics V3.0 released Oct 1996; NetDynamics V4.0 released Dec 1997; Sun Acquires NetDynmaics Jul 1998; NetDynamicsV5.0 released Mar 1999 – becomes I-Planet IAS Application becomes part if SUN ONE Web Services (2001)

Responses from the EAEFs were classified as etic 4 data (see Exhibit 6.6 for the conceptual depiction of a Level VI hermeneutic analysis). Level VI analysis for the NetDynamics case study was presented in Exhibit 6.7 – and represents ethnographic validation, by expert "judges", of cognitive maps supporting the etic 2 analysis.

Generally the judges validated the cognitive mapping process used in the case study (and by inference, the research project). Nevertheless, at the time the EAEFs were sent to the judges, final etic 3 written conclusions and summaries had not been completed. The judges looked forward to full etic 3 conclusions and insights drawn from further analysis of emic 2 views and inputs – beyond the updated maps and notes at that stage of the research project (December 2003).

Overall, the six-level hermeneutic framework for the NetDynamics case study provided a strong and comprehensive framework for capturing and extending dynamic sensemaking for NetDynamics' decision-makers – with the sixth level of analysis providing a critical supporting validation of not just the cognitive mapping processes associated with NetDynamics' decision-making, but for the mapping process as applied throughout the whole research project.

6.7 DSA Model

The initial DSA model was developed from the "NetDynamics" Section of the Iansiti & MacCormack (1996) case study to NetDynamics (see Iansiti & MacCormack 1996, pp. 6-9). The DSA model was constructed out of a rich description of the application development processes at NetDynamics, and is presented in Figure 6.2.

Several groups have inputs to set application objectives and the Initial Target Release Date (box 2). Project Teams are established within NetDynamics and will manage the application process. There are also inputs from selected lead-users of the NetDynamics application including ISVs and selected corporate customers (box 4).

Application Suppliers or ISVs to NetDynamics appear to have significant inputs at various stages of the development (box 5), while NetDynamics may have to work closely with them to resolve application bugs.

As NetDynamics were using very new applications for their development, it was on the one hand a "beta customer" for that ISV, but was also an ISV for its own lead-users. When very new applications are being developed using very new technologies, there may be a blurring of whom is an ISV or a "beta customer" or a lead-user!

Iansiti & MacCormack (1996) highlight parallel development and production streams, with very early integration of code and testing from the two streams (boxes 6 and 7), as a distinguishing feature of Internet application development.

However, early integration of code does not necessarily signify that the architectural design for the application has been finalised. In fact, the streams were integrated daily enabling incorporation of the very latest versions (beta and/or final releases) of applications such as Java and Netscape's Navigator.

Although Java was first demonstrated in 1992, its initial objectives were to support cable television and mobile communication devices. Java was refocused toward being a web-specific technology in 1994.

During 1995 Sun and Netscape worked on incorporating Java technology into Netscape Navigator. Netscape Navigator 2.0, launched in March 1996, incorporated some basic Java technology features and the JavaScript language. More advanced Java technology features such as Applets to enhance animation and other multimedia effects, were incorporated into Netscape's Navigator 3.0, launched in August 1996.

beta releases (box 8) are distributed to customers who also use the application in their development and/or testing activities, plus suggesting bug fixes and further updates. These inputs were incorporated into daily rebuilds of the applications, some of which would be released as further beta releases. Several beta versions are rapidly released and tested.

An important feature of the NetDynamics development process is the training of customers in the application (box 9). A smaller set of lead-users in conjunction with training are not only involved in rigorous testing but also asked to rate current and proposed applications features.

Although an initial release date is set at the start of the development process, it is typically revised during application development. In theory, there could be a constant flow of rebuilds and updates electronically distributed to customers without an official release or launch date.

For example, Microsoft provides, through electronic distribution via the World-Wide Web, a constant flow of application updates for its Windows XP and Office XP applications.

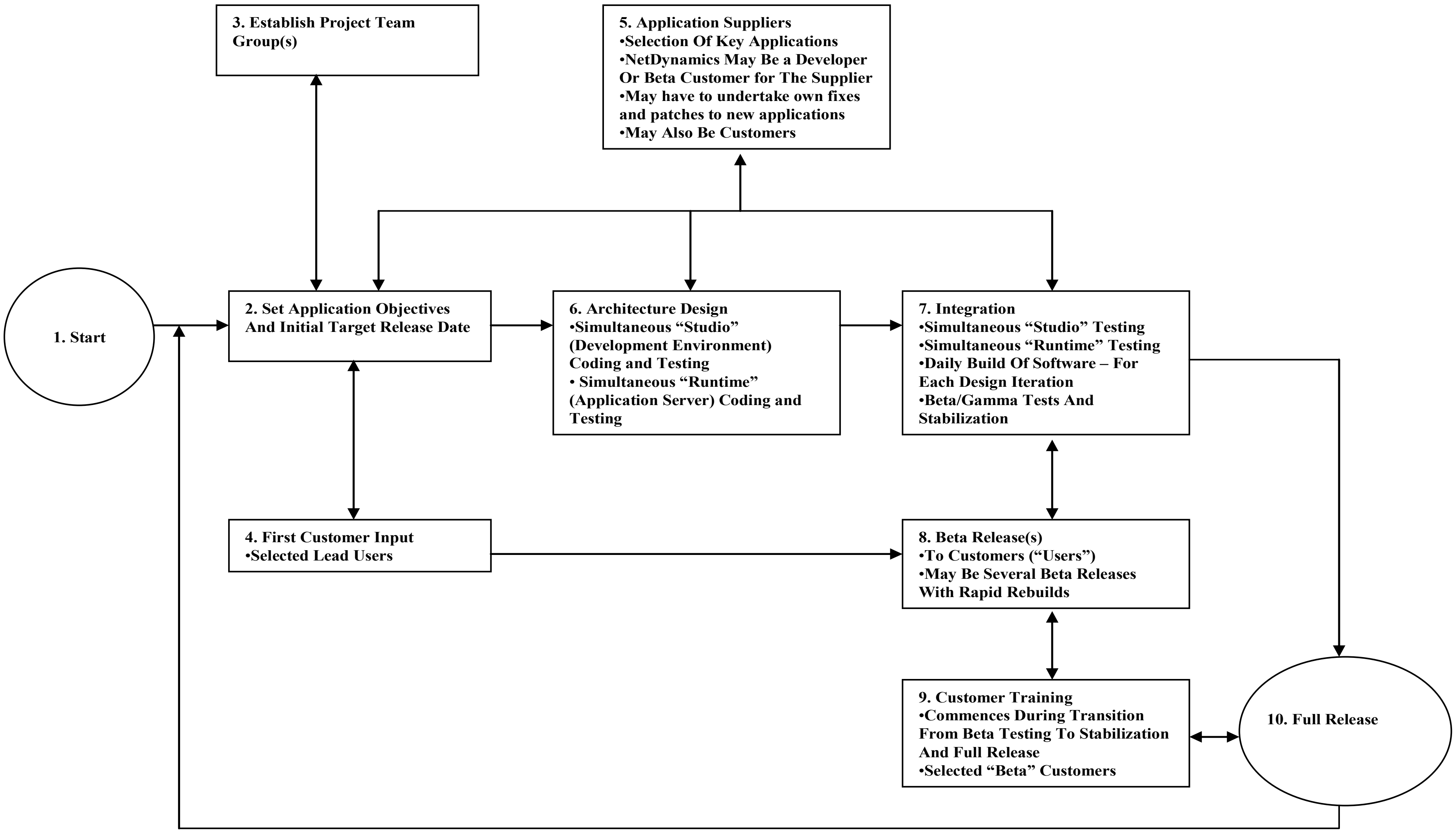


Figure 6.2 Summary DSA Model For NetDynamics Application Development
Created from data in Iansiti & MacCormack (1996)

However, release dates are important for various reasons including:

- focusing development teams to produce applications at specific feature and operational levels,
- for marketing communications to actively promote new applications, and
- for investors to indicate effective business development.

Therefore, at least in the current conceptualisation of application development, the full-release date (box 10) is a marker for a set of vital marketing and technical support events and activities.

The applications development process described here for NetDynamics 2.0 is based on a timeframe of about six months. Subsequent versions of NetDynamics ranged from about six months for Version 3.0, to about 14 months for versions 4.0 and 6.0.

The NetDynamics development processes (at least for the early versions) paralleled new and revolutionary developments in web browser applications.

NetDynamics, Sun Microsystems, Netscape Communications and other developers of Web browser-related applications were experimenting with each other's applications to produce a fully featured multimedia user interface environment.

At that very early stage of World-Wide Web user interface development, opportunities to shape the look and feel, access and connectivity were great. However, the window of opportunity was very small, and new applications had to be delivered rapidly in the face of impending competition from larger and more established application developers.

Microsoft's strategic shift toward the development of applications for the Internet in 1995-96 and the "Browser Wars" of 1995-1998 (see Kwak & Yoffie 1998; Cusumano & Yoffie 2000) underscored a short window opportunity for new web-browser application developers, from mid-1994 through to late 1996.

NetDynamics was the first application developer to successfully deliver the first server-based Java application (outside Sun Microsystems) to work with the World-Wide Web.

6.7.1 Updated DSA Model

Interviews with the then NetDynamics CEO (Rinat 2002a & b) and one of the senior developers of NetDynamics 2.0 (Ebbs 2002a & b) confirmed most of the process details in the original DSA model.

However, revisions were required to reflect additional decision-makers and changes in roles and links for customers and application suppliers. The revised DSA model is presented in Figure 6.3

There was some variation regarding the type of application development method (or lifecycles) that was addressed during the interview stage.

Initial reading of the NetDynamics case study seemed to suggest either a parallel, or an “Extreme Programming” approach to application development. Rinat (2002b) suggested that the approach was distinct to NetDynamics, combining a waterfall approach with early introduction of programming and prototyping.

Prior to the first integration, coding of the Studio and Runtime elements of the application was undertaken in parallel. After the first integration a rapid waterfall approach was used, but with the application rebuilt several times in a compressed timeframe.

Additional groups were identified as providing inputs for NetDynamics Application Objectives (box 2). The actual application development and release is undertaken and managed by the Project Team (box 3), and they provide most technical input to setting application objectives and target release dates.

The Project Team also provides critical input to selection of essential third-party applications based on its application knowledge and programming capabilities and the potential technical benefits a third-party application could deliver to the NetDynamics application.

A significant insight that emerged from the interviews was that for the development of NetDynamics 2.0, relationships with key application suppliers (box 6) such as Sun Microsystems (for Java technology) and Netscape Communications (for Netscape Navigator) were not formal or even close.

According to Rinat (2002a & b), interaction between Sun Microsystems, Netscape and NetDynamics was low, other than receiving beta copies of applications and advising of bugs and fixes. However, for later versions of NetDynamics applications (Versions 3, 4 and 5), there were more formalised applications licensing arrangements and ISV programs.

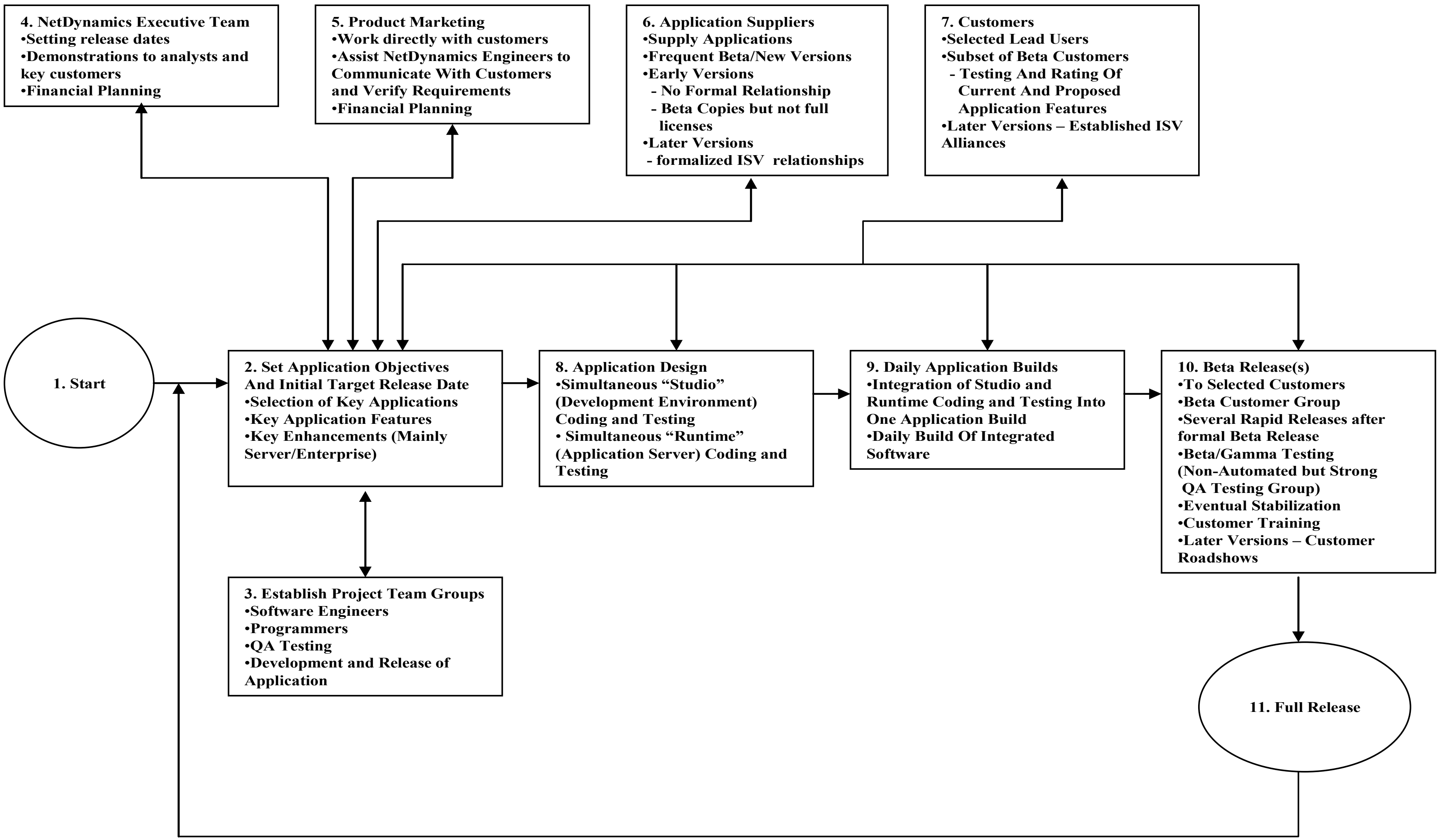


Figure 6.3 Updated Summary DSA Model For NetDynamics Application Development
Created from data in Iansiti & MacCormack (1996); Rinat (2002a & b); Ebbs (2002a & b)

The NetDynamics Executive Team (box 4) provided input on setting release dates, plus other key dates for the demonstration of prototypes or beta releases of the application to financial and technical analysts. In 1995-96, there was a Marketing Director and two or three product marketing staff at NetDynamics (box 5).

The product marketers worked directly with customers to facilitate communication between NetDynamics engineers and customers, and to verify customer requirements back to the development process. They were also involved in financial planning and management.

NetDynamics' "customers" were a mix of large organisations using the application on servers to link databases to WebPages in their websites, and ISVs.

Selected "Lead-User" customers were approached to provide input in the early stages of the development process. A further subset of "beta Customers" was selected to work closely on the design, development and testing of the application. This group of customers received frequent beta versions and identified bugs and fixes. They also provided detailed inputs on rating current and proposed application features.

In later versions of NetDynamics' applications (particularly in Versions 4 and 5), enterprise enhancements including APIs and connectors to ERP applications were incorporated, and enterprise customers were critical for testing and stabilising the application.

Application design, integration and daily application builds (boxes 8 and 9) retain most of the content from the original DSA model. A formal beta release date is defined (box 10), but several iterations of the beta release occur as customer feedback provides communications on identified bugs and fixes and often coded fixes.

Some later beta versions may be described as "Gamma" versions – but versions at this stage were subjected to intense QA testing inside NetDynamics. For earlier versions of NetDynamics, testing was non-automated but still rigorous. Eventually, the application is stabilised and readied for full release (box 11).

During the beta stage, formal customer training was conducted on the new application. For later versions of NetDynamics, a roadshow was undertaken to demonstrate the application to technical and business audiences.

The overall application development process was described as a “waterfall plus quick integration” software process (Rinat 2002b). As there was only one application involved in the development process, there was no parallel or spiral development processes related to alternative applications, but there was parallel processing of some stages in the software process (design, development, and daily integration).

Overall, the original DSA model was almost completely validated through interviews with Rinat (2002a & b) and Ebbs (2002a & b). However, insights obtained from the interviews on additional decision-making groups, and the roles and relationships of customers and applications suppliers, enabled the creation of an enhanced DSA. The enhanced DSA model addressed issues in 1995-96 but some additional insights were incorporated for subsequent versions of NetDynamics up to 1999.

6.8 Events Chronology Map

Iansiti & MacCormack (1996) described the development and release of NetDynamics 2.0 from early 1995 through to July 1996. An events chronology map covering this timeframe was created (see Figure 6.4). Some additional symbols were employed for the map including the flowchart symbol for a delay or problem and a six-pronged box for a solution to the delay or problem.

In the original events chronology map, a need was identified to make available on Microsoft Windows NT development tools for WebPages that link to companies’ internal databases. A start-up company, Spider Technologies, released such development tools as Spider 1.0 on UNIX in summer 1995 (box 2).

In late 1995, Spider Technologies was renamed NetDynamics and CEO Zack Rinat decided to develop the application (also renamed to NetDynamics 1.0) to run on both UNIX and Windows NT operating systems (box 3).

In December 1995 – January 1996, seven engineers were recruited to form a development team (box 4). Initially the team was given about two months to develop and release the new application, with a target launch date of February 1996 (box 5).

Nevertheless, the team produced a basic prototype of a set of mock-up screens during December 1995 (box 6). The prototype was shown to the most advanced users of NetDynamics 1.0, who were asked to evaluate the current prototype and assess potential options and futures that could be added to the product.

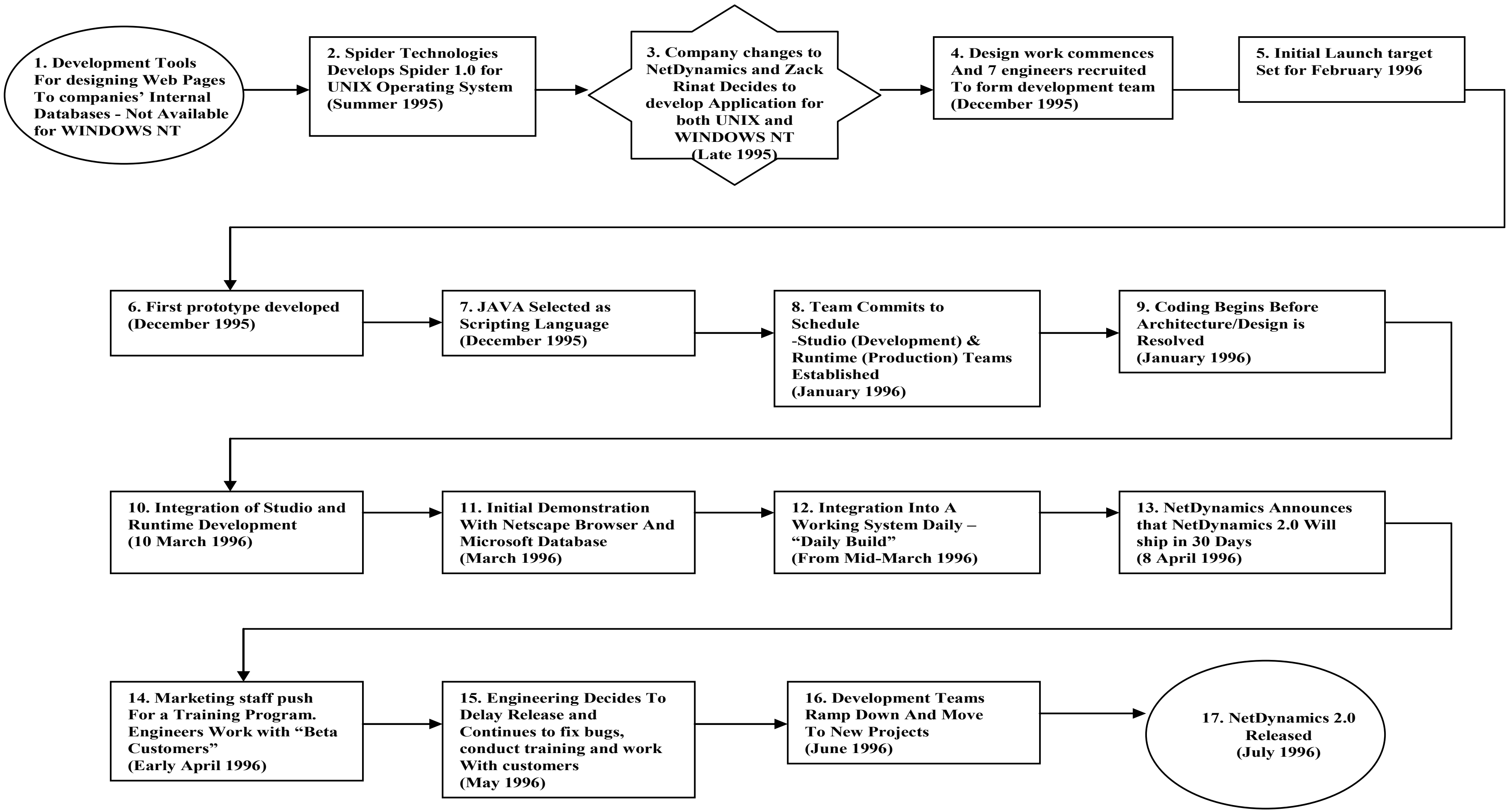


Figure 6.4 Events Chronology Map – NetDynamics
Created from data in Iansiti & MacCormack (1996)

A critical decision-point for the project was the selection of a Scripting Language to ensure effective linking between WebPages and database access languages (for example, SQL Server) or direct access to databases (box 7). There were five main options, but Java technology from Sun Microsystems was chosen because of its potential development capabilities.

NetDynamics' selection of Java was similar to Zaplet, except that by the time that Zaplet made its selection in 1998, Java technology was a comprehensive suite of applications with a very strong SDK and developer tools.

In late 1995, NetDynamics evaluated an early version of Java, spending about a month learning the language and evaluating its potential for development into a full technology platform.

After selecting Java for the project, the project team committed to the February 1996 launch schedule and set up "Studio" (Development) and "Runtime" (Production) teams (box 8).

The Studio team managed the application development environment (including the WebPages and Website sections of the application), while the Runtime team managed the application server, and the program handling data flowing between the server and the database.

At the same time in January 1996, the Studio project team started coding the application environment – before final decisions on architecture and design were resolved (box 7).

The decision to start coding the application environment early was controversial, as there was considerable complexity in ensuring that the server application would effectively either link to a standard database access application or to a database application directly. This was, after all, the very first ISV server-based Java application to be developed to work with the World-Wide Web.

Rinat believed that the ensuing months would bring sufficient additional information (particularly with regard to the development of Java and latest developments with Netscape Navigator) to sort out outstanding design and architectural issues.

The launch of Netscape Navigator 2.0 in March 1996 and advanced beta versions of Netscape Navigator 3.0 (the full version was launched in August 1996) seem to have provided the answers that NetDynamics were looking for. However, waiting for later beta versions of these applications may have contributed in part to pushing back NetDynamics' own application release date by three months to June-July 1996.

Rinat had scheduled a product demonstration in mid-March with analysts in New York, so, regardless of the outstanding architectural and design issues, there was additional pressure to produce something for that meeting.

After one month of development by the Studio team and around three weeks of coding by the Runtime team, the components were integrated into a full application on 10 March (box 10). Almost immediately, the integrated application was demonstrated to the analysts, with links between a Netscape Browser and a Microsoft Database (box 11).

From then on, the Studio and Runtime components were integrated daily into a working application (box 12). These "daily builds" were distributed to engineers and selected customers who were testing the application.

A system for checking new code and managing bugs and fixes prior to integration into the "daily build" was adopted. If the "daily build" process was followed through to the application stabilisation stage in about June 1996, then the application could have been rebuilt over 90 times!

On 8 April, 1996, NetDynamics announced that NetDynamics 2.0 would ship in 30 days, that is, around mid-May 1996 (box 13). There were still many tasks beyond the application coding before the application could be released.

In particular the marketing staff pushed for customer training to commence in early April (box 14). Engineers from the project team would have to conduct the training directly with 20 selected beta customers, as there was insufficient documentation for either other staff to conduct training or for the customers to train themselves.

The project team mixed customer training and application development as the selected customers pushed the limits of the applications. Both the project team and the selected customers unearthed several flaws and bugs that NetDynamics had not found and might not during the development cycle.

However, this phase of combined work stretched for about two months, and about three to four weeks into this timeframe (late April) the project team decided to delay the application release (box 15).

The application was stabilised in June 1996 and as the project ramped down, staff began to focus on development projects (box 16). It is likely that at this stage, teams were already being formed to commence development on NetDynamics 3.0.

NetDynamics 2.0 was formally released in June-July 1996 – about six months after the project team began developing the application.

NetDynamics now possessed an application that would be critical for enterprises to use the World-Wide Web for internal information-sharing, and for the new emerging world of e-commerce from both a B2C and B2B perspective. Nevertheless, the pressure would continue to develop the next version of the application as quickly as NetDynamics 2.0 with the incorporation of new features for it to remain a competitive application.

6.8.1 Updated Events Chronology Map

The original events chronology map was significantly revised after the interview stage to incorporate changes in details, dates and development activities. The updated map is presented in Figure 6.5.

The starting-point of the map was modified slightly to highlight the problem of a general need for designing web pages capable of linking to a company's internal databases.

There are two distinct development requirements to address this problem. One is to develop web pages that can handle database related activities, for example, to accept data for input to a database, to initiate database enquiries and to display results to enquiries.

The other requirement is to develop applications to link the server with the WebPages and the systems with the actual databases, either through direct access or through database enquiry languages and tools.

The development of WebPages to handle database related activities commenced in January 1996. The technology was given commercial life through the incorporation of Spider Technologies as a start-up company in May 1995 (box 2).

Spider 1.0 was released in July 1995 on Iris, the UNIX Operating System for Silicon Graphics systems, and was quickly ported and made available for other UNIX environments including Solaris (Sun) and HP-UX (Hewlett-Packard). Spider Technologies was renamed NetDynamics and the new application was renamed as NetDynamics 1.0 in autumn 1995 (box 4).

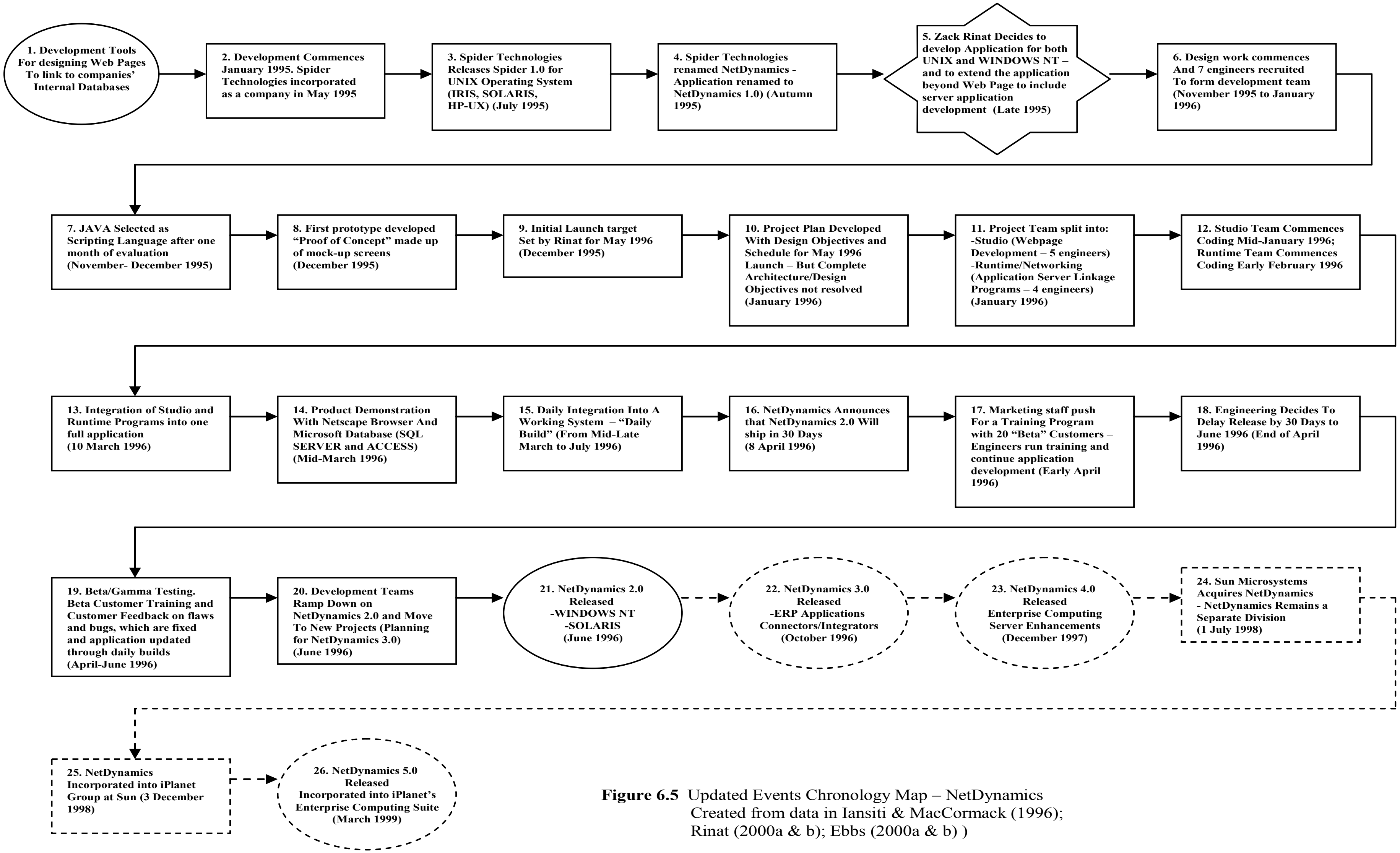


Figure 6.5 Updated Events Chronology Map – NetDynamics
Created from data in Iansiti & MacCormack (1996); Rinat (2000a & b); Ebbs (2000a & b))

In late 1995, the CEO of NetDynamics, Zack Rinat decided to develop a new version of NetDynamics for both UNIX and Windows NT. Rinat also decided to move application development from WebPages to application deployment, that is, to developing server applications, thus moving the application toward the first server-based Java application for the Web.

Design work on NetDynamics 2.0 commenced in November 1995 with seven software engineers recruited by early January 1996 (box 6). The order of events from December 1995 through to February 1996 was revised to amplify some development processes and changed dates (boxes 6, 7, 8, 9, 10, 11, and 12).

Selection of a scripting language and the development of the first mock-up prototypes screens were undertaken at the same time that the engineers were recruited and before the full project team was established (boxes 7 and 8).

According to Rinat (2002a), the initial launch date for NetDynamics 2.0 for May 1996 was set in December 1995 (box 9). Iansiti & MacCormack (1996) indicated that in December 1995, the release date was set for February 1996. However, Iansiti & MacCormack may have identified the requested date for the prototype that was required for demonstration in March 1996 (see box 14).

In January 1996, the project team created a project plan based on meeting the May 1996 launch date, but, as in the original events chronology map, several design and architecture issues were yet to be resolved (box 10).

The revised events chronology map contains additional explanation on the "Studio" and "Runtime" teams (boxes 11 and 12). The initial events chronology map assumed that the "Studio" and "Runtime" groups referred to separate development and production application environments, whereas these groups actually worked on different components of the application, which were integrated into the final application.

The "Studio" group focused on WebPages development and the "Runtime" group worked on the server application and networking applications required working with the application server.

The development timelines for the groups were different, with the "Studio" group commencing its coding in mid-January 1996, while the "Runtime" group commenced in early February 1996 (box 12).

The first application integration occurred on 10 March 1996 - around 8 weeks after "Studio" coding commenced and four to five weeks after the "Runtime" coding commenced (box 13).

The result was a prototype application for demonstration to analysts in New York using a Netscape Browser and Microsoft SQL server with Access database. There was no networking but the components demonstrated were sufficient to show effective end-to-end linkage of WebPages with a database (box 14).

Sometime in mid-late March and most likely after the product demonstration, a "daily build" process emerged where application coding and development was integrated into a working on a daily basis (box 15).

The 30-day announcement date was unchanged in the revised events chronology map (box 16). The project team's involvement with the beta customers was revised – training and development with them took about two months (boxes 17 and 19). About half-way through this period, the engineers decided to delay the release date by 30 days (box 18).

Comments on ramping down the project were revised to indicate confirmation that most of the project team moved on to the development of the next version of the NetDynamics application (box 20).

The actual launch date of NetDynamics 2.0 for Windows NT and UNIX (Sun SOLARIS) was updated to June 1996 (box 21) – a development cycle of around six months.

The remaining circles and boxes (22 to 26) are beyond the scope of Iansiti & MacCormack's (1996) case study. Nevertheless, they are included as indication of the subsequent development cycles NetDynamics V3.0, V4.0 and V5.0, that is, versions up to the time when the application was directly incorporated into Sun's iPlanet Enterprise Computing Suite in March 1999.

6.9 Cognitive Maps

Cognitive maps created for the NetDynamics case study were designed to complement DSA models and events chronology maps through additional exploration of selected decision-making issues and contexts.

All cognitive maps were created using the following procedures:

1. Selected extracts from Iansiti & MacCormack's (1996) case study were extracted to text files
2. One text file version of each extract was processed through the TACT package to produce a complete word-frequency list
3. Word frequencies were reduced through one or two iterations to ranked lists of words with frequencies ranging from over two to three words
4. A second version of the text file was input to NVivo
5. Ranked words were input to NVivo and coded as nodes in association with the text file
6. The overall extract was deconstructed into selected phases and sentences and coded directly as nodes through the InVivo tool within NVivo
7. The nodes associated with each extract were input to the Model Explorer and mapped using links incorporating codes of association plus addition node "items"
8. Several iterations of dynamic mapping were undertaken with the final cognitive map exported from NVivo to PowerPoint
9. Each PowerPoint version of the cognitive map was reformatted into a more readable and project-consistent format
10. The revised PowerPoint cognitive maps were presented to interviewees and validated or modified
11. The updated PowerPoint cognitive maps incorporated changes after the interview process.

Three sections were extracted from Iansiti & MacCormack (1996) and subjected to further text analysis and cognitive mapping to enhance insights derived from the DSA model and events chronology map.

The three cognitive maps created for the NetDynamics case study focused on the following issues:

6.9.1 First Prototype of NetDynamics 2.0 (1995)

6.9.2 Selection of Java (November-December 1995)

6.9.3 From Integration To Release (March-June 1996)

The first and third cognitive maps address the earliest and latest snapshots in time, highlighting the different issues to be addressed at the early stages of application development when compared to the testing and product release stages.

The second cognitive map is chronologically between the two other maps (in November-December 1995) but provides further insights into the evaluation and selection of Java as the main language/application technology for the new application.

6.9.1 Cognitive Map 1 – First Prototype of NetDynamics 2.0 (1995)

Development details for the first cognitive map include:

- Extract section, size and exhibit:
    - Case study introductory paragraph, plus "NetDynamics" Section (introductory paragraph), plus "Net Dynamics 2.0" subsection, Iansiti & MacCormack (1996, p. 1, and pp. 6-7); 2 paragraphs plus ¾ page; Exhibit 6.8
- TACT word frequency lists and comments on key words:
    - First pass: 34 words with frequency > 2
    - Second & final pass: 22 words with frequency > 3
    - Final ranked word list: Table 6.1
    - The highest ranking words reflected the theme of *NetDynamics*' *develop*ment of *products* for the *Internet* (italicised words were highly ranked words in the final wordlist).
- NVivo ranked word, sentence and phrase node count = 102
- Association of concepts coding used: +, -. +/-, =
- NVivo PowerPoint cognitive map: Figure 6.6
- Reformatted PowerPoint cognitive map: Figure 6.7

The cognitive map traces the genesis of a software application for designing web pages with links to databases through to a simple prototype. The map also highlights the need for the application to be developed for more than one popular operating system.

Exhibit 6.8 Text Extract: NetDynamics – First Prototype of NetDynamics 2.0 (1995)

Harvard Business School
9-697-052 Rev. April 21,1997
Living on Internet Time:
Product Development at NetDynamics
May you live in interesting times...
- Ancient Chinese curse

During the early 1990s, a number of small organisations began exploring a networking technology developed by a group of Swiss scientists in the late 1960s. That technology became the internet. One of these small organisations was Netscape. It and its fellow 'net travelers' adopted radically new development methods, and used the internet itself for distributing their products, saving the huge overheads associated with packaging and retailing. They used it to coordinate project tasks, run simulations and track progress. They used it for getting early and rapid feedback on the performance and usability of their products, avoiding the need for massive internal testing. They thus turned their small size to their advantage, churning out new product releases in incredibly short development cycles, often only a few months long.
The world had moved to internet time...

Living on Internet Time 697-052
NetDynamics provides development tools to help design web pages with links to a company's internal databases. The product is extremely easy to use (a simple web page with links can be designed in minutes) and extremely "open" (it supports internet and database software products from any vendor, for example, both Microsoft's "Explorer" and Netscape's "Navigator" browsers). The company, originally named Spider Technologies, launched its first product, Spider 1.0 for the UNDC operating system, in the summer of 1995. It soon became clear, however, that the rapid growth of Microsoft's "Windows NT" operating system would require a product that also ran on the NT platform. Late in 1995, Zack Rinat, NetDynamics CEO, met with his senior engineers and decided that rather than trying to port across the original design to the new operating system, it would be better to develop a completely new version of the product for both UNDC and NT. That project is outlined below (see Figure 3).

Net Dynamics 2.0
Design work for the new product started on December 1, 1995. To staff the project, seven engineers were recruited, each with a background in developing software for Windows-based systems. The new hires included Nanda Kishore, who became the Vice President of Engineering, Dan Lee, and Yarden Malka, who became the technical leads for the project. These engineers joined the two who had developed the original product, to form the initial development team.
While Kishore and Lee came from another software company in Silicon Valley, Malka moved all the way from Israel to join NetDynamics. Rinat had hired him after the recommendation of a friend and a single (expensive) telephone interview. When Malka arrived in California, two days later, Rinat was waiting for him at the airport. Their first stop was at the NetDynamics office, where Malka immediately began to contribute to the 2.0 project. Malka describes the experience:

Exhibit 6.8 Text Extract: NetDynamics – First Prototype of NetDynamics 2.0 (1995) (Cont'd)

76
Living on Internet Time
NetDynamics
It was incredibly exciting. Before joining Spider, I had worked for three other startups in Israel. None of them succeeded, because they did not manage to attract enough money. But Spider had funding! This one could actually work...

Living on Internet Time 697-052

The initial target was extremely aggressive; Rinat wanted to have a product ready by the end of February 1996!
During December, the team developed a simple first prototype comprising a mock-up of potential screens. The most advanced users among existing customers were brought in to evaluate the prototype before being given a list of potential options and features that might be added, and asked to rank their importance. This was done at an extremely detailed level to provide input to the design specification (for example, asking which specific database connections were important). The results from this exercise were tabulated on a spreadsheet and used to inform design tradeoffs.

Source: Iansiti & MacCormack (1996, p. 1, pp. 6-7).

Table 6.1 NetDynamics – First Prototype of NetDynamics 2.0 (1995) – Ranked Word List

Anchor Concepts	Frequency
product	12
develop	10
internet	8
was	6
their	6
new	6
Netdynamics	6
design	6
with	5
use	5
time	5
project	5
Malka	5
live	5
became	4
were	4
they	4
system	4
spider	4
Rinat	4
extremely	4
engineer	4

Source: data analysis performed using TACT (1997)

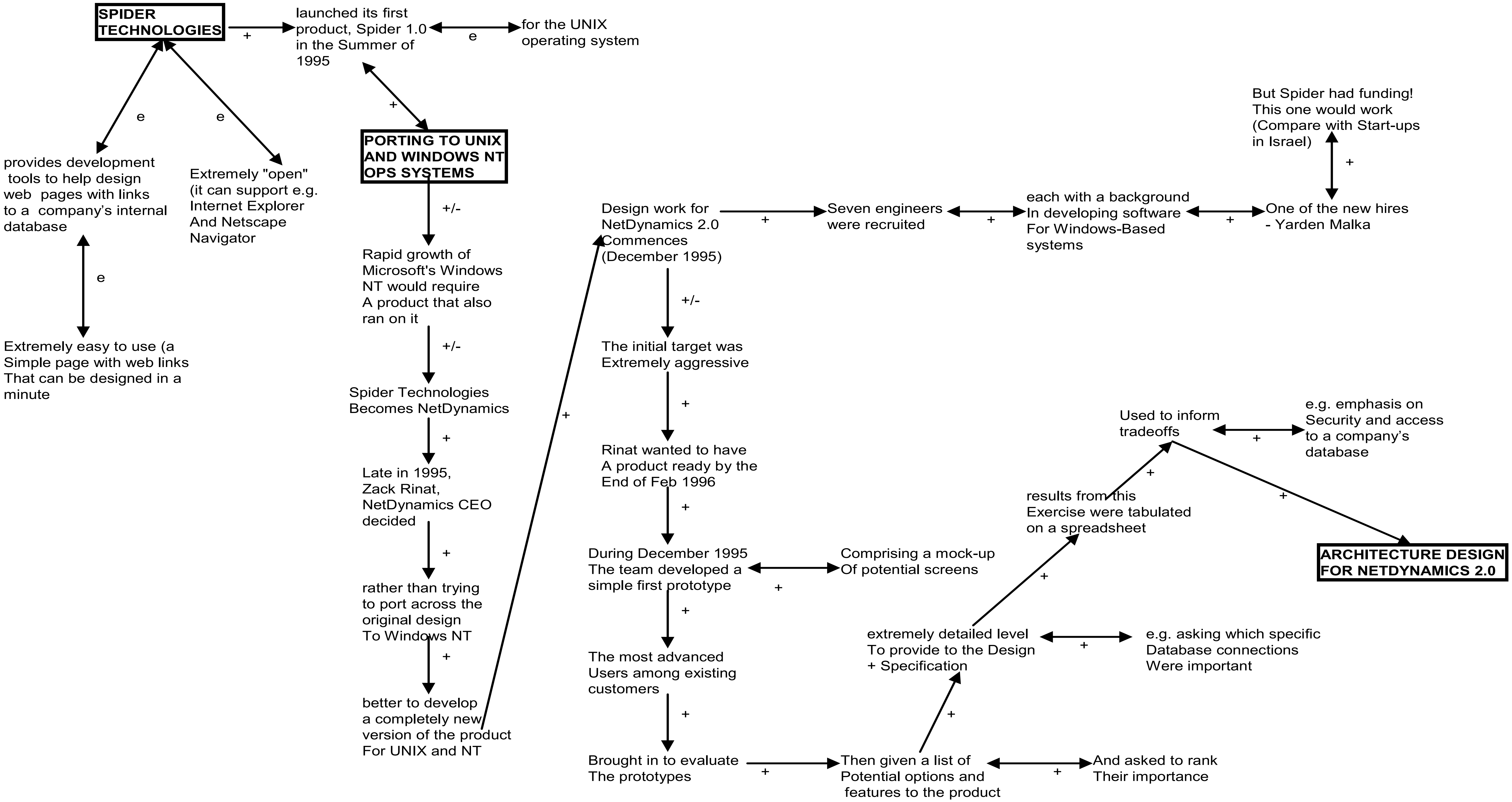


Figure 6.6 First Cognitive Map (NVivo Model) – NetDynamics – First Prototype of NetDynamics (1995) Created from data in extract (Exhibit 6.8).

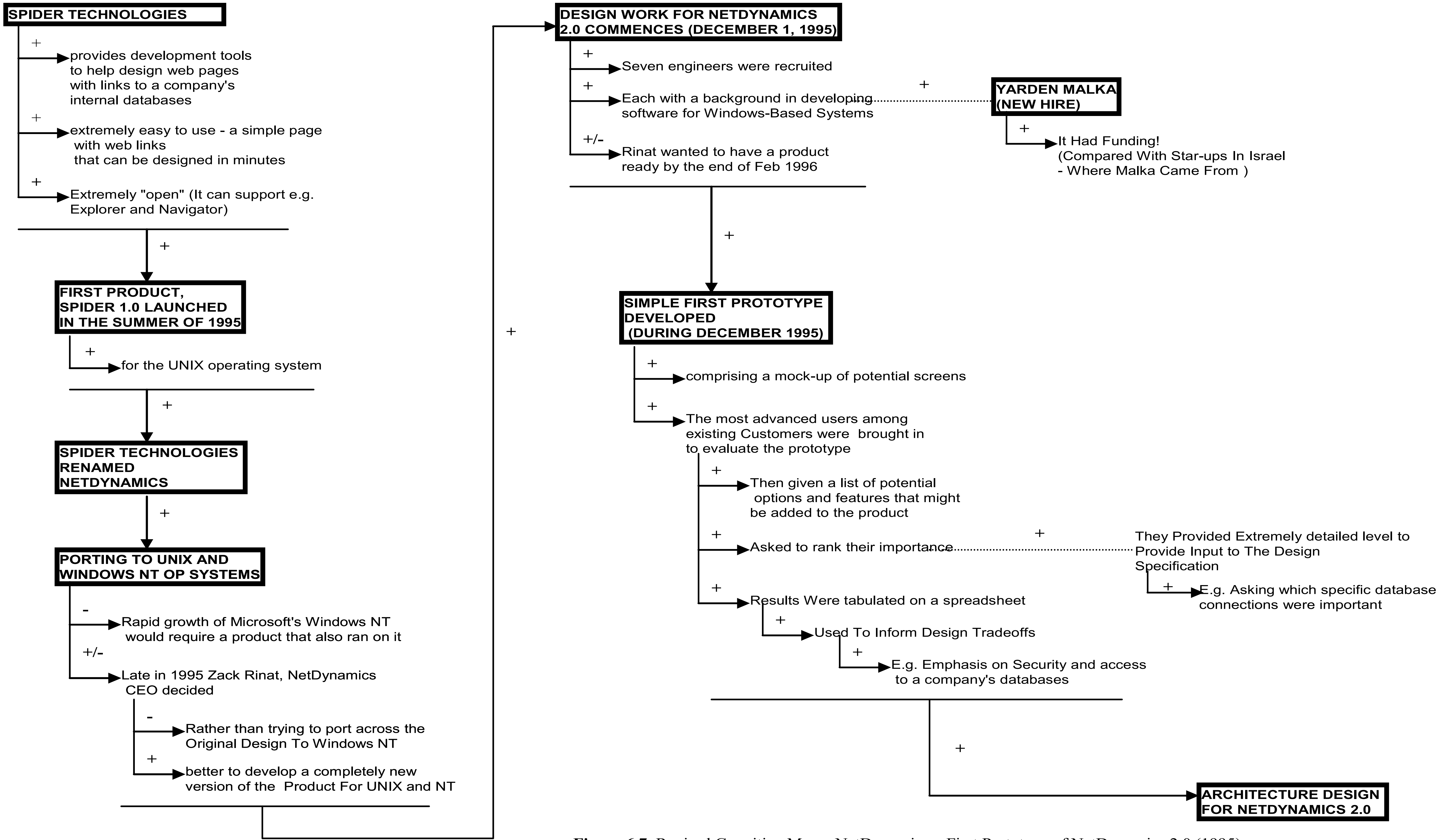


Figure 6.7 Revised Cognitive Map – NetDynamics – First Prototype of NetDynamics 2.0 (1995)
Translation of NVivo Map
Note: This revision was shown to participants in emic 2 stage for their (Rinat and Ebb's) interpretation.

The application would be a "killer application" for enabling e-commerce and business-to-business (B2B) activity over the World-Wide Web.

Spider Technologies developed such an application that was easy to use and could run with Netscape Navigator and Microsoft's Internet Explorer. Spider 1.0 was launched in the summer of 1995, enabling simple web page links within a server running the UNIX operating system.

Spider Technologies was renamed NetDynamics in 1995, but Zack Rinat, CEO of NetDynamics, recognised that Microsoft's Windows NT operating system was growing rapidly and an application would be required to run on it.

Rinat decided that, rather than porting the existing application across from UNIX to Windows NT, NetDynamics would develop a completely new application to run on both UNIX and Windows NT operating systems.

Development of a Windows NT-based application required new software engineers with experience in developing Windows-based software. Seven software engineers were recruited in late 1995, including Yarden Malka, who joined NetDynamics from Israel.

According to Malka, the new NetDynamics start-up company had funding and resources, whereas start-ups in Israel struggled to find funding for software development.

Design work for the new application, to be known as NetDynamics 2.0, commenced on 1 December 1995, before most of the engineers had been recruited. The new project team worked quickly toward developing a simple prototype in December 1996, which comprised a mock-up of potential screens.

The most advanced NetDynamics 1.0 customers were brought in to evaluate the prototype, plus a list of potential options and features that could be added to the application.

The customers provided detailed input to design specifications, for example, on which database connections were important to them. The rankings of the list of potential options and features were tabulated into a spreadsheet and assisted with various design tradeoffs such as company database security and access issues.

The results from the customer evaluation of the NetDynamics 2.0 prototype enabled the project team to create an architectural design for the application – but actual coding and design issues were not resolved at this stage of the development process.

6.9.1.1 Updated Cognitive Map 1 – First Prototype of NetDynamics 2.0 (1995)

The cognitive map was revised after the interview stage through direct editing in PowerPoint (see Figure 6.8). There were some revisions required in terms of dates and description. The major sections were numbered and a new section (box 4) was created referring to moving the NetDynamics application from web page development to include application server development.

The sections on Spider Technologies (boxes 1 and 2) were revised to indicate that Zack Rinat and others started developing the web-based development tools that would become the Spider (and then the NetDynamics) application in early 1996. The development work was incorporated into a start-up company in May 1995, under the name NetDynamics.

The first application, Spider 1.0, was launched in summer 1995, providing development tools to help design web pages with links to a company's internal databases, where the operating environment was UNIX-based.

Spider 1.0 could be used with Netscape Navigator and Microsoft's Internet Explorer web browsers. Spider 1.0 was specifically ported to UNIX versions on systems from Silicon Graphics (SGI – Iris), Hewlett-Packard (HP-UX) and Sun Microsystems (Sun Solaris).

Spider Technologies was renamed NetDynamics and with it Spider 1.0 to NetDynamics 1.0 in autumn 1995 (box 3 was revised to indicate change in application name).

Decisions related to deploying new application for both UNIX and Windows were updated and captured under a new section (see box 4 "Application Deployment for UNIX and Windows (Late 1995)". The new application would be designed around one common source code but deployed across both operating systems environments.

The common source code decision was critical, leading directly to consideration of a language or application technology that could be "operating system independent" – one of the main claimed benefits of the new Java technology for the Web at that time. Java technology could assist in both speeding up development of the application, and in the efficiency and consistency of the coding of such an application.

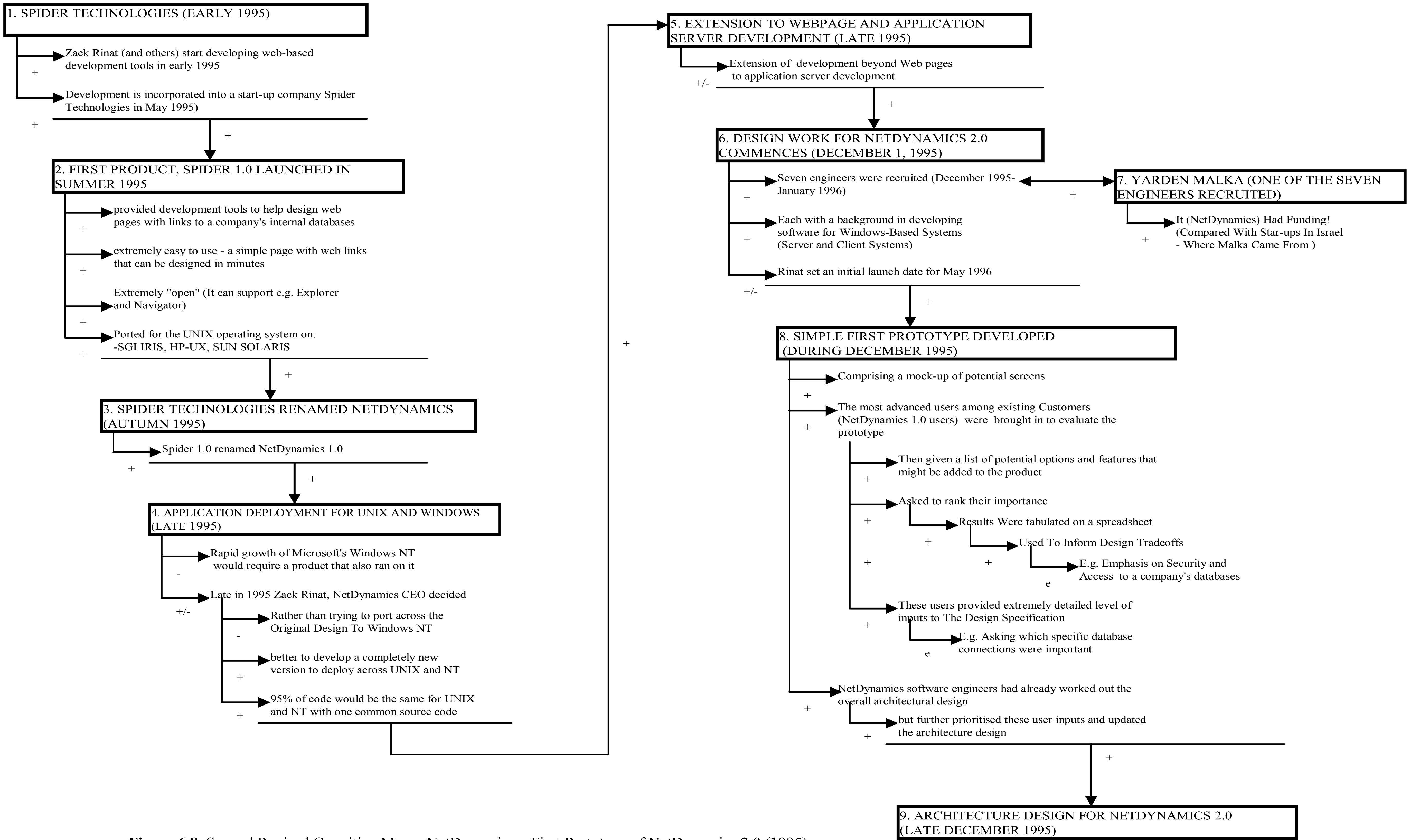


Figure 6.8 Second Revised Cognitive Map – NetDynamics – First Prototype of NetDynamics 2.0 (1995)
Note: This map represents etic 3 interpretation.

A new section on the extension of the application beyond webpage development to include application server development was added to the cognitive map (see box 5). Two outcomes emerged from this decision:

- development focus for NetDynamics changed toward the creation of the first server-based Java application
- development focus also changed from a computer-system-based concept to a purely software concept of "application servers".

During the early 1990s the term "application server" represented a specific computer system with some systems and application software on it. For example a Sun Microsystems midrange computer system with an Oracle database running on it plus an SQL Server program, could be regarded as an "application server".

However, during the late 1990s through to the present day, the term "application server" was extended to mean a database (or an application) running across several systems and communicating with several other systems. In other words, the concept "application server" can now be both computer system and operating system-independent.

With respect to the new NetDynamics 2.0 application, it was developed mainly under the computer-specific definition of the "application server", but subsequent versions became more computer-and-operating system-dependent.

By NetDynamics 4.0 in late 1997, the application was a full "Enterprise Server" and evolved into the iPlanet Enterprise Computing Suite which in turn was incorporated into Sun's Web Services solution, Sun One in 2001/2002.

Sun One is a Web-based operating system-independent infrastructure capable of deploying applications plus creating and distributing information "objects" across any infrastructure.

The section covering commencement of design work for NetDynamics 2.0 was revised with additional commentary (box 6). Recruitment of the seven software engineers took about two months with recruitment continuing after the first prototype had been developed.

The initial launch date was set by Rinat for May 1996. While less aggressive than the February 1996 date indicated by Iansiti & MacCormack (1996), it still was a very ambitious target as several architectural and design issues were still to be addressed.

Although Yarden Malka's comments (see box 7) were unchanged in the revised cognitive map, they suggest that Malka was confident about NetDynamics having access to the level of funding and resources required for developing the new application in the short timeframe.

Description of the first prototype was unchanged in the revised cognitive map, with the exception that, according to Rinat (2002b), the NetDynamics software engineers had already worked out the overall architectural design for NetDynamics 2.0. The design work included the selection of Java before the prototype was created during December 1995 (see box 8).

Nevertheless, the engineers accepted the inputs from the advanced NetDynamics 1.0 users (that is, advanced existing users), further prioritised and incorporated them into an updated architectural design in late December 1995 (boxes 8 and 9). These revisions completed the cognitive map.

6.9.2 Cognitive Map 2 – Selection of Java (November-December 1995)

Development details for the second cognitive map include:

- Extract section, size and exhibit:
    - "Net Dynamics 2.0" sub section in Iansiti & MacCormack (1996, p. 6); ¾ page; Exhibit 6.9
- TACT word frequency lists and comments on key words:
    - First & final pass: 16 words with frequency > 2
    - Final ranked word list: Table 6.2
    - The words "develop" and "Java" scored the highest frequencies with "engineer" and "language" scoring much lower frequencies but reinforcing the theme of evaluating Java as a language to be used in the development of NetDynamics 2.0.
- NVivo ranked word, sentence and phrase node count = 76
- Association of concepts coding used: +, -. +/-, e
- NVivo PowerPoint cognitive map: Figure 6.9
- Reformatted PowerPoint cognitive map: Figure 6.10

The final cognitive map provides a detailed description of the tasks undertaken and the thought processes for evaluating different languages for the new application. There is a greater level of description of the processes associated with the evaluation of Java;

mainly because it was new and very little was known about its current and potential capabilities.

Exhibit 6.9 Text Extract: NetDynamics – NetDynamics Selection of Java (November-December 1995)

The single most important technical decision taken during this period was the choice of scripting language. By the end of December, the company had reviewed five alternatives: a proprietary language (nicknamed "E"), Visual Basic (a popular, but rather elementary language), C++ (another popular language, but with implementation drawbacks due to its compilation time, which would lengthen customers' development cycles), Python, and Java. The choice quickly came down to either "E" or Java, mainly because these would result in the maximum speed and simplicity with which a customer could develop their application.

After much debate, the team finally chose Java.

What helped the team make such tradeoff decisions was the clear original vision for the product developed in conjunction with lead users: Emphasise ease-of-use in application, and adopt an "open" platform philosophy (if a standard was emerging, that should take precedence over developing proprietary technology). However, while Java was ultimately chosen, the choice generated significant controversy, especially given its relative immaturity at the time and the limited information about its capabilities. "We knew Java was going to be big," recalled Malka, "but it was still only available as a 1.0 beta version. This meant that the development tools which went along with it were either terribly buggy or nonexistent. Subsequently, we had to develop many of our own development tools."

To make the decision, NetDynamics engineers spent weeks experimenting with various options, trying to become as comfortable as possible with the benefits and risks of each language. Since they were experienced with other approaches, the engineers spent the bulk of their time familiarising themselves with Java. They began by writing simple routines, and gradually migrated to more complex programs, attempting to gauge the advantages its use would give to the end-user. They also investigated available programming tools, debating the relative advantages of each and attempting to forecast their likely evolution. With such detailed experimentation, the decision took over a month to resolve. While Java, as it turned out, was the right bet, the ground could have shifted mid-project. If it had, NetDynamics' development effort would have had to shift with it.

Working with other companies' beta software was a continual problem during the development cycle. For example, much of the work carried out with Java unearthed bugs that even Sun Microsystems had not known about. However, in order to demonstrate that the bug was in Java, not their own code, the engineers often had to develop small test programs to "expose" the bug. On one occasion, with the team using a fast beta version of Netscape's Navigator software, all the engineering systems at NetDynamics suddenly crashed. After several hours investigation, the engineers traced the fault to a "timebomb" installed in Navigator, to stop unauthorised use past a certain date.

Source: Iansiti & MacCormack (1996, p. 7).

Table 6.2 NetDynamics – Selection of Java (November-December 1995) – Ranked Word List

Anchor Concepts	Frequency
develop	10
java	9
was	9
engineer	5
language	5
decision	4
would	4
beta	3
bug	3
choice	3
netdynamics	3
programs	3
team	3
time	3
tools	3
use	3

Source: data analysis performed using TACT (1997)

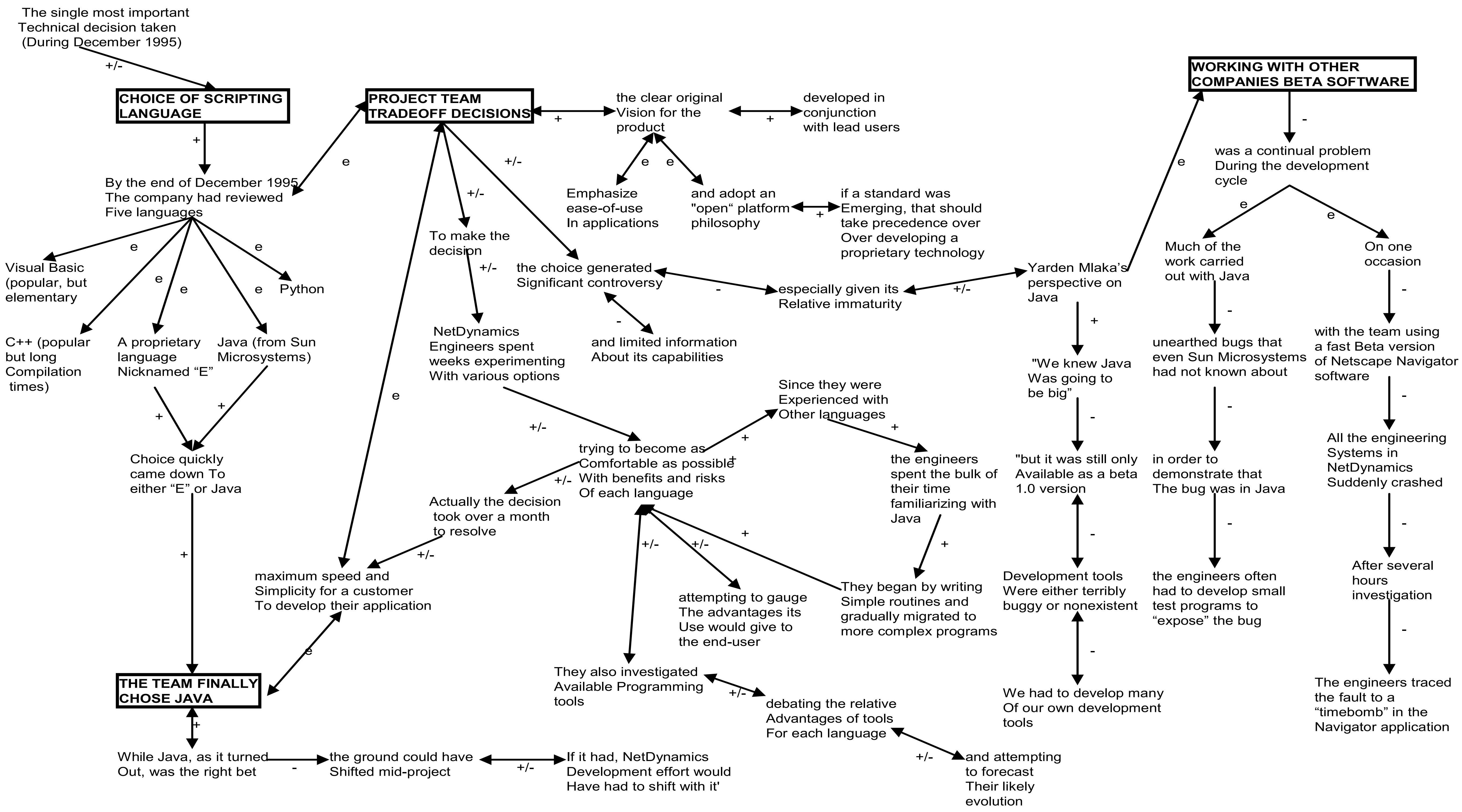


Figure 6.9 First Cognitive Map (NVivo Model) – NetDynamics – Selection of Java (November-December 1995)
Created from data in extract (Exhibit 6.9)

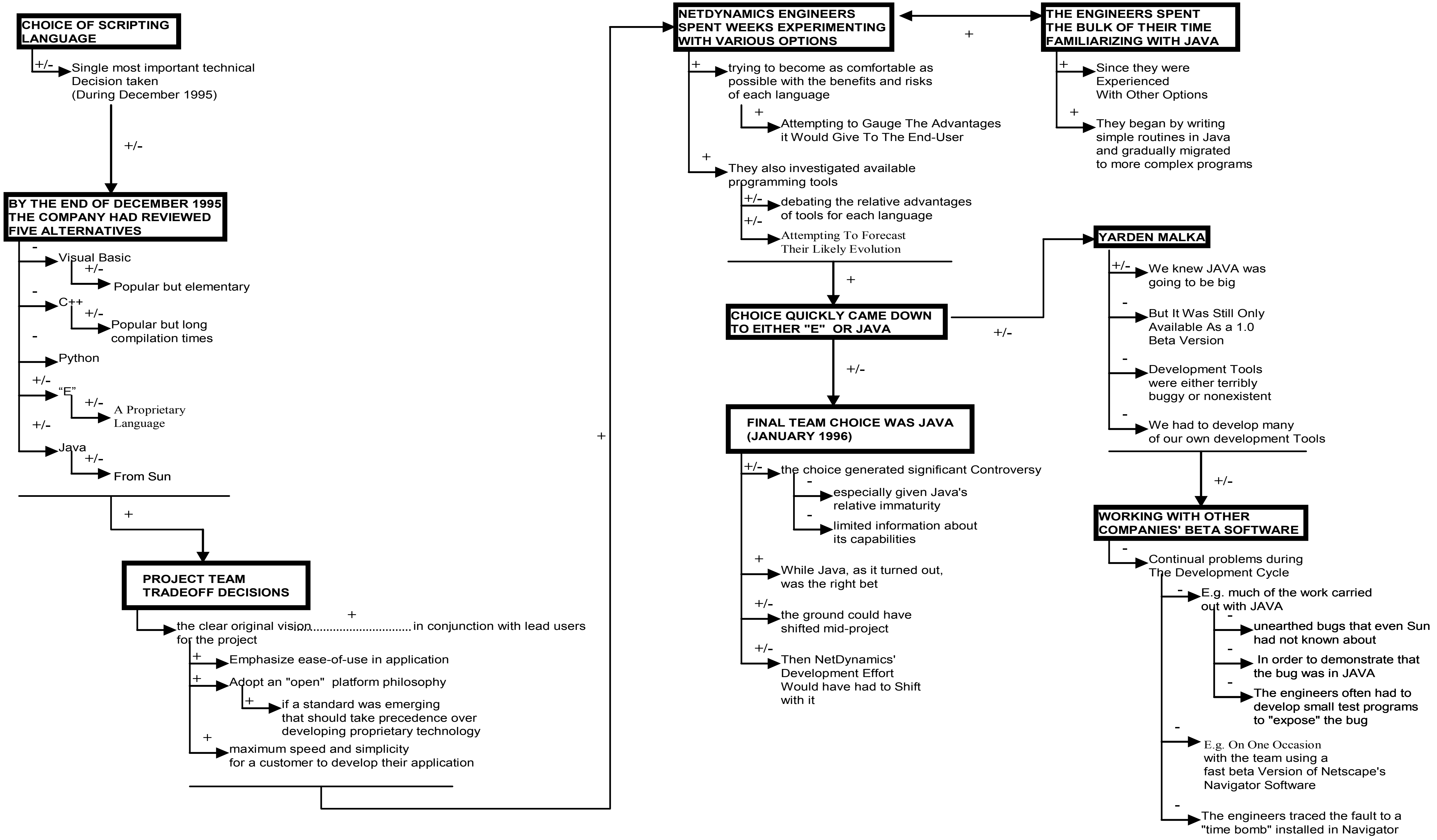


Figure 6.10 Revised Cognitive Map – NetDynamics – Selection of Java (November-December 1995)
Translation of NVivo Map
Note: This revision was shown to participants in emic 2 stage for their (Rinat and Ebb's) interpretation.

According to Iansiti & MacCormack (1996), the choice of scripting language for the new application was the most important decision taken in the project. NetDynamics' software engineers had evaluated five alternative languages by the end of December 1995.

Microsoft's Visual Basic was a popular language, particularly for Windows-based application development. However, for the version available in 1995, elementary graphics and very simple actions could be coded in Visual Basic – but not advanced multimedia actions or links to other applications such as databases.

C++ was popular for developing UNIX-based applications. Object-oriented applications could be developed using C++, providing functionality for the development, transfer and storage of more advanced multimedia information files.

However, C++ applications were still computer-system and operating-system-dependent, and advanced applications required long compilation times. Nevertheless, most software engineers developing applications for the Internet in the mid-1990s would have been experienced with C++.

Although not discussed in Iansiti & MacCormack (1996), *Python* is a portable, object-oriented programming language. Python was created in the early 1990s by Guido van Rossum at Stitching Mathematisch Centrum (CWI 2003) in the Netherlands as a successor of a language called ABC.

Guido remains Python's principal author, although the application now includes contributions from several other developers. In 1995, Guido continued his work on Python at the Corporation for National Research Initiatives (CNRI 2003) in Reston, Virginia, where he released several versions of the software.

Python can be used directly with C++ but it was portable across UNIX and Windows platform, thus making it a serious contender for the new NetDynamics application.

However, according to sources covering the history of Python in 1995, development was focused on UNIX and "Open Systems" development, with enhancements and tools for the Internet/World-Wide Web, emerging two or three years later around 1998 (PYTHON 2003).

The original *E* language was developed around 1993 by Electronic Communities as a set of programming tools with object-oriented and distributed applications development capabilities.

E appeared to have similar development capabilities to Java. At the time of the case study E could have been used for application server development. However, during 1996 the E language was developed to act as a set of extension tools to Java. From 1996 onwards Java has been required to run the E language.

Java was developed by Sun Microsystems in the early 1990s and offered an operating-system-independent development environment. Development for Web-based applications was new – Sun and Netscape agreed to develop Netscape Navigator incorporating features of Java technology in July 1996. At the time of the case study, this first version of Java to work with Netscape Navigator was still in a beta form.

The NetDynamics 2.0 project team wanted a language that would assist to achieve its original vision for the new application. The team was working with leading NetDynamics 1.0 users (see first cognitive map).

Both the engineers and users envisioned an application which was easy to use, was "open" in terms of operating systems platforms, and provided customers with maximum speed and simplicity to develop their applications. These requirements were quite ambitious given the relative primitive web-browser and programming capabilities in late 1996.

The NetDynamics engineers experimented with the languages selected for evaluation over several weeks in December 1995. The engineers tried to understand the benefits and risks of each language, not only from their own perspective, but from that of the NetDynamics application end-users.

Each language was evaluated for its available programming tools, not only on current tools but also on the forecast likely evolution of each of the languages and their tools.

The engineers were experienced in Visual Basic, C++, and probably to a lesser extent, with Python and "E" – so they concentrated on evaluating the new Java language through developing their programming skills in it.

Given that "E" and Java possessed the strongest existing and potential capabilities for application server development, there was an obvious reduction of choice down to these two languages.

Yarden Malka could see the huge potential for Java as it developed, but could also see substantial challenges in working with an early (1.0) beta version of the technology. Development tools were either buggy or nonexistent and NetDynamics engineers had to spend extra time developing many of their own development tools.

The final selection of Java in January 1996 was controversial because of the very early development level of Java and limited information from Sun Microsystems about its capabilities. Java was at that time a big gamble for the small start-up company – and it turned out to be a successful bet.

Nevertheless, if a different and more appropriate application development language had appeared during early 1996, then, according to Malka, the NetDynamics' project team would have shifted quickly – even mid-way through the project - toward using the new language.

Challenges working with Java and beta versions of Netscape's Navigator browser at that time highlighted a set of issues faced by software houses using beta versions from other software houses (or ISVs).

For example, as NetDynamics worked with the Java technology, they unearthed bugs that Sun did not know about. The NetDynamics engineers had to prove to Sun that the bugs were their problem by exposing the bugs through test programs developed by them.

In another example, NetDynamics' systems crashed while the project was using a beta version of Netscape Navigator, and the engineers eventually traced the fault to a "time bomb" installed in the Netscape Navigator beta version.

In fact NetDynamics (itself an ISV developing its own application which would be tested with its own users) was acting as a lead-user/test-bed for both Netscape Communicator and Sun Microsystems in 1995-1996. Cross-testing of applications was common as the World-Wide Web environment was still at a very early stage of development at that time.

6.9.2.1 Updated Cognitive Map 2 – Selection of Java (November-December 1995)

The updated cognitive map for NetDynamics Selection of Java is presented in Figure 6.11. Although most section headings were validated as unchanged through the interview stage, there were substantial additional insights and comments for every section in the original map.

There was some variation in the actual timing of the decision to go with Java for the development of NetDynamics 2.0. Rinat (2002a) indicated that the decision to go with Java was made in November 1995, while Ebbs (2002a) claimed that the decision was made in late December 1995 – early January 1996.

It is possible that the announcement made by Netscape Communications and Sun in July 1995, to incorporate Java features into the next versions of Netscape Navigator (Versions 2 and 3), was a catalyst for some developers to quickly incorporate Java into their own development projects.

NetDynamics may have noted the Netscape/Sun decision and was to some degree predisposed toward favourably viewing Java for the project. From that perspective, when planning for the project commenced around November 1995, the decision to run with Java may have already been made based on a bet that the new language for emerging Web browsers would quickly become a preferred scripting language for that environment.

Software engineers also viewed testing Java as the next big language for software development and were eager both to seek out applications to be developed using Java, and to learn Java programming skills. As the project team assembled and new software engineers were recruited there was plenty of incentive for engineers to choose to learn and work on Java – reinforcing a predisposition toward selecting Java.

The choice of scripting language (box 1) was critical to the project as NetDynamics 1.0 was written mainly in C++ which was adequate for some object-oriented activities and for UNIX-based systems, but the new application would have to run both in the Web and across systems. A new and more powerful language or software technology was required for that type of development.

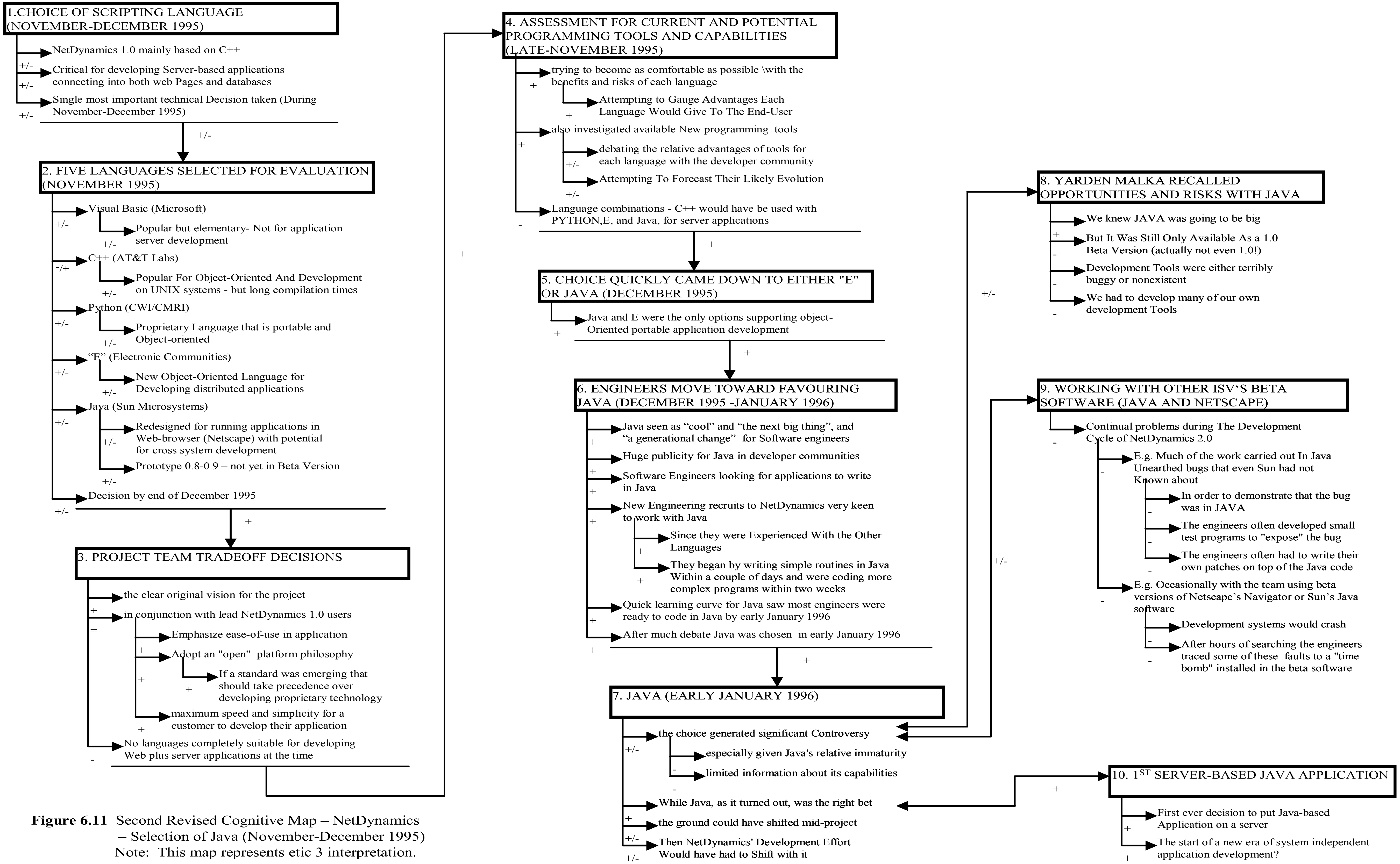


Figure 6.11 Second Revised Cognitive Map – NetDynamics – Selection of Java (November-December 1995)
Note: This map represents etic 3 interpretation.

Officially five languages were selected for evaluation (box 2). The updated cognitive map provides additional details about each language including features and capabilities.

Capabilities ranged from simple graphics and processes under Microsoft Visual Basic through to Web-development capability plus potential cross-system development with Sun Microsystems' Java.

The updated descriptions for each language give an indication as to the decision to quickly reduce the shortlist down to "E" and Java was simple (the actual short-list to these two languages is noted in box 5).

The trade-off decisions for the project team (box 3) provide additional insight in that the vision for developing NetDynamics 2.0 was developed in conjunction with leading NetDynamics 1.0 users.

A further key point was added, highlighting that none of the alternatives was completely suitable (at least in its then current form or version) for developing the type of application NetDynamics had envisioned.

The actual evaluation section (box 4) was renamed but remained unchanged except for an additional insight into language combinations. This point recognised that as no one language could completely meet the project vision then combinations would be required.

In fact it was common from about 1995 onward for applications to be coded with C++ and extensions in Java or Python or "E". As Java developed more of the applications developed C++ would be migrated to Java.

From about 1997 onwards, "E" was redesigned as extensions for Java. Nevertheless, C++, Corba and Java language code were used in the last versions of NetDynamics in 1999-2001.

As noted in the evaluation process (box 2), the choice was quickly reduced to "E" and Java (box 5). This section was revised with an additional comment to reflect the reason for the choice, that is, they were only the options supporting object-oriented portable application development.

The section on the engineers familiarising themselves with Java in the original map was recast as there were substantial insights into how and why the software engineers moved toward favouring Java (box 6).

Java was, at the time, viewed by many software engineers as being a “cool” language, or “the next big thing”, with younger engineers seeing it as representing a generational change.

Sun had generated high levels of publicity with its announcement to incorporate features of Java into Netscape Navigator Java in developer communities – and particularly those working on web-based applications. Software developers in these communities were eager to develop applications using Java – even if the language was still in pre-beta versions.

Although the new engineers recruited by NetDynamics in December 1995 – January 1996 were brought on board mainly for their Windows-based capabilities, they were all keen to work with Java. As they were experienced in most of the other languages to be evaluated, the new engineers took the opportunity to focus on and learn more about Java.

That they were able to write simple routines in a couple of days and more complex programs within two weeks, was a major factor in the project team’s moving toward choosing Java. Most engineers were “Java-Ready” by early January 1996 – in time for commencement of coding the new application.

According to Ebbs (2002b) Java was chosen by the engineers in early 1996. It was a fairly chaotic time as new engineers were coming on board and the first prototypes were developed, while the evaluation process was completed. Although the choice seemed obvious because of the requirement for server and webpage development, it was risky because of the immaturity of Java at that time.

The revised cognitive map maintains the insights in the final selection of Java (box 7) and Yarden Malka’s perspectives on Java (box 8). Actually Java was far less advanced in development than indicated in Iansiti & MacCormack (1996) – actually being in pre-beta stage at the time (as Rinat (2002b) put it – at 0.9 or 0.9 and not even beta 1.0!).

There was really a “triple beta version” situation in which the Java technology for the Web was pre-beta, the upcoming Netscape Navigator 2.0 browser was in beta version, and of course the NetDynamics 2.0 application was in early prototype form.

This particular ISV-to-ISV beta development environment (box 9) saw a continual reporting of bugs and fixes – and even development of development tools by ISVs – other than just being a supplying ISV.

Time-bombs in beta versions of applications were normal, as developers were encouraged to upgrade either to later beta versions or to full release versions – but they were disruptive to development when they were not detected.

A new section (box 10) was added to reflect an important new insight from Rinat (2002b) that NetDynamics 2.0 was the first ever server-based Java application (other than specific system tools from Sun Microsystems).

Java was the right bet for NetDynamics – but it was also the right bet for Sun and Netscape in 1995-96 as Java heralded a new era multimedia Web content and laid foundations for system independent application development, and new enterprise based e-commerce applications.

6.9.3 Cognitive Map 3 – From Integration to Release (March-June 1996)

Development details for the third cognitive map include:

- Extract section, size and exhibit:
  - o "Net Dynamics 2.0" sub-section in Iansiti & MacCormack (1996, pp. 7-8); 1 ½ page; Exhibit 6.10
- TACT word frequency lists and comments on key words:
  - o First pass: 44 words with frequency > 2
  - o Second & final pass: 23 words with frequency > 3
  - o Final ranked word list: Table 6.3
  - o The word "product" dominated the list, reflecting that from the time the application was fully integrated it was viewed as a "product". Words relating to the team of engineers working on designing features for the new application were prominent in the list ("team", "design", "engineer", and "feature").
- NVivo ranked word, sentence and phrase node count = 100
- Association of concepts coding used: +, -. +/-, =, e (e was dropped in reformatted map)
- NVivo PowerPoint cognitive map: Figure 6.12
- Reformatted PowerPoint cognitive map: Figure 6.13

The cognitive map highlights tasks and issues faced and resolved by the project team as it rushed to deliver NetDynamics 2.0 three months after the application was demonstrated as an incomplete prototype.

Exhibit 6.10 Text Extract: NetDynamics – NetDynamics Integration (March-June 1996)

By the end of February, the Studio team had the program framework up and running (without Java or networking capability) and started usability testing. By March 10th, the two component parts of the system (Studio and Runtime) had been integrated for the first time. The pressure for this integration came from Rinat, who had scheduled a meeting with analysts in New York to demonstrate the product. The team programmed the demonstration version to work only with a Netscape browser and a Microsoft database. While this initial version was a little fragile (and some features, such as networking, were not yet complete), it did show off the capabilities of the product from "end-to-end."

From early March onwards, the software was integrated into a working system each day. Every morning there was a "daily build," and everyone who needed the most recent baseline version of the program received a copy. The team adopted configuration management software to ensure this baseline product remained robust. Any new code added to the baseline was "checked-in" by the engineer, who first ran a number of local experiments to ensure that the new features did not interfere with the functioning of other parts of the system. When bugs were found, they were reported against the current version. A centralised system tracked and allocated these bugs to the design team. By the end of the project, the system had logged over 640 bugs since the daily build began.

78
Living on Internet Time 697-052

On April 8th, NetDynamics announced that the product would ship within 30 days. Even though it wasn't complete, marketing staff pushed for a training program in early April, which, due to the lack of documentation, had to be run by the same engineers developing the product. While this took the development team away from their primary task, scheduling the training session gave them valuable "hands-on" experience of user problems and requirements. As a result, they began working closely with 20 hand-picked beta customers who would stretch the limits of the product's performance and bring the product up to the required level of reliability. Some features were still in a state of flux, as were some major design issues. It had been expected that this would happen, however, as the market was evolving in a rapid fashion. "We have to expect things to change," said Malka, "and so do our engineers. They have to be prepared to retro-fit their work as we determine how the feature set has to change."

By late April, it became clear that the product would not be fully stabilised or "feature complete" by the end of the month. At this point, the engineering team decided to delay the release. During May, they worked fixing bugs, carrying out training with major customers, and performing another round of acceptance tests with "gamma" customers. During this period, NetDynamics learned to embrace those users that stretched their product to the limit, as this was often the way that they learnt where these limits were.

Exhibit 6.10 Text Extract: NetDynamics – NetDynamics Integration (March-June 1996) (Cont'd)

Customers often detected design flaws that NetDynamics would never have found until much later in the cycle, if at all. These users were as excited by the product's potential as NetDynamics' engineers. Indeed, one of them actually ended up sleeping in their offices for a week to maximize his interaction with the development team! Critical to such an approach however, was providing rapid fixes as problems emerged.

With such complex and advanced products, customers are often willing to accept some bugs in the software, but only if the technical support to resolve them can respond rapidly.
In mid June, as the product approached release, the development team began to ramp down, with many of the staff beginning to focus on new development projects. The chief engineer described the challenge for the future:

Development is a never-ending cycle of integrating new features and standards, and bringing them to the market. Given the rate of change in the industry, we cannot rest. HTML is dying now, and Java applets are taking over. As soon as this product is released, we have to start developing again. If we don't begin to sow the seeds of our own destruction next week, someone else will!

Source: Iansiti & MacCormack (1996, pp. 8-9).

Table 6.3 NetDynamics – From Integration To Release (March-June 1996) – Ranked Word List

Anchor Concepts	Frequency
product	13
team	8
design	6
engineer	6
feature	6
began	5
bugs	5
customers	5
end	5
new	5
system	5
they	5
we	5
would	5
work	5
integrate	4
Netdynamics	4
program	4
some	4
version	4

Source: data analysis performed using TACT (1997)

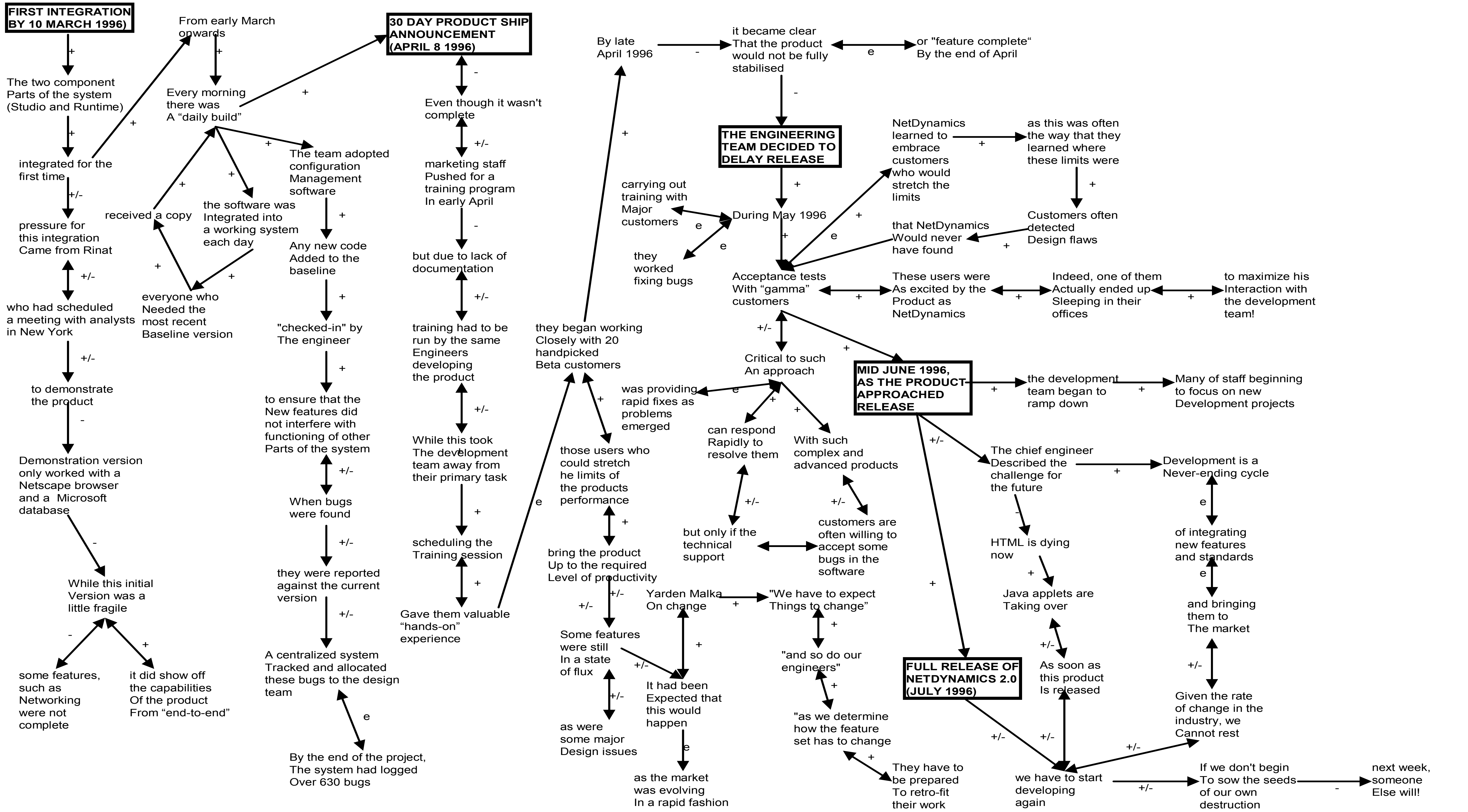


Figure 6.12 First Cognitive Map (NVivo Model) – NetDynamics – From Integration To Release (March-June 1996)
Created from data in extract (Exhibit 6.10)

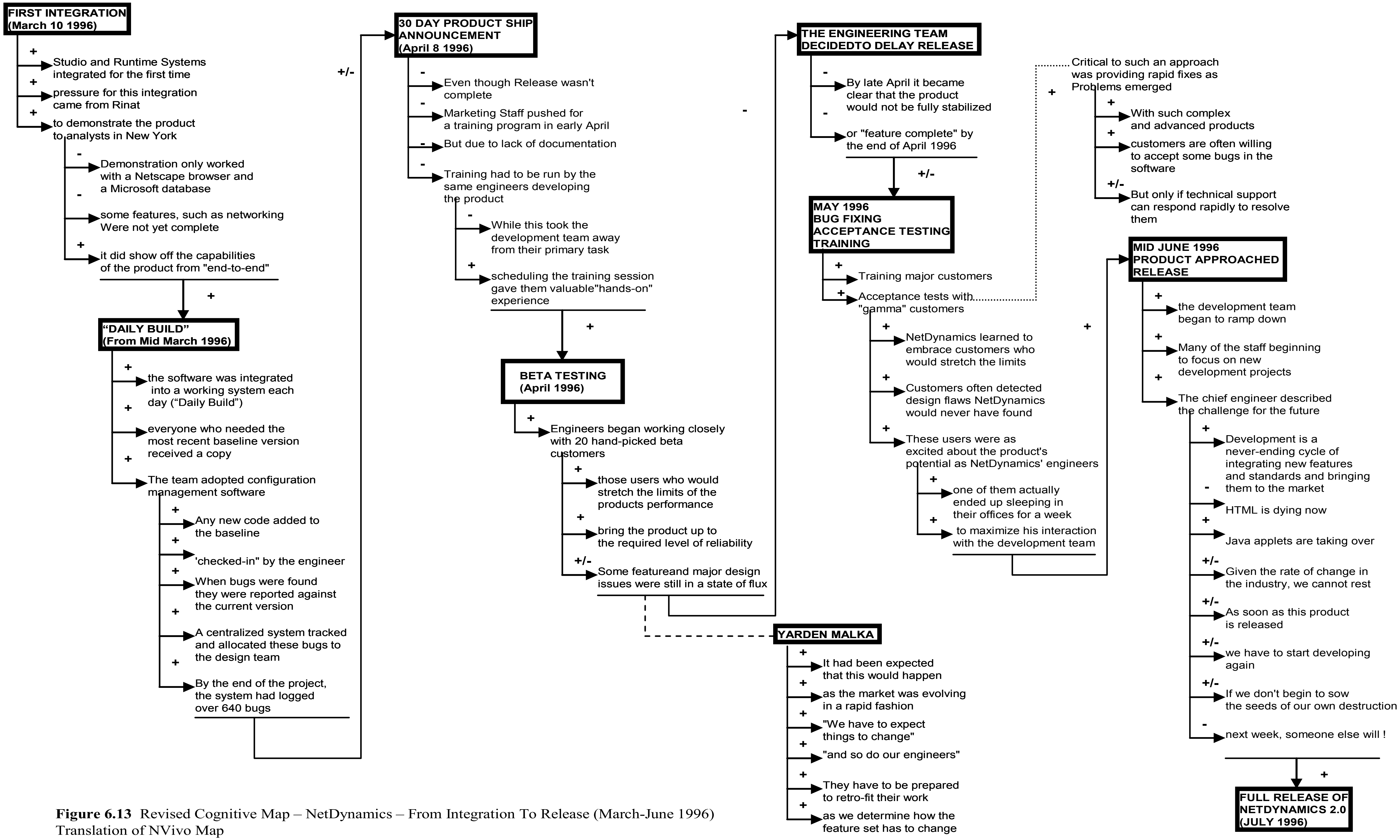


Figure 6.13 Revised Cognitive Map – NetDynamics – From Integration To Release (March-June 1996)
Translation of NVivo Map
Note: This revision was planned to be shown to participant in emic 2 stage

Most of the milestones in the map were highlighted in the events chronology map. The web page development group (Studio) and the application server/networking programs were integrated around March 10 1996, in time for a demonstration of the application to analysts in New York in mid-March.

The application was demonstrated with a Netscape Navigator browser linking with a Microsoft Database. Although the demonstration application was based on one computer system (that is, not a set of networked systems) it could show an end-to-end perspective of linking web pages to a database.

At about the same time as the demonstration, a formalised system of integrating the application daily was established. A "daily build" working copy of the application was distributed to anyone who needed to work with the latest version of the application.

Procedures for addition or modification of programming code into the current daily version of the application were established. Bugs were reported against the current version (not necessarily the version that was released with the bugs in it) and managed using a tracking system to log the bugs and allocate them to the design team to fix them.

About three to four weeks after the demonstration in New York, NetDynamics announced that NetDynamics 2.0 would ship in 30 days, that is, be released around May 8, 1996. The May launch date was in line with Rinat's target date set in December 1996. The announcement triggered a new set of pre-release activities, including training existing NetDynamics 1.0 users in the new application.

In established software houses designated groups of engineers or training staff conducted training using comprehensive documentation, usually in the form of application reference and training manuals.

Training in some new applications or radically different versions of applications may be conducted by engineers and/or trainers on a "roadshow" travelling to major cities and customers around a nation.

By the mid-1990s such documentation may be in hardcopy and on CDs and some software houses were starting to provide online documentation, either on a specific computer system, or available through the Web.

Given that the application was still under development, appropriate documentation was not available for training, meaning that engineers would have to conduct "hands-on" training with customers – but still continue to further develop the application at the same time.

However, combining training and development tasks turned out to be positive for the engineers as they too gained "hands-on" experience with the application with their customers.

During April 1996, after several daily builds, the application was deemed to have reached beta status. beta copies were released to 20 selected NetDynamics 1.0 users who had received training in the new application and were capable of pushing it to the limits of its performance, and to improve its reliability.

Even at the beta stage there were features and design issues that remained unresolved with the new application. Yarden Malka recognised that this would happen and with the market (that is, the Web environment and those developing applications for it) evolving rapidly, the engineers should expect significant changes in applications requirements and features. The engineers should be able to incorporate those changes in the current stream of application development.

Nevertheless, by late April, extra tasks associated with customer training, plus incorporating beta customer inputs into the new application meant that NetDynamics 2.0 would not be ready for full release by the declared date. The engineers decided to delay the release by about one month.

During May 1996, the engineers conducted further customer training. They worked with a subset of the beta customers (labelled "gamma" customers), conducting acceptance tests to push the application even further in attempts to discover more design flaws and bugs.

The "gamma" users were as enthusiastic about the new application as the NetDynamics engineers – one user actually worked and slept in the NetDynamics offices to work closely with the project team.

The strong commitment of the gamma customers to NetDynamics 2.0 was vital to the final stages of development as these customers were prepared to put up with some bugs as long as the NetDynamics engineers could respond rapidly by fixing these bugs.

NetDynamics was acting in a similar "beta/gamma" role for both Netscape Communications and Sun Microsystems for Netscape Navigator (versions 2 and 3) and Java respectively.

NetDynamics 2.0 was stabilised in June 1996 and was released in June-July 1996 – a development cycle of about six months. The project team began to ramp down and focus on other projects (most likely the next version of NetDynamics, version 3.0).

The chief engineer of NetDynamics recognised that future application development would have to be fast and be continuously integrating new features and standards in their applications.

By mid-1996, HTML was starting to fade for web page development with Java applets taking over to run applications within web pages and SGML as an enhanced mark-up language.

SGML did not take over from HTML as the more sophisticated XML/XSL language was preferred to it when it emerged in 1998-99. In 1996 the foundations of XML (SOAP) were in planning stages prior to development.

Developers such as NetDynamics had to stay across these changing standards and quickly incorporate them into new versions of their applications. Rapid adaptation often meant redeveloping applications to get to market first with more advanced applications.

So while NetDynamics had managed to develop a new application in six months, the next version, which could be completely new, would have to be developed in the same timeframe or faster if possible.

6.9.3.1 Updated Cognitive Map 3 – From Integration to Release (March-June 1996)

Following the interview stage, an updated cognitive map was created (see Figure 6.14). Most sections in the original map were validated with additional insights and comments.

Integration of the Studio (web page development) and Runtime (application server/networking) programs into a complete NetDynamics 2.0 application for the first time occurred around 10 March 1996, in time for a demonstration to analysts in New York (see box 1).

The demonstration was regarded by Rinat as successful as it highlighted the key objectives of the new application – users could, from within a Web page, access and use information from a database. The Netscape Navigator browser was used with a Microsoft Access database and SQL Server, based on one computer system.

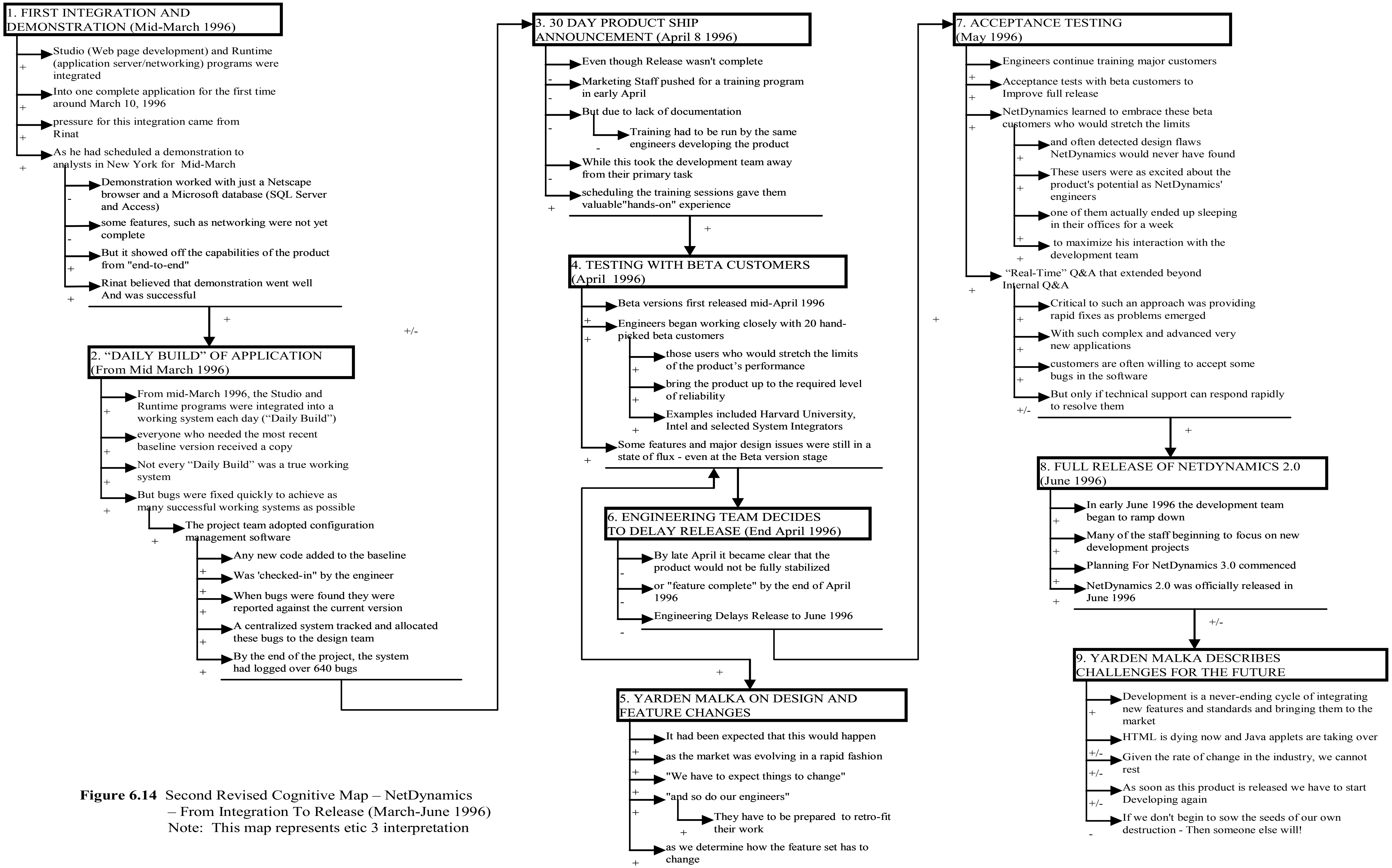


Figure 6.14 Second Revised Cognitive Map – NetDynamics
– From Integration To Release (March-June 1996)
Note: This map represents etic 3 interpretation

The "Daily Build" application section (box 1) was largely unchanged from the original map. There was a slight variation where Ebbs (2002b) indicated that not all daily builds turned out to be working systems, but that bugs were fixed quickly to achieve as many working systems as possible.

The Daily Build approach would have meant that around 20 to 30 daily builds would have been completed by the time the first beta versions were released in mid-April 1996.

The sections on the 30-day product ship announcement (box 3), testing with beta customers (box 4), Yarden Malka on design and feature changes (box 5), and the Engineering Team deciding to delay release (box 6), were almost unchanged except for some minor wording and formatting, and comments on examples of beta customers (box 4) and the delayed release date to June 1996 (box 6).

The section on acceptance testing (box 7) was revised to highlight that acceptance testing was really about improving the application with beta customers just before release.

Gamma customers were actually the same as beta customers. Ebbs (2002b) saw the acceptance testing as "Real-Time" Q&A that extended beyond internal Q&A. The subsection in the original map on providing rapid fixes was incorporated in the acceptance testing section with Ebbs' comments on Q&A.

The sections in the original map on approaching release and the actual full release of NetDynamics 2.0 were merged into one section (box 8). It was recognises that during June 1996, planning for NetDynamics 3.0 had commenced (it was released in October 1996). The actual release date for NetDynamics 2.0 was revised from July to June 1996.

The section in the original map on the Chief Engineer describing the challenge for the future was placed in a separate section (box 9) and attributed to Yarden Malka.

The original cognitive map reflects a detailed description of the application development process by Iansiti & MacCormack (1996), so the updated map does not change the processes, rather it expands some sections with additional insights and comments.

It also highlights that although NetDynamics was under heavy pressure to deliver the application within about six months from planning it, the project actually achieved delivery in six months, with just one decision to delay release for about one month.

Given the "beta" state of most of the applications required to build NetDynamics 2.0, achieving a June 1996 delivery was an extraordinary achievement.

While outside the scope of the cognitive maps presented for the case study, NetDynamics managed to develop the next version NetDynamics 3.0 with significant enhancements in four months from June to October 1996. After version 3.0, development cycles moved out to about 14 to 15 months as substantial enterprise computing enhancements were incorporated into these new versions.

6.10 Extension Questions

Interviewees for the NetDynamics case study were asked extension questions addressing Dominant Logic, Shared Vision, Leverage Points, and Strategic Marketing Principles.

6.10.1 Dominant Logic

During the interviews, the CEO of NetDynamics (Zack Rinat), and the Chief Engineer (Yarden Malka) were identified as dominating decision-making, or providing dominant logic throughout the development and delivery of NetDynamics 2.0.

Rinat commenced the initial application development that was incorporated into Spider Technologies and dominant in setting company directions, and launch dates for the application, but Malka was dominant in specific technical and development issues. Nanda Kishore was also identified as providing significant input from an engineering perspective.

6.10.2 Shared Vision

The interviewees agreed that there was a strong sense of shared vision across the whole company during the development of NetDynamics 2.0, with engineering and marketing ideas shared across the company.

There was a degree of sharing NetDynamics' vision with some groups in Sun Microsystems which may have been quite influential in NetDynamics choosing Java technology. Some NetDynamics engineers were given close working access to the Chief Technical Officer from Sun to understand about Java beta version developments.

As Sun was developing Java to be incorporated with Netscape Navigator, there was some sharing of vision with Netscape Communications, although on a much lesser scale than Sun.

NetDynamics not only shared its vision for NetDynamics 2.0 with its "lead-user" customers, but they developed objectives and features for the new application in conjunction with NetDynamics.

User commitment to NetDynamics 1.0 was high as there was strong demand for an application that could link web pages with corporate databases. The existing lead-users seemed to have a lot of input in terms of ideas for improvements, contributed to rigorous testing/QA for the new application.

NetDynamics also shared some of its ideas and vision for the new application with a range of System Integrators. System Integrators were effectively de facto marketers on behalf of NetDynamics for the application to their customers.

The theme of System Integrators being "marketing agents" for application developers, particularly in enterprise markets, also appeared as important for sharing vision in the Zaplet case study in this project.

6.10.3 Leverage Points

There were five leverage points identified during the interviews. Two leverage points were highlighted as critical points in Iansiti & MacCormack's (1996) case study, while three additional leverage points were identified from the interviews. The leverage points identified in chronological order are:

6.10.3.1 Decision to Extend Development Beyond Web Pages to Server Applications (Late 1995)

6.10.3.2 Selection of Java as the Scripting Language (November-December 1995)

6.10.3.3 First Integration (10 March 1996)

6.10.3.4 Testing and Feedback from the beta Customers (April-May 1996)

6.10.3.5 Acquisition of NetDynamics by Sun Microsystems (July 1998)

6.10.3.1 Decision to Extend Development beyond Web Pages to Server Applications (Late 1995)

Rinat's decision to shift development from Web Pages to include application servers in late 1995 were put at critical point in Web infrastructure development. There was a pressing requirement for additional server applications that could work across different computer systems – but the window of opportunity for development and delivery was quite short.

Sun Microsystems “seized the moment” by forging an alliance with Netscape Communications to incorporate its new Java technology to enable sophisticated multimedia content and other applications to be run within the Netscape Navigator browser. In fact the establishment of the Sun/Netscape alliance to develop a new browser could be regarded as an indirect leverage point, as this relationship appeared to have significant implications for other decisions for the development of NetDynamics 2.0.

Rinat realized that for the Web to be acceptable for enterprise computing solutions, the Sun/Netscape vision of a new Web environment required applications using a browser to access, view, and update corporate databases that were on multiple computer servers. Rinat’s decision to extend NetDynamics to include server application development put NetDynamics on a path to be the first ISV to develop such an application for the Web.

6.10.3.2 Selection of Java as the Scripting Language (November-December 1995)

This particular decision has been rigorously analysed through the models and maps for this case study, but it was validated in the interviews as the single most important decision in the development process. Perhaps the most intriguing aspect of this decision is when it was actually concluded.

Although Iansiti & MacCormack (1996) have described a selection and evaluation process for choosing Java technology running through December 1995, there are some indications that the decision to choose Java may have been concluded before and independently of this process.

The Sun/Netscape alliance (noted in the previous leverage point commentary) to enhance the dominant Web-browser at that time would have set off a ripple effect in Web developer communities such that any prospective ISV intending to develop new Web applications had to consider using the Java technology ahead of other languages and applications.

Perhaps the decision to choose Java, although concluded in late 1995, was predetermined from the time the Sun/Netscape alliance was announced in July 1996. Nevertheless, if the decision to go with Java was actually made then, there was still a high level of risk choosing to go with a technology that was so early in its development lifecycle.

Some other factors may have swayed an earlier decision for Java than in December 1996. The geographic and personal proximity of NetDynamics to Sun Microsystems and Netscape Communications would have meant that NetDynamics software engineers could gain access to Sun more easily than other providers of alternative languages – NetDynamics was located in Menlo Park with Sun Microsystems and Netscape Communications both located in Mountain View, approximately 5 miles from each other.

There was also a strong groundswell among the software engineers themselves to learn and apply the new Java technology to new applications – and the engineers with NetDynamics were probably part of the groundswell and keen to develop using Java almost at the outset of the project. Even so, they also understood that there were still high risks associated with developing an application with an early stage of an unreleased technology.

No matter when the decision to go for Java was concluded, it can be marked as a point where Web-based enterprise computing was born. The genealogy of the NetDynamics application can traced right through to Sun iPlanet Enterprise Computing Solutions which can be linked into the new Web Services application environments in 2002-03.

6.10.3.3 First Integration (10 March 1996)

Ebbs (2002a) saw the first integration of the Studio and Runtime programs as the point of emergence where the full application could be seen and appreciated.

The full integrated application was a real product that could be demonstrated, assessed by analysts and customers – and then rebuilt several times until it met the design objectives for it. For the engineers, the first integration shifted their perception of the new application from an idea toward a real product.

6.10.3.4 Testing and Feedback from the Beta Customers (April-May 1996)

NetDynamics worked very closely with a selection of existing NetDynamics 1.0 users to define the vision for the new application. However, a smaller group of lead-users worked with NetDynamics engineers to test and upgrade beta versions of the new application.

Each updated beta version incorporating enhancements recommended by the beta customer group could be regarded as a “micro-leverage point”, but collectively these points contributed to an radically changed application. Several “customer-requested” –

and "customer-tested" – features and enhancements were added during the beta phase of application development.

6.10.3.5 Acquisition of NetDynamics by Sun Microsystems (July 1998)

Although this decision was strictly outside the scope of the case study timeframe, it initially provided NetDynamics with significant extra funding and resources. The ISV, although a subsidiary of Sun, was able to continue enhancement of its applications through to a powerful enterprise computing application suite.

NetDynamics was incorporated into Sun's iPlanet division in 1999, with the NetDynamics application subsumed into the iPlanet suite of applications in 2000. Sun's iPlanet suite was superseded by Sun One (Sun's Web Services offering) in March 2002.

6.10.4 Strategic Marketing Issues

Four strategic marketing issues were identified during the interviews, and were related to:

6.10.4.1 Rapid Application Diffusion and "First Mover" Advantage

6.10.4.2 Pricing Evolution from Client to Server/Enterprise Applications

6.10.4.3 New Channels for Enterprise Applications – Especially System Integrators

6.10.4.4 Changes in Structure (and Ownership) to Support Rapidly Growing Enterprise Customer Base

6.10.4.1 Rapid Application Diffusion and "First Mover" Advantage

Development of The World-Wide Web was characterised by rapid Application Diffusion during the mid-1990s. Mass acceptance of the Netscape Navigator browser as the dominant browser for accessing the World-World Web rippled through to applications that enhanced browser features and Web Page development tools.

Java was able to ride with this diffusion wave, as was NetDynamics – as a separate company and its application within Sun's iPlanet enterprise solutions.

New Web development tools seemed easy to sell to developer communities eager to find the next new "killer application". Some of the new Web development applications of that time literally sold themselves to developers looking for new features and tools that would enable them to also develop new Web-based applications.

A prevailing perception of the mid-1990s was that those organisations with possible new killer applications could gain strong "first-mover advantage", and then their application could become an industry standard giving the organisation sustained market leadership.

This perception is in line with Moore's (1995) crossing of the "chasm" for high technology products from just being an interesting idea, into mass acceptance and diffusion – that is, "inside the tornado".

6.10.4.2 Pricing Evolution from Client to Server/Enterprise Applications

NetDynamics found that while pricing new client-based web development applications seemed to be easy, pricing server applications was a whole new ball-game.

NetDynamics shifted its pricing strategy from a low-end client-based perspective and focused on high-end positioning for server-based enterprise applications. Shifts in pricing strategy started with NetDynamics 2.0 – and increased as enterprise enhancements were incorporated in subsequent application versions.

6.10.4.3 New Channels for Enterprise Applications – Especially System Integrators

NetDynamics' transition toward focus on enterprise applications and customers prompted the ISV to develop a different channels strategy. A direct sales force was created to focus on large customers. NetDynamics established a substantial (Value Added Reseller) VAR network for smaller and industry-specific customers.

Certain System Integrators (SIs) were identified as critical for consulting on the design and implementation of enterprise computing solutions. NetDynamics developed strategies to target those SIs, offering critical application assistance for the development of customised Web-related enterprise solutions.

Although NetDynamics possessed one of the main essential applications for Web-related enterprise solutions, most large SIs spent the second half of the 1990s implementing non-Web-based enterprise solutions. NetDynamics initially struggled to sell their Web-based enterprise vision to key SIs.

However, a wave of new-generation SIs committed to developing new Web-based enterprise computing solutions embraced the NetDynamics visions and effectively became de facto marketers for NetDynamics.

6.10.4.4 Changes in Structure (and Ownership) to Support Rapidly Growing Enterprise Customer Base

NetDynamics' "First-Mover" advantage position with the NetDynamics 2.0 application opened up opportunities for alliances-based development of enterprise e-commerce applications suites. Development and delivery of enterprise e-commerce suites meant substantial investments in application development, channels delivery options, and customer support.

Investment and ongoing management of a rapidly growing enterprise customer base is a fundamental marketing and business management challenge for a young software house. Software houses may attempt to address the issue through deepening alliances with SIs and other specialist developer groups.

They may attempt to gain extra funding through going public through an IPO, or through other sources. Another obvious option is to seek investment from other larger hardware, software or e-commerce companies and to gain access to their established customer support development resources and infrastructure. A common trade-off for this option is the full acquisition and incorporation of the company and its applications further down the track.

In late 1996 – early 1997, NetDynamics sought assistance from Sun. Sun invested in NetDynamics, thus providing the required funding and resources for ongoing development and customer management. NetDynamics then developed and released the substantially expanded V4.0 under these conditions. However, after about 18 months, it made more strategic sense for Sun to bring NetDynamics "in-house" and into the iPlanet Group.

NetDynamics the company may have ceased to exist but its genetic imprint lives on in Sun's current enterprise and Web Services computing solutions.

6.11 Summary

The NetDynamics case study maps and provides insights into high-speed and high-intensity application development within a context of an emerging but not well-defined environment (in this an emerging computer-mediated environment).

Decision-making from conceptualisation of an idea to use web-pages as an access-point for databases, to a full application that could also run across multiple computer systems, was captured and mapped in a DSA package of models and maps. The timeframe for all of these decisions was approximately seven to nine months – rapid by any development measurement.

Key insights from the case study include:

6.11.1 Diffusion of Key Application Technologies for Emerging Computer-mediated Environment

6.11.2 Multi-dimensional ISV Concurrent Application Development

6.11.3 Systems Integrators as Extended or Virtual Application Marketers

6.11.4 Startup Business Management of Large Customer-Installed Bases over Time

6.11.1 Diffusion of Key Application Technologies for Emerging Computer-mediated Environments

During the mid-1990s the World-Wide Web represented an emerging computer-mediated environment (CME). New applications and technologies were researched, trialled, and delivered – of which a small subset would become integral components on the new environment.

Development within this context was a very high-risk proposition. Bets were put on technologies that might be become development tools of choice for the emerging (CME).

The NetDynamics case study highlights a perception that Netscape dominated Web browser technology (at least for 1995-96), and Sun's Java language would become a fundamental Web application development technology.

NetDynamics was able to create the first server-based Java application. Moreover, the "killer application" was a key enabler for e-commerce activity, that is, enabling businesses to link up their databases with Web pages, across multiple computer operating-system environments.

NetDynamics was successful in that its application was rapidly accepted, diffused and incorporated into Web technology.

6.11.2 Multi-dimensional ISV Concurrent Application Development

At the time of the focus of the NetDynamics' Case study, 1995-96, Web browser technology was still in its formative stages. All developers were both experimenting and rushing to produce new application versions with quantum leaps in functionality and features.

Whether developers were working individually or as alliances or consortia, most of their applications were in very early stages of development. It was not unusual then for ISVs to use early beta versions of applications within their applications, while also sharing very early versions of their own packages with their ISVs.

Multi-way beta-ISV application development is well represented in the NetDynamics case study. Netscape (Browser), Sun Microsystems (Java) and NetDynamics (NetDynamics 2.0) were effectively acting as beta-testers for each other, creating fixes, patches and new tools for all three applications.

The case study provides limited insight into "closeness" of relationships between the three ISVs. NetDynamics appears to have been closer to Sun, working with very early versions of Java, but that closeness was not in the form of a deep and established contractual relationship. It was close in terms of the high interest in an application that could greatly enhance acceptance of Java (and the Web).

6.11.3 Systems Integrators as Extended or Virtual Application Marketers

The NetDynamics case study highlights challenges faced by start-up application developers to market their products and services to large enterprise customers. Most small software houses have sales and marketing resources limited to a few salespersons, product managers and product marketers, and maybe one marketer focused on marketing communications.

These emerging software houses may then depend on establishing strong relationships with System Integrators (SIs) who may incorporate their applications in their enterprise e-commerce solutions. The SI may have its own developer and delivery expertise in the ISVs applications; they may effectively act as marketers or salespersons for them.

The SI as an extension of the ISVs marketing resources is an important principle for enterprise application development and ongoing management and for expansion into markets where the ISV cannot cover it. Rinat (2002a & b) noted SIs as critical for NetDynamics for servicing, supporting, but especially marketing a rapidly emerging enterprise customer base.

6.11.4 Startup Business Management of Large Customer-Installed Bases over Time

The NetDynamics case study also highlights further challenges associated with managing rapidly growing enterprise customer bases. While SIs are important enablers for enterprise customer bases, start-up software houses must grow to a critical mass to stay in business.

Some software houses or computer vendors have grown over time to create a sustainable infrastructure to support and manage substantial numbers of enterprise customers. However, significant numbers of software houses created in the “Internet era” (1994-2000) grew rapidly but not enough to build up sufficient support infrastructure.

Several software houses ran out of cash or capital and did not survive. Some software houses survived or grew through mergers (Kana in this project). Other software houses were acquired either quickly or over time by larger organisations with established infrastructure (for example, Sun, IBM, Microsoft).

NetDynamics struggled to manage its enterprise customer base, even with support from its SIs. Sun saw great enterprise potential for NetDynamics and acquired it in 1998. Seven months later, the whole NetDynamics business and its products and services were integrated into Sun’s iPlanet Division where there was a strong infrastructure to support global enterprise customers and partners, including large SIs.

There was no clear formula or set of principles discerned from the case study addressing factors, variables or issues that may be important for sustainability of software houses focused on enterprise applications, although organisational structures, resources and decision-making capability may be different for an emerging software house compared to an organisation with a large support infrastructure.

Although not readily apparent in the NetDynamics case study but relevant to most other case-studies in the project, there is another perspective related to entrepreneurial disposition of the founders of start-up software houses. Rather than transitioning start-ups to an established organisation with large supporting infrastructure, founders may be more attracted to moving and setting up new ventures.

In the case of NetDynamics, Rinat had moved on to start at least one new software house since NetDynamics. A number of founders or early developers in other cases in the project (for example, Kana, Intuit QuickBooks) had created new businesses over the last four to five years.

Different management teams are often required for either transitioning a software house toward a larger organisational structure or if it is to be acquired or merged into another organisation.

6.12 Conclusions

The NetDynamics case study highlights direct decision-making associated with rapid application development with extension to ongoing application delivery and support.

Conclusions are presented in the sections below:

6.12.1 Theoretical Issues Addressed in the NetDynamics Case Study

The storytelling framework used for analysis in the NetDynamics case study, was effective at identifying key decision-making issues, events, and linkages associated with development and delivery of the NetDynamics 2.0 application.

The methodology provided mapping and validating of initial accounts of decision-making, followed up with subsequent revisions of those accounts using a systematic emic/etic representation system. An extra sixth level of hermeneutic analysis was developed for the NetDynamics case study, whereby selected cognitive maps were submitted to a panel of experts for evaluation using a humanistic inquiry framework. The selected cognitive maps were validated as acceptable in terms of the process for creation and depiction of data from the original case study extract accounts.

The mapping of decision-making associated with creating a "killer application" for and with a young and rapidly emerging computer-mediated environment is addressed in the NetDynamics case study.

An extension of application development is the ongoing support of an emerging enterprise customer base. Significant insights were gained from the interviews on the challenges faced by NetDynamics with managing enterprise customers.

Extension questions added insights for developing a software house focused on servicing enterprise customers. System Integrators (SIs) were very important to software houses as extended providers of developer resources – but also as extended or virtual marketers for the software house.

Even with extended SI resources, transitioning a small software house toward a large enterprise-focus remains very difficult, with many not surviving or being acquired by larger more established players.

There may be an entrepreneurial disposition dimension where founders may prefer to move on to new ventures rather than transitioning a start-up business into a larger organisation. However, this area was largely beyond the scope of the case study.

6.12.2 NetDynamics Case Study Limitations

The NetDynamics case study is one of the complete cases in the project for multiple interviews and full validation and updates for the DSA package.

Nevertheless, the case study may benefited from additional interviews from NetDynamics staff (including the NetDynamics Chief Scientist), plus developers at Sun

Microsystems and Netscape who were working together with NetDynamics during 1995-96.

There may have been further benefits through attempting to write up and map a more comprehensive cognitive map focusing on NetDynamics evolution from 1996 through to its incorporation into Sun Microsystems' iPlanet Group in 1999.

6.11.3 Contributions from the NetDynamics Case Study to Strategic Sensemaking

The NetDynamics case study maps DSA in three main dimensions:

- Mapping of very rapid product development where enabling application development technology is under simultaneous rapid development ("product development in Internet Time")
- Mapping of new business development from a new idea through to a new application and then onto acquisition by a large high-technology company
- Mapping of issues associated with growing rapidly toward a critical mass to support large enterprise customers.

6.12.4 Contributions from the NetDynamics Case Study to Practice

The NetDynamics case study provides insights into the very high risks associated with developing applications and technologies for very new infrastructures such as the World-Wide Web. The risks highlighted include:

- very short windows for development of new applications
- Selection and learning about new relatively untested enabling development technologies
- Managing product development within multiple ISV alliances – especially where every ISV is concurrently testing each other applications
- Effective prototyping of applications in order to seek more investment, funding and support for further complete application development.

6.12.5 NetDynamics Case Study Endnote

NetDynamics V2.0 won several awards in 1996-97 including:

- NetworkWorld and IntraNet Magazine 1996 Intranet excellence Award
- Computerworld Internet Commerce Expo – NetDynamics Wins two "Best-in-Class" awards
- Gartner Group and InformationWeek iEC Solutions award for the best intranet application.

- PC Magazine Editors' Choice Award for the Best Web Database Tool (SUN, 2003)

NetDynamics Versions 3 and 4 also won several awards before the company was acquired by Sun in July 1998 for a reported $US160 million. NetDynamics V5.0 was migrated to I-Planet's I-Application Server iAS V6.0 which shipped from April 2000. Sun Microsystems announced its Sun Open Net Environment (Sun One) on 5 February 2001, which incorporated iAS V6.0.

The latest version, Sun One V7 (iAS v7) was released in October 2002. Sun One represents a consolidation of application server technologies into a new open-source application development environment.

Chapter 7: Case Study – Kana Communications

Abstract

The Kana Communications (Kana) case study applies using two storytelling methods for analysis – an **advanced hermeneutic framework**, and an **extended form of decision systems analysis (DSA)** incorporating **cognitive mapping** – an exploration of sustained and dynamic rapid software application development. Established DSA is augmented by cognitive mapping to provide rich insights into the development of seven full versions of a rapidly expanding application suite over a four-year timeframe.

A key aspect of decision-making in the Kana case study is rapid application development and expansion through mergers and acquisitions. The consolidated Kana Inc. of mid-2001 is a culmination of several mergers and acquisitions – all of which have enabled Kana to offer a comprehensive enterprise e-commerce application suite (a full "e-CRM" application suite).

Other key findings in the case study include: management of channels of distribution to support a rapidly expanding enterprise customer base; the importance of Systems Integrators (SI) as an extended set of developer, sales and marketing resources; and raising a new concept of "Channels of Development".

7.1 Background

Kana Communications Inc. (Kana), of Menlo Park California provides CRM applications designed to operate in and with e-business (or Internet-based) enterprise applications (positioned by Kana as e-CRM applications).

The Kana Communications organisation of 2002 is the culmination of several large mergers and acquisitions, the most recent being between Kana Communications and Broadbase Inc. in April 2001.

The Kana Communications (noted throughout the remainder of this study as **"Kana"**) case study traces the rapid emergence of a software house initially with an application for e-mail management through to being a provider of an extensive customer communication management application suite. The outline and chapter structure for this case study is presented in Figure 7.1.

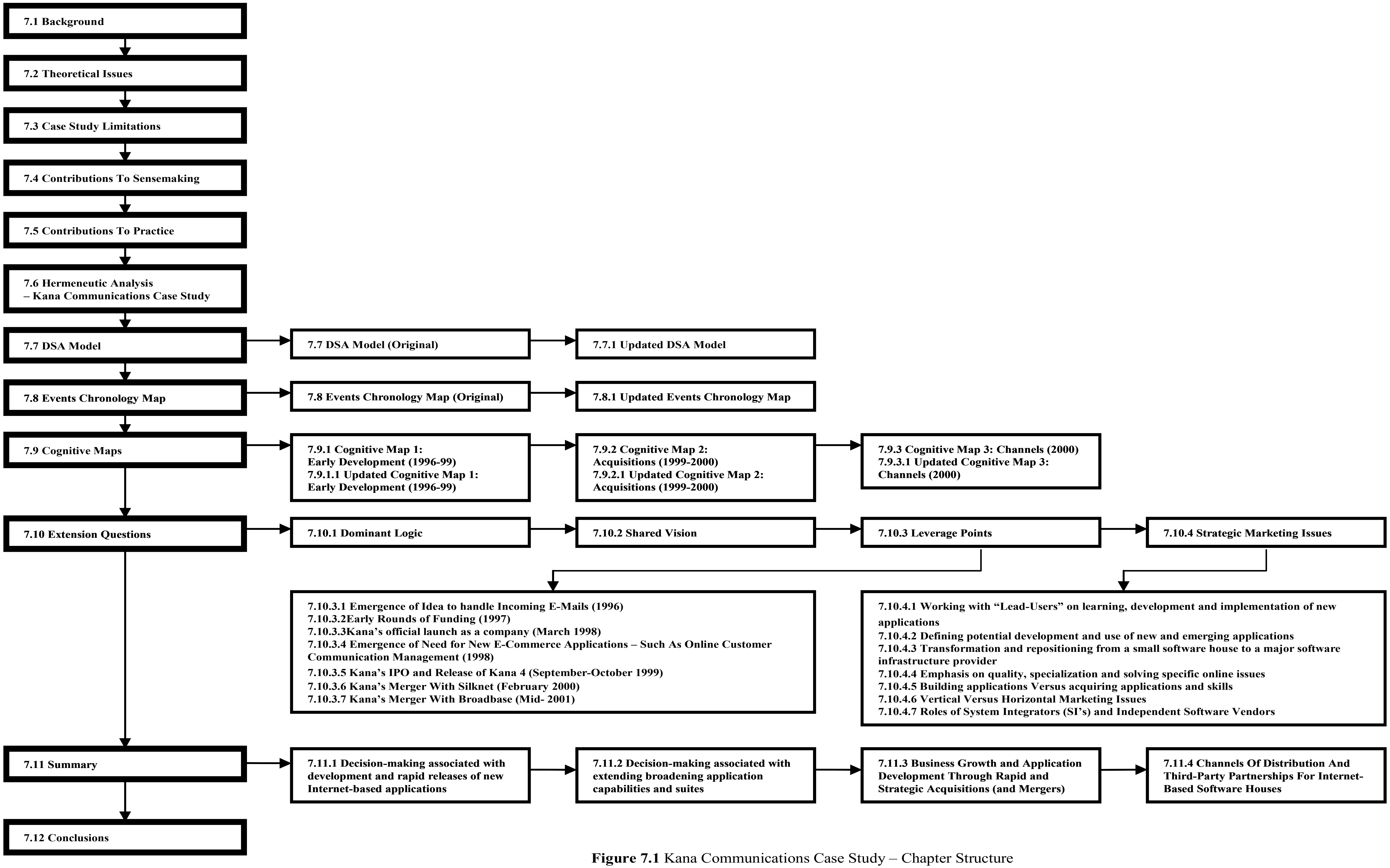


Figure 7.1 Kana Communications Case Study – Chapter Structure

Kana was included in the final case study list for the project as it met the initial criteria:

- Software has to be "beta"-tested and commercially introduced
    - The case study addresses five commercially released versions of Kana's main application
- Case-studies were written by well-known authors
    - Professor Miklos Sarvary is recognised as an expert on the marketing of information products with special emphasis on content generation (knowledge management)
- The application development process studied must be over a timeframe of more than 12 months
    - the original case study covers a timeframe of just over four years of several versions of Kana's applications from 1996 to 2000
- Names of case study companies and participants can be disguised if requested
    - Some identities in the original Harvard Business School Case study were disguised. However, permission was gained from Mark Gainey and Michael Wolfe, both for a combination of face-to-face and digitally recorded telephone interviews and use of that information in the research project, without disguise
- Applications must be Internet-based
    - Kana developed applications to manage Internet-based communications that evolved into early e-CRM systems, which are both Internet-based application managing online customer communications activity

Additional case study selection criteria included:

- Size: Small
- Success: Yes (through at least two substantial mergers with Silknet and Broadbase)
- Disruptive Technology: Yes, related to e-mail and online communication management
- Harvard Business School Case: "Kana Communications", Sarvary (2000)

Sarvary (2000) describes the creation of Kana from 1996 through to February 2000, highlighting Kana's meteoric rise to market leadership in customer management applications; a highly successful IPO toward the peak of investments in Internet stocks in 1999; and further acquisitions of software houses to build up Kana's application capabilities into a full e-commerce application suite.

Sarvary (2000) also reviewed Kana's distribution channels and a key strategic decision in 2000 related to the development of an Application Service Provider (ASP) division.

Kana's original application in 1996 was focused on inbound e-mail only, but by late 1999, newer versions of the product contained full inbound and outbound e-mail management.

Such was the rapid pace of application development and software house acquisitions that within three to six months of gaining outbound e-mail management capability, later versions of Kana's product incorporated complete customer communications management, and connectors to other major e-commerce solutions.

Indeed, by February 2000, the acquisition of Silknet by Kana provided an opportunity to create an integrated e-commerce application suite. However, the actual integration of Silknet's applications into Kana's application suite was beyond the scope of Sarvary's (2000) Case study.

A description of Kana's application suite in early 2000, Kana V5 can be found in Exhibit 7.1.

Sarvary covers application development in the context of Kana products in the case study. Compared to other case-studies used in this project, the description of application development is small but sufficient for developing a representative DSA model and conducting further analysis on application development and delivery.

The DSA model and maps in this case were updated, plus extension questions on dominant logic, shared vision, key leverage points and strategic marketing issues were addressed through interviews with former Kana executives (see Gainey & Wolfe 2002; Gainey 2002; Wolfe 2002).

Exhibit 7.1 Text Extract: Kana 5 Application Suite

Kana 5 included the following modules:

- **Kana Assist**, an online self-service solution that improved the customer experience by delivering context-sensitive answers to customer questions directly on the web site – allowing customers to quickly and conveniently obtain answers to their questions without the intervention of a customer service representative.
- **Kana Classify**, a powerful, advanced and highly scalable artificial intelligence technology that enabled automatic responses to customer e-mail. The system's dynamic customer database provided an efficient system to store customer information, identify needs and proactively target individuals with personalized messages.
- **Kana Connect**, an electronic direct marketing solution that enabled e-businesses to proactively deliver individually targeted messages to increase the lifetime value of customers. It enabled marketers to profile, target and engage customers in one-to-one conversations through permission-based, e-mail communication.
- **Kana Forms**, a web-form communication technology that input web site form information directly into the Kana system while preserving the structure of the data. Once the information was in Kana, it could be used to enhance workflow automation, integration, and personalization.
- **Kana Link** rapidly integrated a broad range of systems, including call center systems, customer care applications, and commerce systems. It enabled the customisation of Kana Response to share valuable information between applications. Kana customers could share the integration information, answers, and solutions among all customer enterprise systems to maximize productivity and reduce costs.
- **Kana Notify** extended customer retail experience by automating, customising and managing transaction-related communications such as order status and receipts, and helped reduce customer service inquiries by sending proactive messages.
- **Kana Realtime**, an e-commerce solution that enabled companies to engage in one-to-one real time communication with customers to generate higher sales conversions. The solution's live two-way web-based dialogue between the company and customer provided immediate online sales assistance to help companies turn browsers into buyers, reduce shopping cart abandonment, cross-sell and up-sell during the sales cycle, and reduce sales service costs.
- **Kana Response**, a high performance e-mail customer service solution that enabled e-businesses to respond to, monitor, and manage more than 350,000 e-mail messages a week.

Source: Sarvary (2000, pp. 3-4).

7.2 Theoretical Issues (or Questions) for the Kana Case Study

The Kana case study highlights decision-making issues associated with rapid product development through acquisitions. Kana moved from one simple application to a full e-commerce application in a three-year period through several acquisitions, and rapid incorporation of new applications and application development capability into its organisation.

The case study also explores the selection of channels of distribution and development for rapidly emerging applications.

The storytelling methodology used for analysis in the case study – an advanced hermeneutic analysis, including an extended form of **decision systems analysis** (DSA) incorporating **cognitive mapping** – were effective at identifying key decision-making issues, events, and linkages.

The methodology provided mapping and validating of initial accounts of decision-making, followed up with subsequent revisions of those accounts using a systematic emic/etic representation system.

7.3 Kana Case study Limitations

The Kana case study contains a set of interviews that effectively address issues within the 1996-99 timeframe of the original Sarvary (2000) account. However, there were no respondents who are currently with Kana (2002 or 2003).

The case study may have benefited from additional useful insights through interviews with developers and managers in companies acquired by Kana.

7.4 Contributions from the Kana Case Study to Strategic Sensemaking

The Kana case study explored decision-making issues, events, and linkages using the storytelling method developed for the project. An advanced hermeneutic analysis provided several levels of emic/etic representation of decision-making, highlighting key case study issues.

The Hermeneutic framework incorporated an extended form of Decision Systems Analysis (DSA) highlights specific subsets of decision-making, through mapping of decision-making from original accounts with revisions, through subsequent contact and discussions with original decision-makers. The extended DSA also explored changes over time in shared vision and new application development.

The Kana case study contributes DSA augmented by cognitive mapping. This provides insights into decision-making for:

- "scalability" of dynamic decision-making associated with rapid expansion of software applications
- identification and incorporation of third-party applications into an emerging suite of applications
- conversion of an idea or opportunity into a large enterprise-wide suite of applications
- determination of channels of development and distribution

7.5 Contributions from the Kana Case Study to Practice

The Kana case study provides practical insights into the conversion of a business idea into a specific application, and then rapidly scaling it up to an enterprise-wide e-commerce application suite.

There is particular emphasis in the case on a company growth strategy focused on the acquisition and effective incorporation of acquired development skills and application into an ongoing and rapidly growing organisation

The Kana case study also highlights the importance of channels of distribution and, a new concept, channels of development. Small software houses can significantly extend their developer, sales and marketing resources through intensive channel selection and management.

7.6 Hermeneutic Analysis – Kana Case Study

The Kana case study offers an application of the proposals described in Chapter 1 for advancing hermeneutic research in B2B decision-making.

Five levels of hermeneutic analysis were developed to address decision-making issues, events, and linkages in the Kana case study. These levels are presented in Exhibits 7.2 to 7.6.

The original case study by Sarvary (2000) serves as the etic 1 report of the case study – although there are elements of emic 1 reporting in the form of direct quotes from decision-makers in the account. Quotes and perspectives from decision-makers were encapsulated into the Level 1 emic analysis in Exhibit 7.2.

Exhibit 7.2 Level I Analysis of Kana Communications Case

Emic 1 view:
•Transition from accidental discovery of opportunity – managing e-mails – to new application
•Development of online communications management application mainly through acquisition of software houses with appropriate applications
•Founder (Gainey) always thought that Kana should add outbound e-mail management capability to its offering
•Kana wanted to be a leader offering a platform solution that could link online communications products to additional e-commerce applications
•Three pieces of infrastructure needed to build a billion-dollar e-commerce operation – content, to manage the transaction process, and manage the communication process
•Comprehensive channels options including Direct Salesforce, Key Business Partners and Kana Online (KOL)
•Investigation of Application Service Provider (ASP) channel option

↓

1996-2000. Initial mental model:
•Initial business opportunity identified as setting up online storefronts for Sports Goods manufacturers
•Identified a different issue – complaints associated with managing e-mails
•Rapid application development directly with customers
•Rapid expansion into online communications management through acquisition of software houses with relevant applications
•Transformation to e-commerce provider through further acquisitions and mergers
•Establishment of channels to service enterprise customers
•Exploration of Application Service Provider (ASP) model
Decisions-actions:
•Worked directly with customers to manage e-mails; Developed Kana V1 (1996-99); Expanded online communications management capabilities through acquisitions of Connectify, BEI, NetDialog; IPO Sep 1999; Move to full e-commerce application provider through merger with Silknet (2000); Kana V5 released (1999-2000); investigation of new channels options

Exhibit 7.3 Level II Analysis of Kana Communications Case

Etic 1 view:
•Email management opportunity found while seeking different opportunity (online shopfronts for sports goods manufacturers)
•Expansion to inbound and outbound email management through acquisitions
•Transformation to supplier of e-commerce application through acquisitions and mergers with funding from an IPO
•Channels options focused on expanding Kana Online (KOL) and developing an Application Service Provider (ASP) group

Emic 1 view:
•Transition from accidental discovery of opportunity – managing emails – to new application
•Development of online communications management application mainly through acquisition of software houses with appropriate applications
•Founder (Gainey) always thought that Kana should add outbound e-mail management capability to its offering
•Kana wanted to be a leader offering a platform solution that could link online communications products to additional e-commerce applications
•Three pieces of infrastructure needed to build a billion-dollar e-commerce operation – content, to manage the transaction process, and manage the communication process
•Comprehensive channels options including Direct Salesforce, Key Business Partners and Kana Online (KOL)
•Investigation of Application Service Provider (ASP) channel option

1996-2000. Initial mental model:
•Initial business opportunity identified as setting up online storefronts for Sports Goods manufacturers
•Identified a different issue – complaints associated with managing emails
•Rapid Application Development directly with customers
•Rapid Expansion into Online Communications Management Through Acquisition of software houses with relevant applications
•Transformation to e-commerce provider through further acquisitions and mergers
•Establishment of Channels to service Enterprise Customers
•Exploration of Application Service Provider (ASP) model
Decisions-actions:
•Worked directly with customers to manage emails; Developed Kana V1 (1996-99); Expanded online communications management capabilities through acquisitions of Connectifi, BEI, NetDialog; IPO Sep 1999; Move to full e-commerce application provider through merger with Silknet (2000); Kana V5 released (1999-2000); investigation of new channels options

Key case study issues as articulated by Sarvary (2000) were added to the emic 1 data in the Level II analysis and presented as etic 1 data in Exhibit 7.3. These issues included:

- E-mail management opportunity found while seeking different opportunity (online shopfronts for sports goods manufacturers)
- Expansion to inbound and outbound e-mail management through acquisitions
- Transformation to supplier of e-commerce application through acquisitions and mergers with funding from an IPO

- Channels options focused on expanding Kana Online (KOL) and developing an Application Service Provider (ASP) group.

The Level II analysis also explored mental models and decisions – actions associated with these issues.

The third level of analysis, as presented in Exhibit 7.4, is an etic 2 representation based on the current researcher's summarisation of event milestones and the emic 1 sensemaking views identified in the data in the Sarvary (2000) case study. Subsequently to developing Exhibit 7.4, emic 2 and etic 2 interpretations along with DSA and event maps provide revisions to Exhibit 7.4.

Sarvary's (2000) account highlights Kana's founder (Mark Gainey) identifying an unmet need for e-mail management. Creating both an application and a business to meet and enhance that need was sufficient for the development of an initial (etic 2) representative DSA model and to undertake further analysis on application development and delivery.

The initial DSA, event chronology, and cognitive maps reported below offer details supporting Exhibit 7.4. Etic 2 perspectives are mainly based from the initial DSA model, initial events chronology map, and initial cognitive maps.

Key etic2 issues included:

- Founders worked directly with early customers to development an inbound e-mail management application
- Focus on the rapid acquisition of software houses to build up online communications application capability
- Management combined efforts on acquisition and application development during the IPO process
- Adaptation of strategy toward offering a complete e-commerce application suite (still with an emphasis on online communication management)
- Constant acquisitions and mergers to build up critical mass and application development and service capabilities
- Channels Partners for extended sales/marketing and developer resources (particularly SIs).

Exhibit 7.4 Level III Analysis of Kana Communications Case

Etic 1 view:
•Email management opportunity found while seeking different opportunity (online shopfronts for sports goods manufacturers)
•Expansion to inbound and outbound email management through acquisitions
•Transformation to supplier of e-commerce application through acquisitions and mergers with funding from an IPO
•Channels options focused on expanding Kana Online (KOL) and developing an Application Service Provider (ASP) group

Etic 2 view:
•Founders worked directly with early customers to development an inbound email management application
•Focus on rapid acquisition of software houses to build up online communications application capability
•Management combined efforts on acquisition and application development during IPO process
•Adaptation of strategy toward offering a complete e-commerce application suite (still with emphasis on online communication management)
•Constant acquisitions and mergers to build up critical mass and application development and service capabilities
•Channels Partners for extended sales/marketing and developer resources (particularly SI's)

Emic 1 view:
•Transition from accidental discovery of opportunity – managing emails – to new application
•Development of online communications management application mainly through acquisition of software houses with appropriate applications
•Founder (Gainey) always thought that Kana should add outbound e-mail management capability to its offering
•Kana wanted to be a leader offering a platform solution that could link online communications products to additional e-commerce applications
•Three pieces of infrastructure needed to build a billion-dollar e-commerce operation – content, to manage the transaction process, and manage the communication process
•Comprehensive channels options including Direct Salesforce, Key Business Partners and Kana Online (KOL)
•Investigation of Application Service Provider (ASP) channel option

1996-2000. Initial mental model:
•Initial business opportunity identified as setting up online storefronts for Sports Goods manufacturers
•Identified a different issue – complaints associated with managing emails
•Rapid Application Development directly with customers
•Rapid Expansion into Online Communications Management Through Acquisition of software houses with relevant applications
•Transformation to e-commerce provider through further acquisitions and mergers
•Establishment of Channels to service Enterprise Customers
•Exploration of Application Service Provider (ASP) model
Decisions-actions:
•Worked directly with customers to manage emails; Developed Kana V1 (1996-99); Expanded online communications management capabilities through acquisitions of Connectify, BEI, NetDialog; IPO Sep 1999; Move to full e-commerce application provider through merger with Silknet (2000); Kana V5 released (1999-2000); investigation of new channels options

The etic 2 DSA model and maps for the case were updated following extensive questioning of the accuracy and completeness of the original Sarvary's (2000) case – the collected data represents emic 2 material and was added during the fourth level of analysis (See Exhibit 7.5).

Thus, additional (emic 2) data were collected for etic 3 description and interpretation of the decision-making process as reported in the original case study and for a period of one year beyond that reported by Sarvary. Emic 2 data consists of responses from interviews with the Founder of Kana and the then VP of Development. The initial long interview was in fact a joint meeting with these respondents (see Gainey & Wolfe 2002; Gainey 2002; Wolfe 2002).

Exhibit 7.6 is a fifth level of analysis with the addition of etic 3 data summarising emic 2 interpretations of mental models and events covered in the original case study as well as the work completed for the etic 2 interpretation – including the DSA, event chronology, and cognitive maps developed for the etic 2 interpretation.

The revised DSA, event chronology, and cognitive maps presented in subsequent sections below follow from the emic 2 interpretations, and these maps are part of the etic 3 interpretation.

The new data from the emic 2 and etic 3 rounds of interpretation validated etic 2 data and highlighted new insights related to:

- Decision-making for rapid development andrelease of internet-based applications
- Importance of System Integrators (SIs) and ISVs in marketing, referral and further application development and customisation
- Broadbase effectively took over Kana but kept building up the Kana brand and applications
- Kana survived as a leading e-CRM provider repackaging its applications into Kana iCare in mid-2001 and releasing Kana IQ V8 in Sep 2003

Overall, the five-level hermeneutic framework for the Kana case study provided a strong and comprehensive framework for capturing and extending dynamic sensemaking for Kana's decision-makers over a number of phases of application development, evolution, and business growth.

Exhibit 7.5 Level IV Analysis of Kana Communications Case

Etic 1 view:
- Email management opportunity found while seeking different opportunity (online shopfronts for sports goods manufacturers)
- Expansion to inbound and outbound email management through acquisitions
- Transformation to supplier of e-commerce application through acquisitions and mergers with funding from an IPO
- Channels options focused on expanding Kana Online (KOL) and developing an Application Service Provider (ASP) group

Emic 2 view:
- Importance of working with "Lead-Users" on learning, development and implementation of new applications
- Focus on transformation and repositioning from a small software house to a major software infrastructure provider
- Emphasis on quality, specialization and solving specific online issues
- Balancing between building applications Versus acquiring applications and skills
- Focus on special application development and porting efforts for major hardware and technology partners
- Roles of System Integrators (SI's) and Independent Software Vendors as extended sales & marketing and developer resources

Emic 1 view:
- Transition from accidental discovery of opportunity – managing emails – to new application
- Development of online communications management application mainly through acquisition of software houses with appropriate applications
- Founder (Gainey) always thought that Kana should add outbound e-mail management capability to its offering
- Kana wanted to be a leader offering a platform solution that could link online communications products to additional e-commerce applications
- Three pieces of infrastructure needed to build a billion-dollar e-commerce operation – content, to manage the transaction process, and manage the communication process
- Comprehensive channels options including Direct Salesforce, Key Business Partners and Kana Online (KOL)
- Investigation of Application Service Provider (ASP) channel option

Etic 2 view:
- Founders worked directly with early customers to development an inbound email management application
- Focus on rapid acquisition of software houses to build up online communications application capability
- Management combined efforts on acquisition and application development during IPO process
- Adaptation of strategy toward offering a complete e-commerce application suite (still with emphasis on online communication management)
- Constant acquisitions and mergers to build up critical mass and application development and service capabilities
- Channels Partners for extended sales/marketing and developer resources (particularly SI's)

1996-2000. Initial mental model:
- Initial business opportunity identified as setting up online storefronts for Sports Goods manufacturers
- Identified a different issue – complaints associated with managing emails
- Rapid Application Development directly with customers
- Rapid Expansion into Online Communications Management Through Acquisition of software houses with relevant applications
- Transformation to e-commerce provider through further acquisitions and mergers
- Establishment of Channels to service Enterprise Customers
- Exploration of Application Service Provider (ASP) model

2000-2001. Extended mental model:
- Full eCRM application provider focused on Enterprise Customers

Decisions-actions:
- Worked directly with customers to manage emails; Developed Kana V1 (Nov 1997), V2 (Mar 1998); V3 (Dec 1998 with SDK); Expanded online communications management capabilities through acquisitions of Connectify, BEI, NetDialog; IPO Sep 1999; V4 released (Oct 1999); V5 released (Dec 1999); Move to full e-commerce application provider through merger with Silknet (2000); Kana V6 eCRM suite released (mid-2000); Kana merges with Broadbase and releases Kana V7 (mid-2001)

Exhibit 7.6 Level V Analysis of Kana Communications Case

Etic 1 view:
- Email management opportunity found while seeking different opportunity (online shopfronts for sports goods manufacturers)
- Expansion to inbound and outbound email management through acquisitions
- Transformation to supplier of e-commerce application through acquisitions and mergers with funding from an IPO
- Channels options focused on expanding Kana Online (KOL) and developing an Application Service Provider (ASP) group

Emic 2 view:
- Importance of working with "Lead-Users" on learning, development and implementation of new applications
- Focus on transformation and repositioning from a small software house to a major software infrastructure provider
- Emphasis on quality, specialization and solving specific online issues
- Balancing between building applications Versus acquiring applications and skills
- Focus on special application development and porting efforts for major hardware and technology partners
- Roles of System Integrators (SI's) and Independent Software Vendors as extended sales & marketing and developer resources

Emic 1 view:
- Transition from accidental discovery of opportunity – managing emails – to new application
- Development of online communications management application mainly through acquisition of software houses with appropriate applications
- Founder (Gainey) always thought that Kana should add outbound e-mail management capability to its offering
- Kana wanted to be a leader offering a platform solution that could link online communications products to additional e-commerce applications
- Three pieces of infrastructure needed to build a billion-dollar e-commerce operation – content, to manage the transaction process, and manage the communication process
- Comprehensive channels options including Direct Salesforce, Key Business Partners and Kana Online (KOL)
- Investigation of Application Service Provider (ASP) channel option

Etic 3 view:
- Etic 2 views validated plus:
  - Decision-Making For Rapid Development and Release of Internet-Based Applications
  - Importance of System Integrators (SI's) and ISV's in marketing, referral and further application development and customization
- Broadbase effectively took over Kana but kept building up the Kana brand and applications
- Kana survived as a leading eCRM provider repackaging its applications into Kana iCare in mid-2001 and releasing Kana IQ V8 in Sep 2003

Etic 2 view:
- Founders worked directly with early customers to development an inbound email management application
- Focus on rapid acquisition of software houses to build up online communications application capability
- Management combined efforts on acquisition and application development during IPO process
- Adaptation of strategy toward offering a complete e-commerce application suite (still with emphasis on online communication management)
- Constant acquisitions and mergers to build up critical mass and application development and service capabilities
- Channels Partners for extended sales/marketing and developer resources (particularly SI's)

1996-2000. Initial mental model:
- Initial business opportunity identified as setting up online storefronts for Sports Goods manufacturers
- Identified a different issue – complaints associated with managing emails
- Rapid Application Development directly with customers
- Rapid Expansion into Online Communications Management Through Acquisition of software houses with relevant applications
- Transformation to e-commerce provider through further acquisitions and mergers
- Establishment of Channels to service Enterprise Customers
- Exploration of Application Service Provider (ASP) model

2000-2001. Extended mental model:
- Full eCRM application provider focused on Enterprise Customers

Decisions-actions:
- Worked directly with customers to manage emails; Developed Kana V1 (Nov 1997), V2 (Mar 1998); V3 (Dec 1998 with SDK); Expanded online communications management capabilities through acquisitions of Connectify, BEI, NetDialog; IPO Sep 1999; V4 released (Oct 1999); V5 released (Dec 1999); Move to full e-commerce application provider through merger with Silknet (2000); Kana V6 eCRM suite released (mid-2000); Kana merges with Broadbase and releases Kana V7 (mid-2001)

7.7 DSA Model

The original DSA model was developed from a thorough review of the Sarvary (2000) case study. Sections of the case describing Kana's products and development, plus a description of acquisitions that extended Kana's application suite were selected for DSA model development. The original DSA model is presented in Figure 7.2 and is representative of application development in 1999.

Kana software engineers worked closely with customers to identify features and improvements for new versions of the Kana product (see boxes 2 and 3).

Kana's selected customer list (see Sarvary 2000, Exhibit 3) numbered over 70 across more than 15 substantial industries. These customers were using the product for customer communication management in their e-commerce initiatives.

Kana also sought significant inputs from its partners. Kana developed a comprehensive channels structure to support its large customer base. All channels provided inputs to the engineers (see box 4), and it was decided to explore these inputs further through the development of a specific cognitive map for Kana's channels structure.

The key decision-point for application development of most versions of Kana's application suite was whether the features or enhancements could be developed "in-house" or through external acquisition.

External acquisition options addressed in the case included bringing in engineers with the skills to develop the enhancements or to directly acquire a software house with the relevant application (boxes 5 and 6).

Application development through substantial acquisition is unusual in the case-studies selected for this project – but was relevant during the boom in Internet stocks in the late 1990s. The boom in these and other technology stocks provided some software houses with the funds to access developer resources to support rapid application development.

Enhancements to Kana's application suite were allocated by specific modules, with new modules added when required (box 7). Applications from acquired software houses were mostly incorporated as additional modules in the Kana application suite.

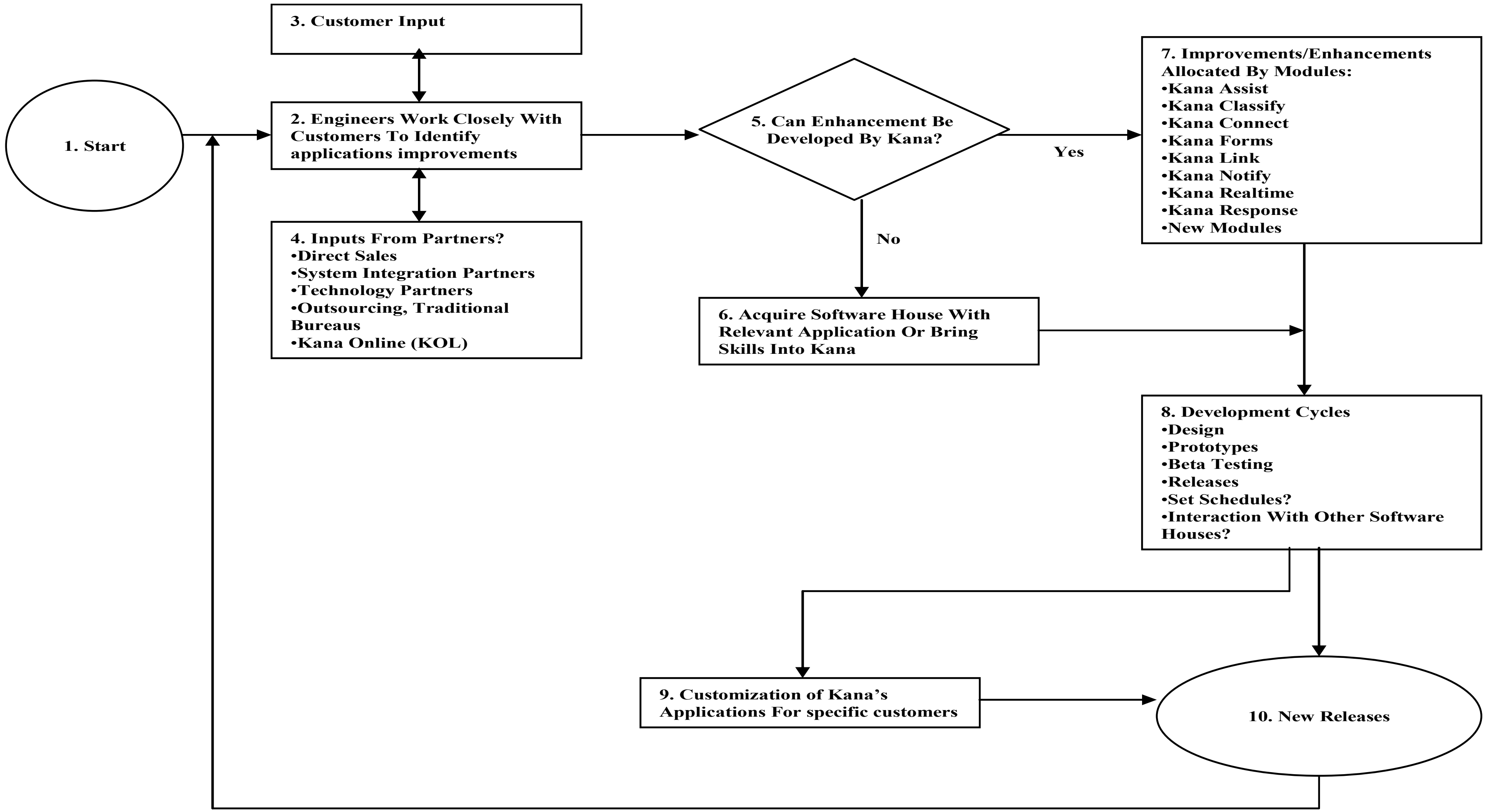


Figure 7.2 Summary DSA Model For Kana Communications Application Development (1995-1999)
Created from data in Sarvary (2000)

There is very limited description of the actual development of Kana's applications from a specific software-development cycle perspective. Apart from the first version of Kana with inbound e-mail management capability only, which was largely developed "in-house", significant blocks of design, prototyping and testing, would have focused on the integration of acquired applications into the Kana application suite (box 8).

Kana's application suite could be delivered directly to customers or System Integrators and/or technology partners, who in turn would customise the suite, or Kana's engineers could directly customise the suite for specific customers (box 9). New releases of the Kana application suite could be offered as standard or customised for specific industries and customers.

The actual development cycle from design to release would have varied significantly through the five years addressed in the case study. However, five versions of Kana were released in five years from 1996 to 2000, suggesting an average development cycle of twelve months.

7.7.1 Updated DSA Model

Two former Kana executives (Gainey & Wolfe 2002; Gainey 2002; Wolfe 2002) who were involved in the company's development and operations from 1996 to 2000 provided substantial additional details and insights into Kana's application development processes.

The DSA model for Kana's application development was substantially revised, mainly to reflect an increasing range of inputs to defining product strategy. The updated DSA model is presented in Figure 7.3.

Application development and design was not mainly the prerogative of the Engineering Group. Gainey & Wolfe (2002) emphasised several other key inputs for defining product/application strategy and design, so a box was created to reflect that these decisions were collective within Kana with additional key inputs from customers and partners.

Seven groups within Kana were identified as providing significant inputs to product strategy and design (boxes 3, 4, 5, 6, 7, 8, 9), reflecting an organisation structure akin to an established large software house of the early to mid 1990s.

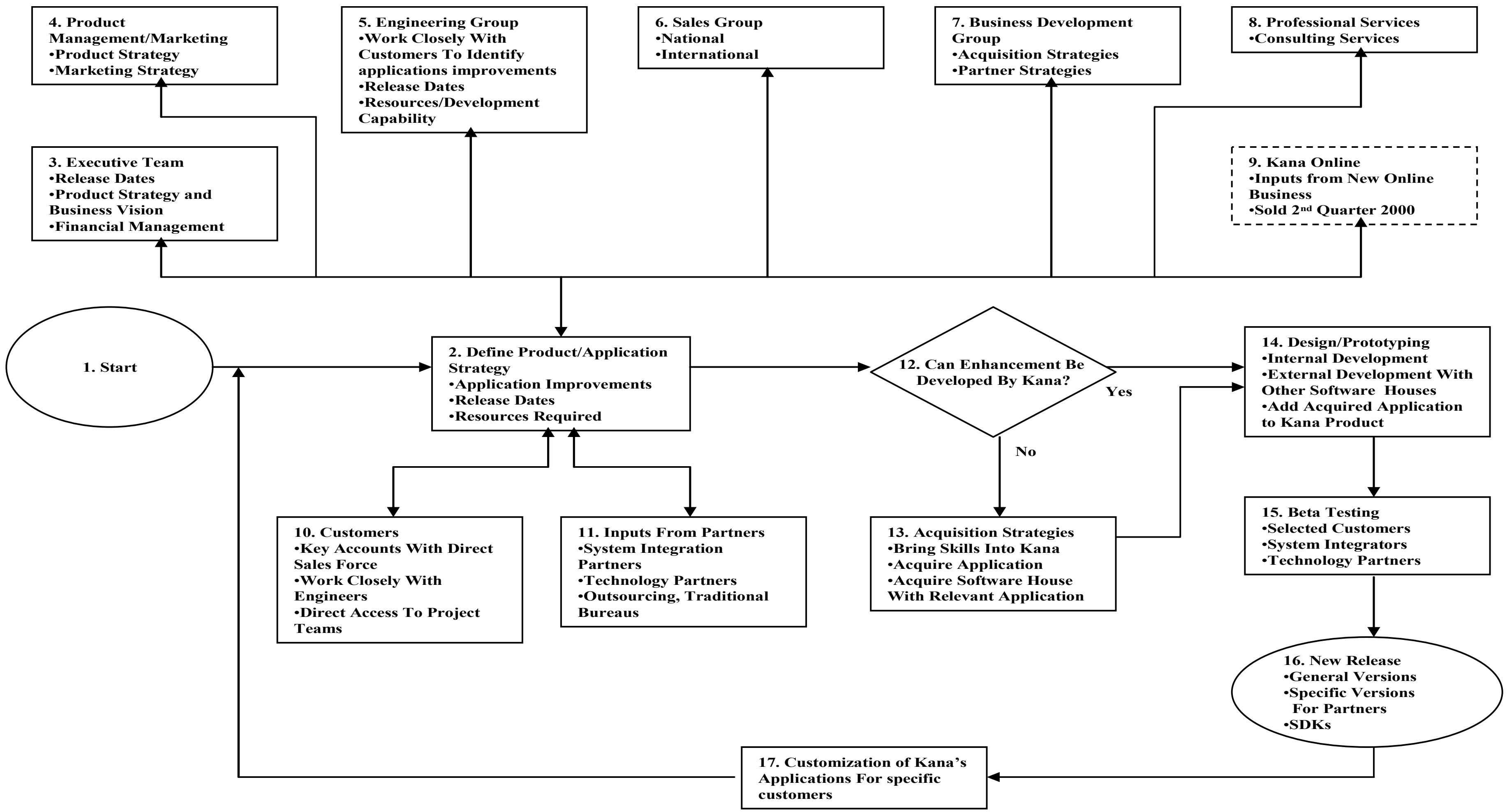


Figure 7.3 Updated Summary DSA Model For Kana Communications Application Development (1995-1999)
Created from data in Sarvary (2000); Gainey & Wolfe (2002)

The development teams are cross-functional, with inputs from groups within Kana including:

- Executive Team
- Product Management/Marketing
- Engineering Group
- Sales Group
- Business Development Group
- Professional Services
- Kana Online Group

The *Executive Team* (box 3) has direct input into product strategy and business vision. Managing the expectations of investors, analysts and customers through setting target release dates, and financial support to develop applications are vital inputs for product strategies.

Product Management/Marketing (box 4) provides significant competitive and marketing strategy inputs for enhancements, features and extending Kana's products into a comprehensive customer communications management suite.

The *Engineering Group* (box 5) works closely with key customers to identify applications improvements and enhancements, providing significant inputs on Kana's available resources and capabilities to develop these improvements. They also provide significant inputs to setting release dates.

The *Sales Group* (box 6) provides a bridge and input from key local, regional and international customers. There appears to be a significant "triangle" of interaction between key customers, the Sales Group and the Engineering Group providing inputs into product strategy.

The *Business Development Group* (box 7) is focused on acquisition and partner strategies – an important plank of product strategy for the rapid expansion of the application suite toward complete customer management and enterprise e-commerce systems.

Professional Services (box 8) provide services to key customers and work with Kana's partners. They provide inputs to product strategy and, as the applications become large and complex enterprise systems, their input is growing in significance.

The *Kana Online* division (box 9) provided significant inputs to product strategy through 1999-2000, but was sold in the second quarter of 2000.

Inputs from Customers (box 11) via the Engineering, Product Marketing and Sales Groups to product strategies were noted in previous sections. Although most customisation occurs after product releases, key customers may directly influence product strategy to incorporate customisation features and enhancements.

Kana relies heavily on partners (box 11) both to provide applications and skills and to provide consulting and installation services. A number of Kana releases were specifically developed for particular System Integrator/Technology Partner requirements, for example, Kana 7.5 was developed specifically for IBM. Development and contribution from Kana's channels partners is explored further through the cognitive map section (6.4.3).

The decision point regarding whether enhancements could be developed by Kana (box 12) remains in the updated DSA model. However, the choices from this decision point were altered to reflect additional acquisition options (either acquiring an application or acquiring the application software house – box 13), and to be less application-specific and focused more on actual development processes.

A Design/Prototyping box was created (box 14), indicating that Kana undertakes both internal and external development. For early versions of Kana, development was mainly internal. However, as the application suite expanded, development may be outsourced or applications may be brought in from ISVs, or an ISV may be acquired.

Kana has a comprehensive beta testing program (box 15) encompassing key customers and system integrators and technology partners – particularly where versions may be released for specific technology platforms and partner requirements.

New Releases (box 16) include general releases and "sub-releases" for specific technology platforms, technology partners and system integrators.

Important additions with new releases are System Developers Kits (SDKs). SDKs are new releases packaged with extra tools, applications and documentation to enable developers (Kana internal, or partners and customers) to quickly install and customise applications.

Customisation of Kana's applications (box 17) was moved from prior to New Releases to after the Release stage, although several customisation features in earlier versions were incorporated in later versions.

Development cycles for general releases appear to be around 12 months, with several "sub-releases", particularly in later versions.

Kana's application development processes could be described broadly as rapid waterfall (or linear) but acquired software houses might not have necessarily followed that approach.

Integration of completed acquired applications could be accommodated under a waterfall approach. It is not clear how much transition would be required for acquired developer teams to work to Kana's application development processes.

Overall, the revised DSA model reflects development cycles moving toward a large, complex and gradually maturing application suite – with the exception that there was still room for some rapid expansion of the suite through acquisition.

While "conventional" software houses may rely heavily on large internal development teams, or outsourcing managed by development teams, Kana retains a strong level of flexibility through using acquisition of skills and applications. Issues related to Kana's partners and acquisitions are further explored through relevant cognitive maps.

7.8 Events Chronology Map

Sarvary (2000) chronicled the creation and development of Kana from an idea in 1996 through to a large and significant e-commerce company – yet still growing aggressively to develop "critical mass" to be able to service a large and established customer base and channels system.

An events chronology map covering the period 1996 to February 2000 was created (see Figure 7.4), drawn from Sarvary (2000). Some additional symbols were employed for the map, including the flowchart symbol for a delay or problem and a six-pronged box for a solution to the delay or problem.

Mark Gainey originally thought about starting up an Internet storefront for sporting goods manufacturers, but identified a different and more pressing issue as manufacturers indicated that they could not handle large amounts of incoming e-mails (boxes 1 and 2).

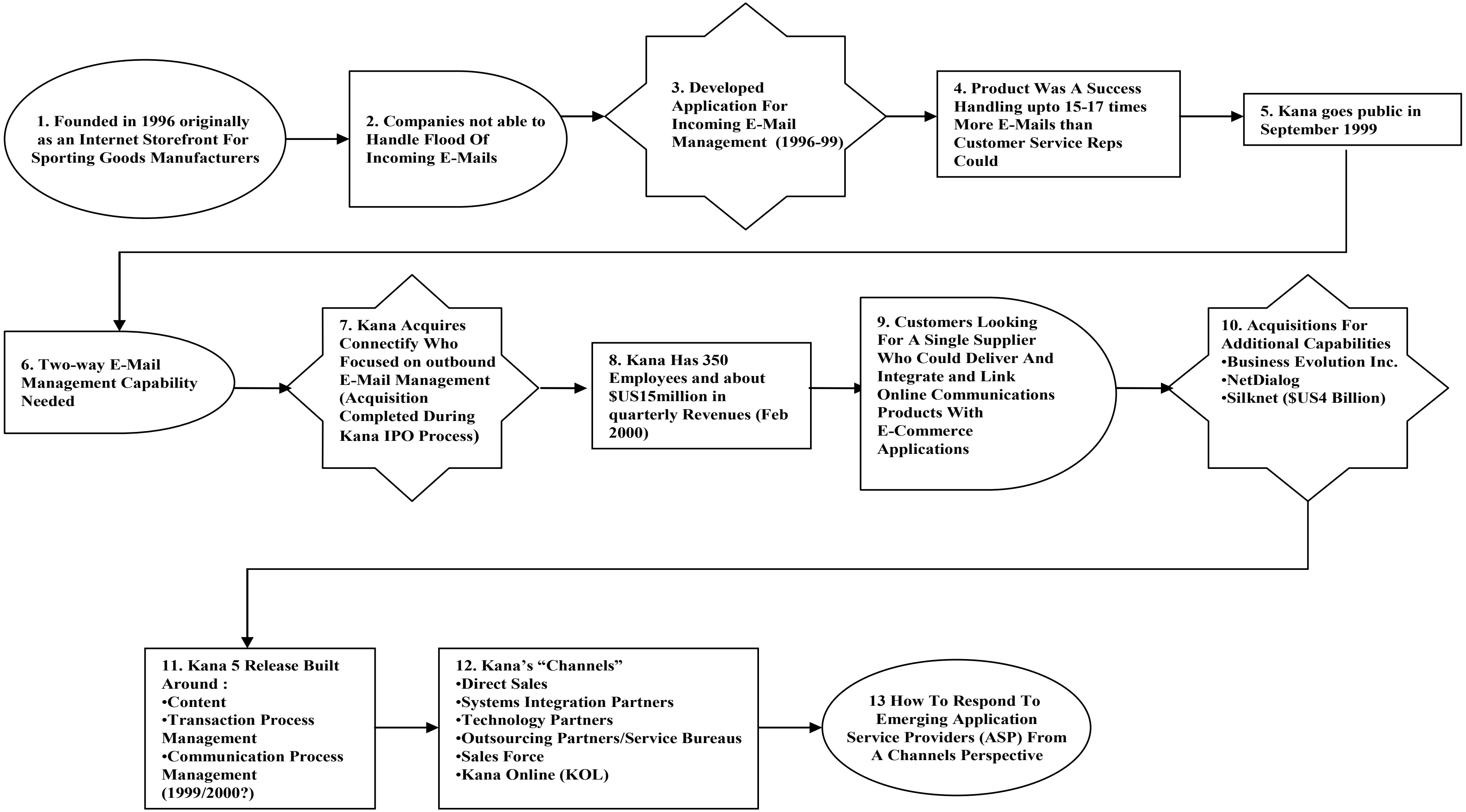


Figure 7.4 Events Chronology Map – Kana Communications
Created from data in Sarvary (2000)

Gainey saw the opportunity to develop an application to manage e-mail management, and worked with Michael Horvath to start Kana in 1996.

Kana's application was an immediate success and Kana grew rapidly as companies established online stores during the "dot.com" frenzy during 1996-99. In September 1999, near the peak of the Internet boom, Kana went public in a successful IPO that enabled the company to access several billions of dollars to pursue a rapid expansion strategy (boxes 3, 4 and 5).

Kana's application needed two-way capability, that is, to handle inbound and outbound e-mail communication. To address this requirement Kana acquired Connectify which had developed an application suite mainly focused on outbound e-mail management (boxes 2 and 7). Kana actually undertook this acquisition prior to the actual IPO date.

By February 2000, Kana had grown to 350 employees and around $US60 million in annual revenues, although revenues were doubling every quarter. Kana's stock price in February 2000 was $US 1,335.00 per share (as at 16 February 2000).

However, customers were moving toward integrating e-commerce applications – with a view to end-to-end integration of their information systems. Just to provide a comprehensive Online Communications application suite – not to provide to full "e" ERP suite – meant that Kana would have to quickly acquire skills and/or applications and to integrate them into Kana's current product (box 9).

Kana was able to use its proceeds from the IPO and its high stock price to acquire Business Evolution Inc. and Dialog in late 1999, and a $US4 billion acquisition of Silknet in February 2000 (box 10).

Kana released Version 5 in early 2000 with enhancements for Content, Transaction Process Management and Communication Process Management – its fifth release in five years (box 11).

By early 2000, Kana's channels structure was comprehensive, with substantial direct and indirect paths. Kana offered direct channels through its direct sales force and Kana Online (KOL) groups, but relied heavily on its System Integration and Technology partners to refer and integrate the Kana product in enterprise e-commerce applications (box 12).

A significant issue for Kana in February 2000 was the evaluation of an option to create an Application Service Provider (ASP) group where customers could have their online communications management applications hosted on Kana's internal computer systems (box 13).

7.7.1 Updated Events Chronology Map

The original events chronology map was updated through the interview stage, incorporating significant increases in details and events relating to Kana's development of its products and the company from 1995 to mid-2001. The revised events chronology map is presented in Figure 7.5.

Gainey's exploration of business opportunities related to starting an Internet storefront service for sporting goods manufacturers. His discovered in 1995-96 that companies were not able to handle a rising flood of incoming e-mails (boxes 1 and 2).

Gainey co-founded Kana Communications Inc. with Michael Horvath in July 1996 (box 3). They commenced development on a new application for incoming e-mail management in mid-1996 and released the first version of the application in November 1997 (box 4).

Kana's application success in handling 15 to 17 times more e-mails than a customer service representative was established through both Kana 1 and 2 applications during 1998 (box 5).

Kana was officially launched as company in March 1998. Kana 2 was released in conjunction with the company launch (box 6).

Kana 3 represented a significant extension in application development with web interface capability (that is, managing e-mail communications directly from websites) and a System Developer Kit (SDK).

The SDK enabled System Integrators, Technology Partners, Enterprise Development Teams and ISVs to quickly commence development using Kana 3. Kana was released in December 1998 (box 7).

Kana's main challenge in 1998-999 was to develop two-way e-mail management capability. Kana commenced an IPO process in mid-1999 but delayed the offering in order to acquire Connectify which had developed applications with outbound e-mail capability (boxes 8 and 9).

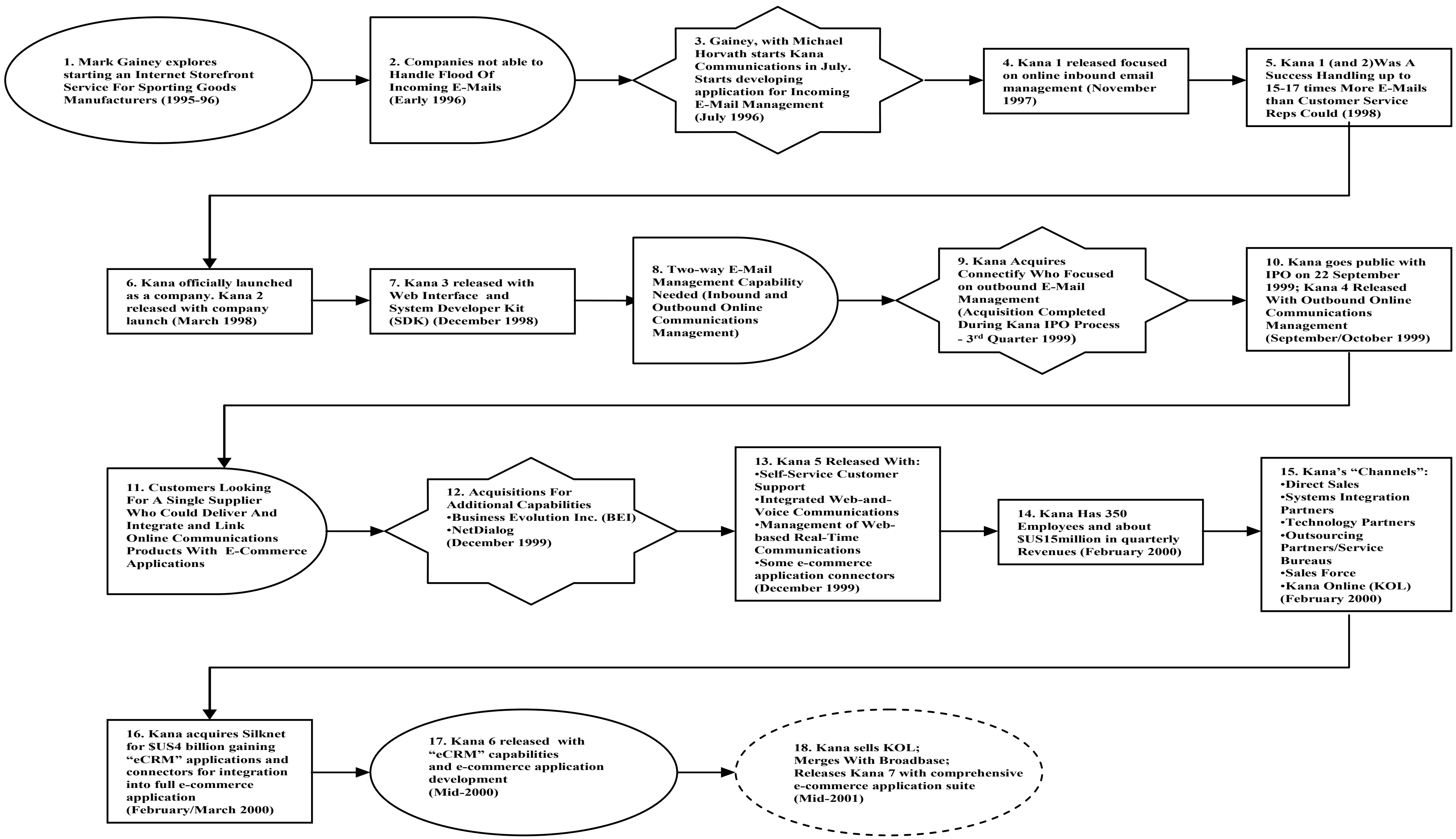


Figure 7.5 Updated Events Chronology Map – Kana Communications – Created from data in Sarvary (2000); Gainey & Wolfe (2002); Gainey (2002), Wolfe (2002)

Kana went to its IPO with Connectify integrated into the company and a new version of Kana with Connectify's outbound call management application incorporated into its application suite.

Kana's IPO commenced trading on 22 September 1999, and launched Kana 4 at the same time as the IPO, although the product was released through September – October 1999 (box 10).

Customer issues with integration and linkage of online Communications products with e-commerce applications remained the same as in the original events chronology map, but acquisitions to address this issue were separated to reflect that BEI and NetDialog were acquired in December 1999, while Silknet was acquired in February 2000 (boxes 11, 12 and 16).

Kana 5, when released in December 1999 (box 13), incorporated applications from the BEI and NetDialog acquisitions, including significant enhancements for integrated Web and voice communications, and Web-based real-time communications management.

Information on Kana's revenues and employee numbers, and its channels structure was repositioned to be sequenced in the February 2000 timeframe after Kana 5 was released (boxes 14 and 15).

Kana's acquisition of Silknet in February – March 2000 netted "e-CRM" applications and connections/interfaces for integration into full e-commerce application suites (box 16). These enhancements, plus applications brought in through the BEI and NetDialog acquisitions, were incorporated in Kana 6, released in mid-2000 (box 17)

Although strictly outside the scope of the case study a "footnote" (box/circle 18) is provided, indicating that in mid-2001 Kana sold KOL and merged with Broadbase. Broadbase offered a full enterprise "e-CRM" application suite plus integrate e-commerce applications.

The revised events chronology map highlights the rapid emergence of a significant enterprise Internet-based applications provider using acquisitions as a major driver of application development and company growth.

7.9 Cognitive Maps

Cognitive maps created for the Kana case study were designed to complement DSA models and events chronology maps through additional exploration of selected decision-making issues and contexts.

All cognitive maps were created using the following procedures:

1. Selected extracts from Sarvary's (2000) case study were extracted to text files
2. One text file version of each extract was processed through the TACT package to produce a complete word-frequency list
3. Word frequencies were reduced through one or two iterations to ranked lists of words with frequencies ranging from over two to three words
4. A second version of the text file was input to NVivo
5. Ranked words were input to NVivo and coded as nodes in association with the text file
6. The overall extract was deconstructed into selected phases and sentences and coded directly as nodes through the InVivo tool within NVivo
7. The nodes associated with each extract were input to the Model Explorer and mapped using links incorporating codes of association plus addition node "items"
8. Several iterations of dynamic mapping were undertaken with the final cognitive map exported from NVivo to PowerPoint
9. Each PowerPoint version of the cognitive map was reformatted into a more readable and project-consistent format
10. The revised PowerPoint cognitive maps were presented to interviewees and validated or modified
11. The updated PowerPoint cognitive maps incorporated changes after the interview process.

Three sections were extracted from Sarvary (2000) and subjected to further text analysis and cognitive mapping to complement analysis presented in the DSA model and events chronology map.

The three cognitive maps created for the Kana case study addressed the following themes:

7.9.1 Early Development (1996-99)
7.9.2 Acquisitions (1999-2000)
7.9.3 Channels (2000)

The first two cognitive maps are sequential snapshots in time, highlighting a strong switch toward acquisition as a key driver for application development.

The third cognitive map explores Kana's channels structure as of February 2000. The map provides additional insights into Kana's channels partners' substantial contributions to gaining enterprise customers' acceptance of Kana's application suite.

7.9.1 Cognitive Map 1 – Early Development (1996-99)

Development details for the first cognitive map include:

- Extract section, size and exhibit:
  - "From founding to market leadership" section, Sarvary (2000, p.2); ½ page; Exhibit 7.7
- TACT word frequency lists and comments on key words:
  - First & final pass: 21 words with frequency > 2
  - Final ranked word list: Table 7.1
  - The highest ranking words were related to *e-mail, customer* and/or *communication management*.
- NVivo ranked word, sentence and phrase node count = 124
- Association of concepts coding used: +, -. +/-
- NVivo PowerPoint cognitive map: Figure 7.6
- Reformatted PowerPoint cognitive map: Figure 7.7

The NVivo cognitive map captures the strategic thinking and action associated with creating Kana and developing early versions of Kana products. The cognitive map highlights Mark Gainey's logic in identifying a huge opportunity for the management of incoming e-mail management in large companies.

Gainey worked with Michael Horvath to found the company in 1996. Through 1996 and 1997, the new company sought funding and hired engineers, and spent most of their time and resources developing an incoming e-mail management application.

Exhibit 7.7 Text Extract: Kana – Early Development (1996-99)

Kana Communications From founding to market leadership

Kana Communications was a software company started by young entrepreneur Mark Gainey in 1996. Being an avid rower on Harvard's crew and a regular skier at Lake Tahoe, Gainey was originally thinking to start an Internet storefront-service for sporting goods manufacturers. When he talked to manufacturers however, he was struck by the big brand-name sites' common complaint of not being able to handle the flood of incoming e-mails. Not only they did not use the valuable marketing information hidden in consumer messages, "most companies weren't even reading the e-mails they were getting".

Looking around, Gainey realized that there was no application out there for incoming e-mail management. He saw a huge opportunity and with Harvard friend, Michael Horvath decided to start Kana Communications. They named the company after Gainey's dog, Kana. Over the next three years, the founders hired a few engineers, raised $16million through three round of financing and spent their time developing software. The resulting product was a success, increasing customers' e-mail response capabilities 10-20 fold. At e-Bay for example – one of Kana's early customers – a customer service representative could respond to 500 e-mails per day, some 15-17 times more messages than a telephone representative could handle. Based on this success, Kana went public in September 1999 in one of Wall Street's most successful IPO's.

Originally, Kana was specialised in the management of inbound e-mail only. Gainey found this to be the major point of pain for customers in the early years of the Internet. He realized the huge potential value that companies could get from direct communications with their customers. But communication goes both ways and Mark always thought that Kana should add outbound e-mail management capability to its offering. Later (right in the middle of the IPO process), Kana acquired Connectify a startup whose efforts were mostly concentrated on outbound e-mail management. Connectify's product allowed clients to easily run customised direct (e-mail) campaigns to their customers by merging pre-defined messages with the firm's existing customer database. This way the e-mails were customised and the right e-mail went to the right customer. Connectify was the logical way to extend Kana. Although its acquisition somewhat delayed the IPO process, the new CEO Michael McCloskey convinced the management and investors to go ahead with the merger.

Source: Sarvary (2000, p. 2).

Table 7.1 Kana Communications – Early Development (1996-99) – Ranked Word List

Anchor Concepts	Frequency
email	11
kana	10
customer	8
was	7
management	5
Gainey	5
company	4
communication	4
for	4
with	4
success	3
start	3
right	3
messages	3
could	3
Connectify	3
by	3
not	3
that	3
this	3
were	3

Source: data analysis performed using TACT (1997)

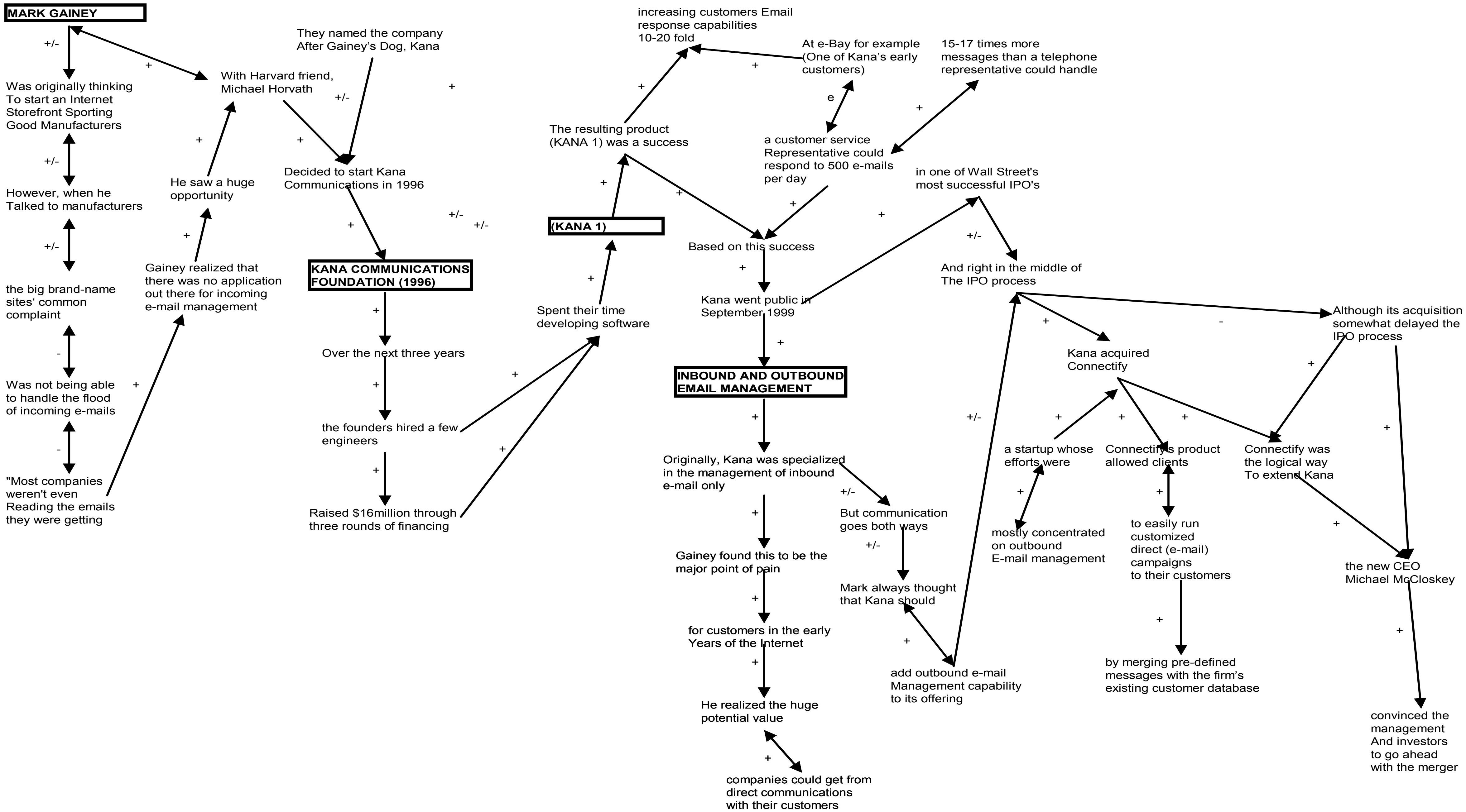


Figure 7.6 First Cognitive Map (NVivo Model) – Kana Communications – Early Development (1996-99)
Created from data in extract (Exhibit 7.7)

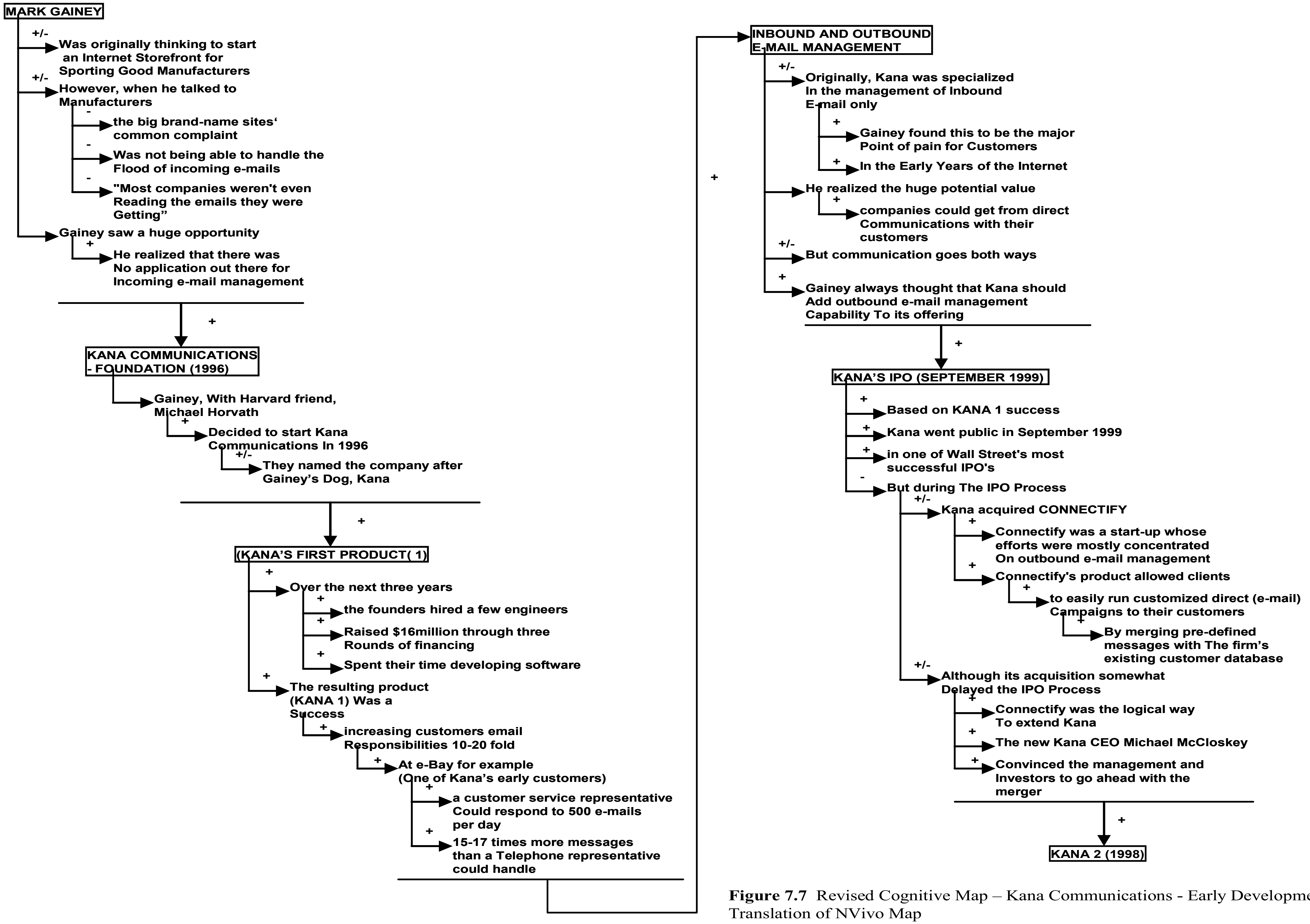


Figure 7.7 Revised Cognitive Map – Kana Communications - Early Development (1996-99)
Translation of NVivo Map
Note: This revision was shown to participants in emic 2 stage for their (Wolfe and Gainey's) interpretation.

Kana 1, released in late 1997, was an immediate success, with the application increasing processing of e-mail inquiries 10 to 20 times greater than a call centre representative could handle. E-Bay found that its typical customer service representative could respond to 500 e-mails per day – but Kana 1 could handle 15 to 17 times more than the customer service representative, that is, 7,500-8,500 per day.

Gainey realized that, although Kana was successful in addressing a key customer requirement with those customers who were early adopters of the Internet for conducting online business activities, the application needed to be extended to manage all aspects of online customer communications management.

However, the window of opportunity for developing a leading-edge product for this area would be very short. In 1997 the Internet was growing rapidly both in terms of users and the numbers of supporting applications.

It would not be long before either established Enterprise Resource Planning (ERP) vendors or other new and emerging software houses would close in on this opportunity. The fastest way to extend Kana 1 one further logical step to handle outgoing e-mail management was to acquire a complementary application.

Kana's success in 1998 with Kana 1 prompted the company to prepare quickly for a public offering in 1999 – a process that would set it up with very large amounts of capital to be used later to undertake mergers and acquisitions to build a comprehensive online customer management application suite.

Despite the very high managerial focus required to set up the IPO, in early 1999, Kana found a software house (Connectify) with strong outbound e-mail applications.

Kana's new CEO, Michael McCloskey convinced management to acquire Connectify before Kana went to IPO. This priority delayed the IPO by a few months, but meant that the company was able to go with IPO with the promise of an upcoming version with both inbound and outbound e-mail management capability.

7.9.1.1 Updated Cognitive Map 1 - Early Development (1996-99)

The cognitive map was revised through direct editing in PowerPoint with inputs from interviews with Wolfe (2002) and Gainey (2002). Significant insights into Kana application development, application release dates and sequence of early acquisitions, were gained from the interviews and incorporated into the revised cognitive map (see Figure 7.8).

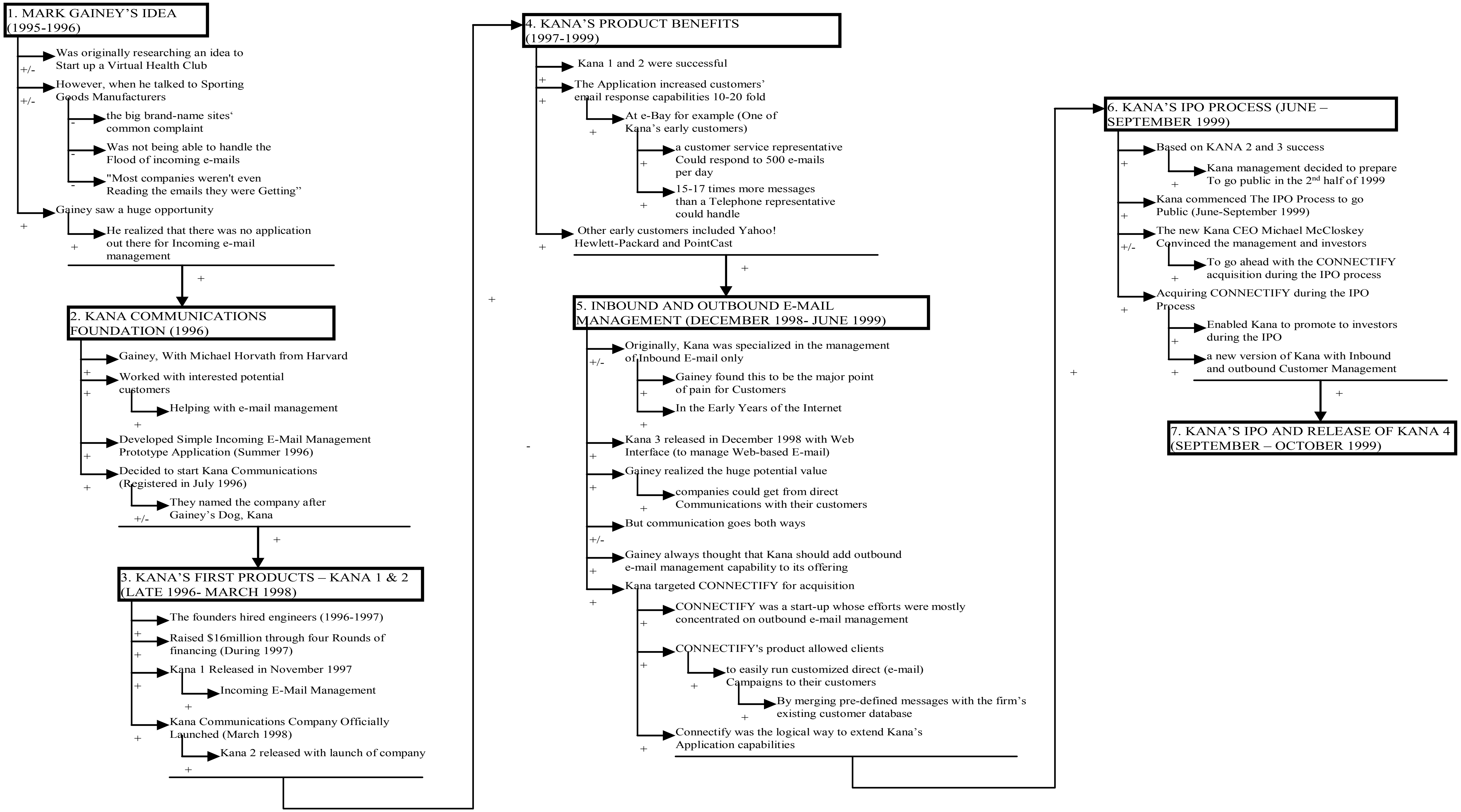


Figure 7.8 Second Revised Cognitive Map – Kana Communications - Early Development (1996-99)
Note: This map represents etic 3 interpretation.

The first section of the cognitive map was renamed to reflect that Mark Gainey was looking for business ideas (box 1). Gainey initially researched an idea to start up a virtual health club (or at least a health club with an online website) in 1995.

However, when he talked to sporting goods manufacturers (who might supply the health club) he found that a major problem for them was managing incoming e-mails.

Mark Gainey teamed up with Michael Horvath from Harvard, working with interested potential customers, helping out with e-mail management (see box 2). They actually developed a simple prototype for managing incoming e-mail management during summer 1996. Kana Communications was registered as a company in July 1996.

Significant additions and revisions covering application development from late 1996 through to late 1998 were incorporated into the revised cognitive map. A new section covering the development and release of Kana 1 and 2 was created (see box 3).

The founders built up the company in 1997 through hiring software engineers and raising over $US16 million of funding through four rounds (not three as in the original cognitive map).

Kana 1 was released in November 1997. Kana started producing revenue and was officially launched as a company in March 1998, with an upgraded version, Kana 2, also released at that time.

Product benefits emerging from Kana 1 and 2 were reclassified into a new section (box 4). Early customers in addition to E-Bay included Hewlett-Packard and PointCast.

The "inbound and outbound e-mail management" section in the original cognitive map was reworked to incorporate additional insights into the acquisition of Connectify which was targeted several months before Kana's IPO process. The revised cognitive map includes the logic for targeting Connectify directly as a result of requiring an outbound e-mail management solution (see box 5).

Kana's IPO Process section was revised to drop the notion that the Connectify acquisition delayed the public stock offering (see box 6). This section now highlights that the CEO Michael McCloskey convinced Kana's management to go ahead with acquiring Connectify while preparing for the IPO.

Kana was able to announce its acquisition of Connectify sufficiently early in the IPO process to be able to promote the promise of a new version of Kana's product, including enhancements from Connectify. These features were incorporated into the next version of Kana's product, released about a month after the IPO.

Kana's IPO was floated on NASDAQ on 22 September 1999. In October 1999, Kana 4 was released in October 1999 with inbound and outbound online customer communications management capabilities.

Kana's application development cycles from 1997-99 were very tight with four versions released in just two years. Kana 1 was developed in under 18 months (July 1996 – November 1997), while Kana 2 was released just four months later (March 1998). The application development cycle for both Kana 3 and 4 was around nine months (versions released in December 1998 and September 1999 respectively). The first three versions of Kana's product were developed internally, while Kana 4 incorporated functionality brought in from Connectify.

7.9.2 Cognitive Map 2 – Acquisitions (1999-2000)

Development details for the second cognitive map include:

- Extract section, size and exhibit:
    - "From founding to market leadership" section, Sarvary (2000, pp. 2-3); ½ page; Exhibit 7.8
- TACT word frequency lists and comments on key words:
    - First & final pass: 20 words with frequency > 2
    - Final ranked word list: Table 7.2
    - Apart from the references to Kana, the word list themes associated with *growing customer* management *applications* through *acquisition* (words in italics were ranked highly in the ranking list).
- NVivo ranked word, sentence and phrase node count = 70
- Association of concepts coding used: +, -. +/-, =, e
- NVivo PowerPoint cognitive map: Figure 7.9
- Reformatted PowerPoint cognitive map: Figure 7.10

The final cognitive map presented provides a cognitive description of Kana's competition and acquisitions during 1999-2000.

By mid-1999 Kana had gained market leadership in a segment defined as online customer communications. The Connectify acquisition enabled Kana to offer a comprehensive inbound and outbound online customer e-mail management system.

Exhibit 7.8 Text Extract: Kana – Acquisitions (2000)

124

By February 2000 Kana Communications counted 350 employees and generated about $15 million revenues per quarter, which roughly doubled each quarter (see Exhibit 1 for Kana's Income Statement). Part of this growth was fueled by aggressive acquisitions. In the second half of 1999, although Kana was clearly the market leader in the online customer communications business, it was by far the only player. Other companies like eGain, NetDialog and a dozen others have developed applications to address part of the online communications process. Many of these applications overlapped with Kana's products but some of them offered additional capabilities such as web-based communication for example.

Kana Communications 501-003

In this environment, customers were increasingly looking for a single supplier who could deliver and integrate all these applications. Furthermore, there was an increasing need among e-commerce customers for a platform solution that could link online communications products to additional ecommerce applications. CEO, Michael McCloskey felt that the market was ready for consolidation. He also thought that there were important first mover advantages to exploit before large software giants like Siebel, who were not presently on the Internet, wake up: "Kana wants to be the leader in this space. To do that, we have got to grow very quickly" - he said. 1

Armed with the highest market capitalization in the industry, Kana could complement its core inbound and outbound e-mail management software with additional capabilities. In the second half of 1999, on the same day, Kana acquired Business Evolution Inc. (BEI) and NetDialog adding new functionalities such as web- and voice-based communications and self-service customer support to its offerings. In early February 2000 Kana announced the acquisition of Silknet in a deal valued at $4 billion. The Silknet acquisition was an important step in the company's history. Besides providing the company with a platform for e-commerce applications it was also a signal to the market that Kana is willing to take the lead and become the dominant provider of e-commerce application software. The acquisitions were also an efficient way to grow. They doubled Kana's customer base as well as the number of employees, bringing hard to find new talent to the fast growing company.

Source: Sarvary (2000, pp. 2-3).

Table 7.2 Kana Communications – Acquisitions (1999-2000) – Ranked Word List

Anchor Concepts	Frequency
kana	11
communication	7
was	7
applications	5
customer	5
acquisition	4
company	4
ecommerce	4
grow	4
market	4
that	4
were	4
with	4
additional	3
also	3
lead	3
online	3
software	3
could	3
this	3

Source: data analysis performed using TACT (1997)

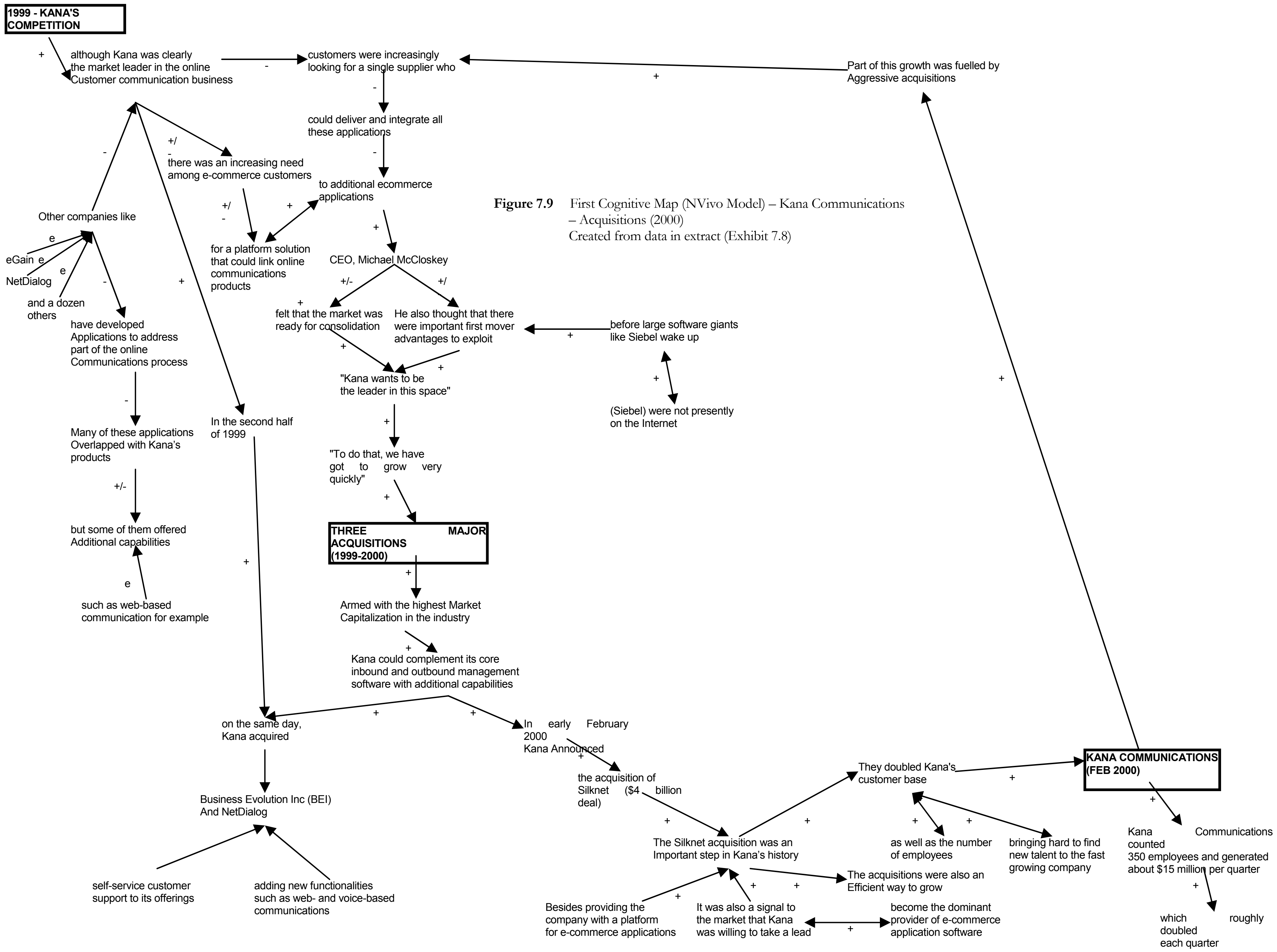


Figure 7.9 First Cognitive Map (NVivo Model) – Kana Communications – Acquisitions (2000)
Created from data in extract (Exhibit 7.8)

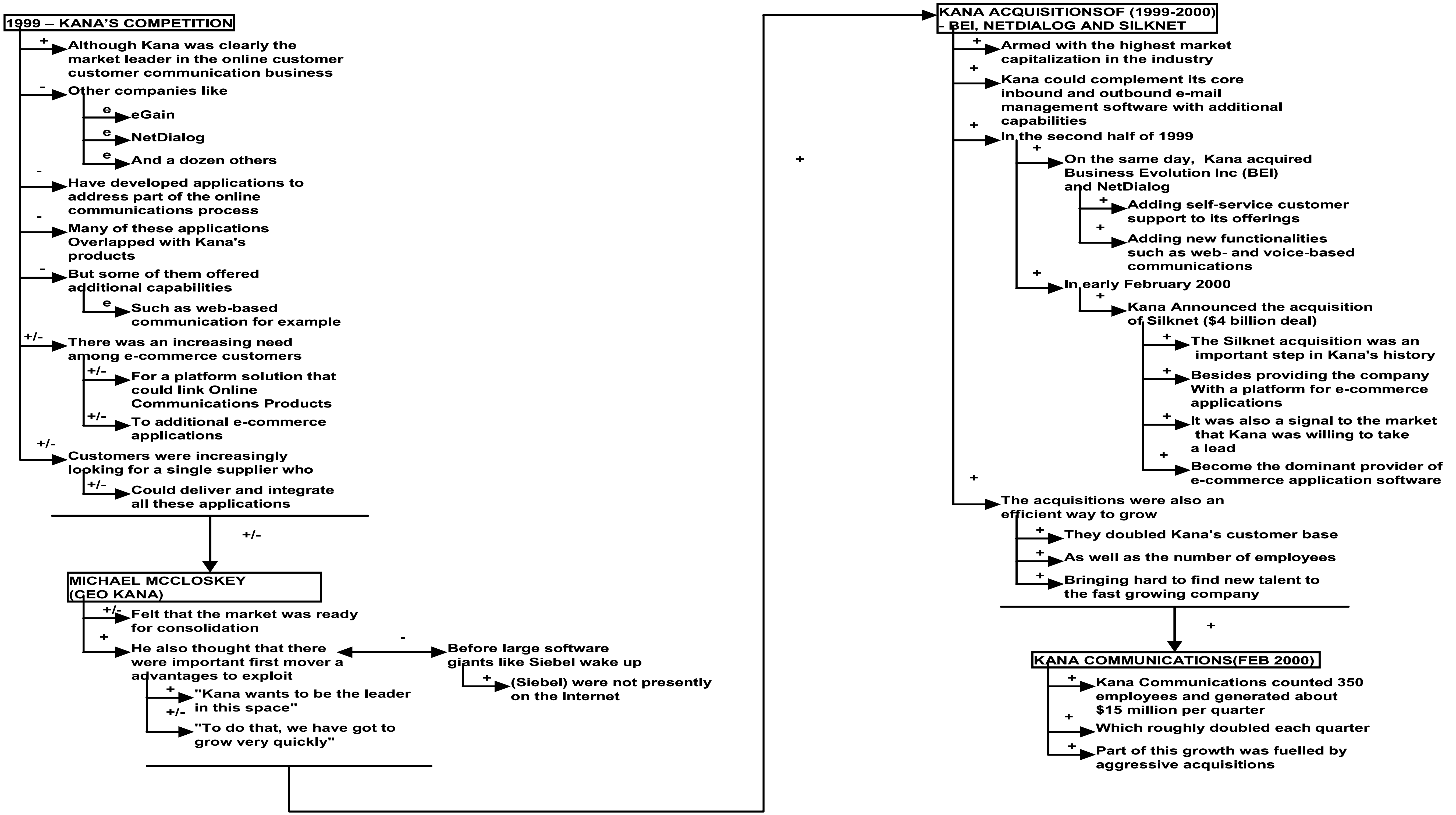


Figure 7.10 Revised Cognitive Map – Kana Communications - Acquisitions (2000)
Translation of NVivo Map
Note: This revision was shown to participants in emic 2 stage for their (Wolfe and Gainey's) interpretation.

There were over a dozen companies developing applications addressing parts of online communications processes and management. Sarvary (2000) identifies eGain and NetDialog as companies effectively competing with Kana, but with additional capabilities including web-based communication.

E-mail was a separate service on the Internet from the multimedia World-Wide Web service, and was managed separately from it. However, as users accessed the Web, pressure increased for options to create and send e-mail within web-browser applications.

Wed-based e-mail systems were released in April 1996 from Juno and USA.Net and in July 1996 by Hotmail (for an account of development of early web-based e-mail applications see Faleiro, Porter, Rastogi, Stokes, Venator & Tucci 2001). Hotmail had originally pursued an idea to link relational databases through the Web (see Net Dynamics Case in this project), and was acquired by Microsoft in 1998.

Customer Relationship Management (CRM) systems emerged in the mid-1990s as a logical evolution of direct marketing/database marketing systems developed during the 1980s and early 1990s. A new generation of CRM vendors focused on the management of customer information from online business activity in the late 1990s, with the exception of the online customer communications.

Integration of front-office or “customer-facing” applications (mainly in the province of CRM providers) to “back-office” applications (a stronghold of ERP) vendors into full enterprise e-commerce suites was “a slow-burning fuse” through the late 1990s – but emerged as a critical development issue in 1999-2000. Extension of CRM applications to include online customer communication management was positioned by Kana as “e-CRM”.

Kana’s new CEO, Michael McCloskey, believed that market consolidation was imminent but Kana had a key advantage over several major CRM players who were not yet developing for the Internet. McCloskey believed that there was a very short window of perhaps six to 12 months for Kana to transform its product offerings toward a comprehensive “e-CRM” suite.

Kana’s successful IPO provided it with a high market capitalization (highly variable but between $US1 and 10 billion in late 1999) enabling rapid and substantial acquisitions in a very short timeframe. These acquisitions brought in new customers and new employees – especially new developers with valuable skills.

The BEI and NetDialog acquisitions added important capabilities such as self-service customer support and web-and-voice-based communications.

The Silknet deal was a much greater challenge, as the two companies had similar customer numbers and revenues. Officially the deal was announced as a merger, although the new company would trade under the Kana Communications name. Silknet brought "e-CRM" applications that would complement Kana's online customer communication management applications.

At the time of the Kana/Silknet merger announcement, Kana's revenues were doubling each quarter, with final quarter 1999 revenues at $US15 million, with 350 employees. Even so, would this company size and momentum be sufficient for Kana to grow into a major e-CRM player in the face of a full frontal assault from large ERP vendors with deep pockets?

7.9.2.1 Updated Cognitive Map 2 – Competition and Acquisitions (1999-2000)

Subsequent to interviews with Gainey (2002) and Wolfe (2002), the cognitive map was revised for title, structure, and significant new insights. The revised cognitive map with the extended title of "Competition and Acquisitions" is presented in Figure 7.11.

The section on Kana's competition (see box 1) was revised to focus competition in the third quarter of 1999 (that is, around the time of Kana's IPO), with more distinct categories and identified existing and potential competitors. Classification of competition was:

- Direct competition with overlapping applications
- Direct competition with additional capabilities
- Call centre companies that had developed their own customer management applications, and that could either sell them or host customers using these applications
- CRM vendors that were developing online customer communication modules
- Vendors moving toward offering fully integrated e-commerce application suite
- ERP vendors looking to combine back-office and front-office applications (dealt with in detail in Section 4 on Consolidation)

Kana was very interested in acquiring competitors with additional capabilities that could complement their own skills and products. Call centre companies were a conundrum as they could be classified as customers and/or competitors, although the interviewee consensus was toward treating them as complementary.

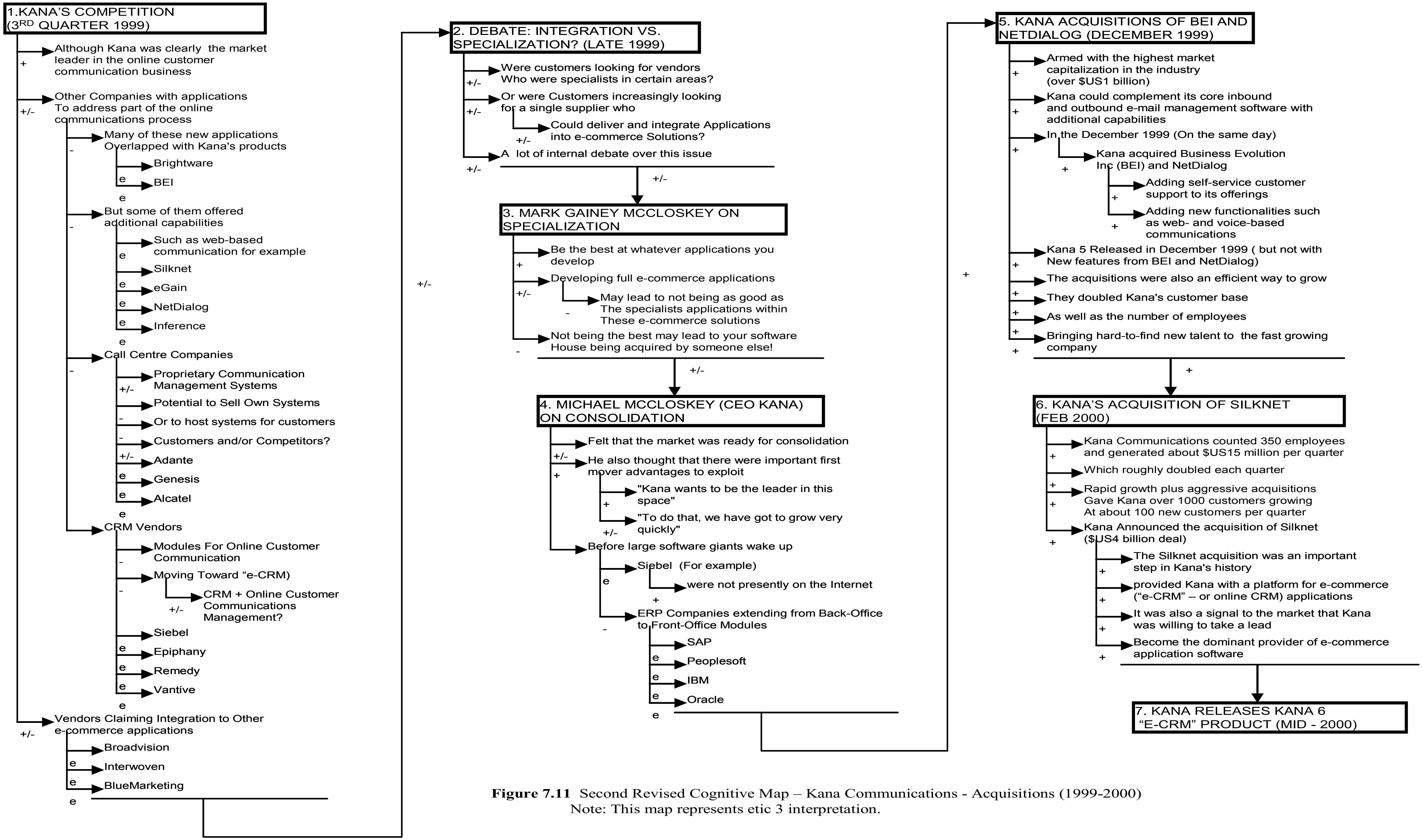


Figure 7.11 Second Revised Cognitive Map – Kana Communications - Acquisitions (1999-2000)
Note: This map represents etic 3 interpretation.

The CRM competitors were of particular strategic interest as there seemed to be a perception that evolution from CRM to e-CRM meant contemporary CRM plus online customer communications, that is, a “CRM vendor + Kana’s products”. Should Kana then move toward at least integration to the level of an “e-CRM”?

Those vendors moving toward offering fully integrated e-commerce applications were also of strategic interest because they effectively raised the bar for “integration” to go far beyond “e-CRM”.

In fact, there were substantial debates over not only how far Kana should pursue an integration strategy, but also about how specialised it should be with its products (see box 2). Although the interviewees suggested that the debate was polarising, the different perspectives are encapsulated by Gainey (box 3) and McCloskey (box 4).

Gainey’s perspective was to be the best at whatever applications you developed and was concerned about trading off excellence for broad functionality. McCloskey believed that existing competitors in Kana’s competitive space would be confronted with consolidation among themselves.

In addition to such consolidation Kana could expect an onslaught from established ERP vendors, intent on integrating their back-office solutions with CRM (and e-CRM) front-office solutions.

There was a window for Kana to consolidate its competition through acquisitions, while ERP vendors and some other large software houses were transitioning their applications to run on the Internet.

The sections of Kana’s acquisitions were reworked to separate the Silknet acquisition to reflect an accurate chronology of events. This reworking also reflects Kana’s development, initially to develop additional web-based functionality, then to becoming a leading e-CRM vendor.

Estimates of Kana’s actual capitalisation were difficult to ascertain, as stocks were escalating through the last of quarter of 1999 through to March 2000. However, according to Gainey (2002), at the time of the BEI and NetDialog acquisitions, Kana’s capitalisation would have been at least $US1 billion.

Kana was able to acquire BEI and NetDialog at the same time, and to release Kana 5, in December 1999, incorporating functionality brought in through these acquisitions. Kana was proficient in not only targeting strategic acquisitions, but planning early incorporation and implementation of acquired functionality in their next product releases.

Kana's acquisition of Silknet deserves a separate section (see box 6). Silknet itself had grown rapidly through several acquisitions and could bring to the table e-commerce integration capability. A merger between Kana and Silknet could enable the new company to become a dominant provider of e-commerce applications.

The merger was executed in the second quarter of 2000, and the newly merged company was able to launch Kana 6 in mid-2000 with substantial additional e-commerce capability (box 7).

7.9.3 Cognitive Map 3 – Channels (2000)

Development details for the final cognitive map include:

- Extract section, size and exhibit:
  - "Kana's Distribution System" section incorporating "Business Partners" and "Kana Online" subsections, Sarvary (2000, pp. 6-7); one page; Exhibit 7.9
- TACT word frequency lists and comments on key words:
  - First pass: 28 words with frequency > 2
  - Second & final pass: 14 words with frequency > 2
  - Final ranked word list: Table 7.2
  - Apart from the word "Kana", the words "partner", "KOL" and "customer" feature in the list, highlighting the themes of Kana's channels partners and description of KOL.
- NVivo ranked word, sentence and phrase node count = 98
- Association of concepts coding used: +, -. +/-, =, e
- NVivo PowerPoint cognitive map: Figure 7.12
- Reformatted PowerPoint cognitive map: Figure 7.13

The map highlights Kana's channels of distribution with emphasis on business partners, plus further insights into Kana Online (KOL).

Exhibit 7.9 Text Extract: Kana – Channels (2000)

Business Partners

Direct sales was not the only way Kana commercialised its application. The company also relied on business partners to generate demand and help in fulfillment. The partner base was often called "the channel" and included three types of partners. The first type of partners were systems integration partners. These would usually be strategy consultants such as Andersen Consulting or KPMG that could potentially recommend Kana to their clients. The key benefit for Kana was based on their capability to provide access to customers.

The second type of partner consisted of technology partners including any e-solution provider who Kana could integrate with. Usually, the partnership included co-marketing arrangements or investment on both sides to make integration easier. This group would also include technology companies with voice solutions, who recently considered moving to the Internet.

Finally, the third-party category of partners consisted of outsourcing partners or traditional service bureaus (that is, 1-800 services or call-centers). Usually, they represented large contracts. They were charged enterprise software prices (roughly 50% higher than list price) and were also charged an additional $5k-10k/customer/year. In some respect these partners competed with Kana, especially with Kana Online. For this reason, Kana was careful to choose outsourcing partners who would offer Kana as a complementary product in a total solution to the customer. For this reason call centers were also called value-added resellers (VARs).

2 All price and cost information has been disguised to respect company privacy.
128

Kana Communications 501-003

Kana Online

Kana also "distributed" its product through Kana Online (KOL). KOL was a service targeted to customers who did not want to purchase and manage the Information Technology (IT) infrastructure needed to run the Kana applications. Instead, they relied on Kana to host the Kana 5 modules for a monthly fee determined based on the size of the application. Originally, Kana introduced this service to match competitive offers. Without any resources allocated to its marketing, KOL grew at a phenomenal rate. The majority of KOL customers were small Internet startups who were cash constrained and did not want to make large investments in infrastructure. Recently however, and increasingly so, some large organisations also subscribed to KOL. By the beginning of 2000, the total number of KOL users grew to over 100, representing about 25% of the customer base. Viewing this success, management decided to put more resources to support the sales of KOL and appointed a Vice President, Timothy Campbell to develop the business. Since Tim's appointment KOL grew even faster signing up about 2 customers per day, a trend that was expected to continue.

KOL customers paid on an ongoing basis. The average price for the KOL service was roughly $10,000 per month. Again, this charge would vary by the size of the purchased application measured by the number of messages handled. For Kana Response, for instance, the basic monthly fee was $5,000, which included 10,000 messages (50 cents per message). Each additional message would cost 25 cents. Kana Connect was priced between 1-2 cents a message with a minimum monthly fee of $5,000 including 100,000 free messages.

Extracted from Sarvary (2000, 6-7).

Table 7.3 Kana Communications – Channels (2000) – Ranked Word List

Anchor Concepts	Frequency
kana	18
partner	12
KOL	10
customer	8
include	6
message	5
price	5
service	5
who	5
with	5
would	5
application	4
base	4
month	4

Source: data analysis performed using TACT (1997)

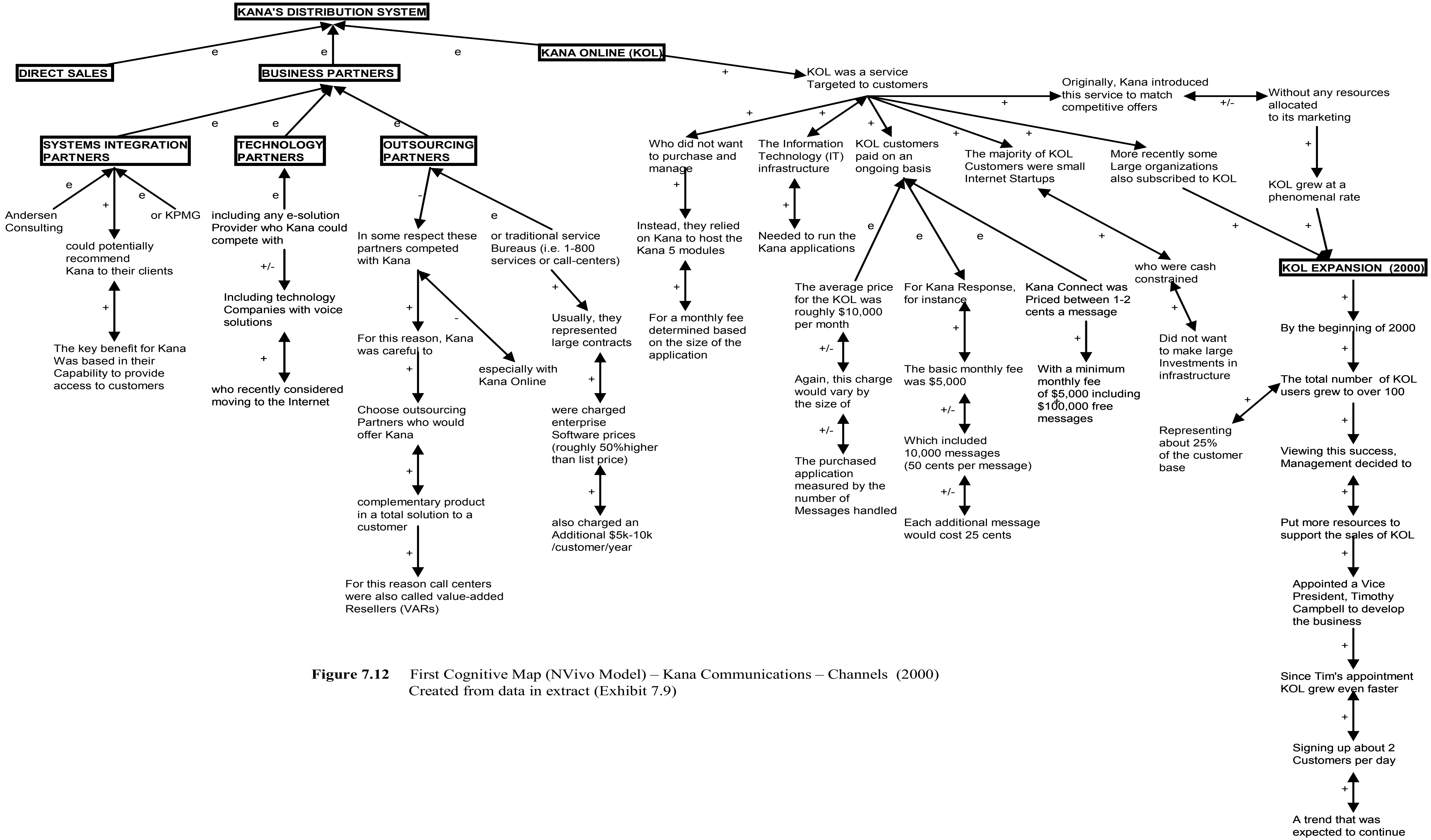


Figure 7.12 First Cognitive Map (NVivo Model) – Kana Communications – Channels (2000)
Created from data in extract (Exhibit 7.9)

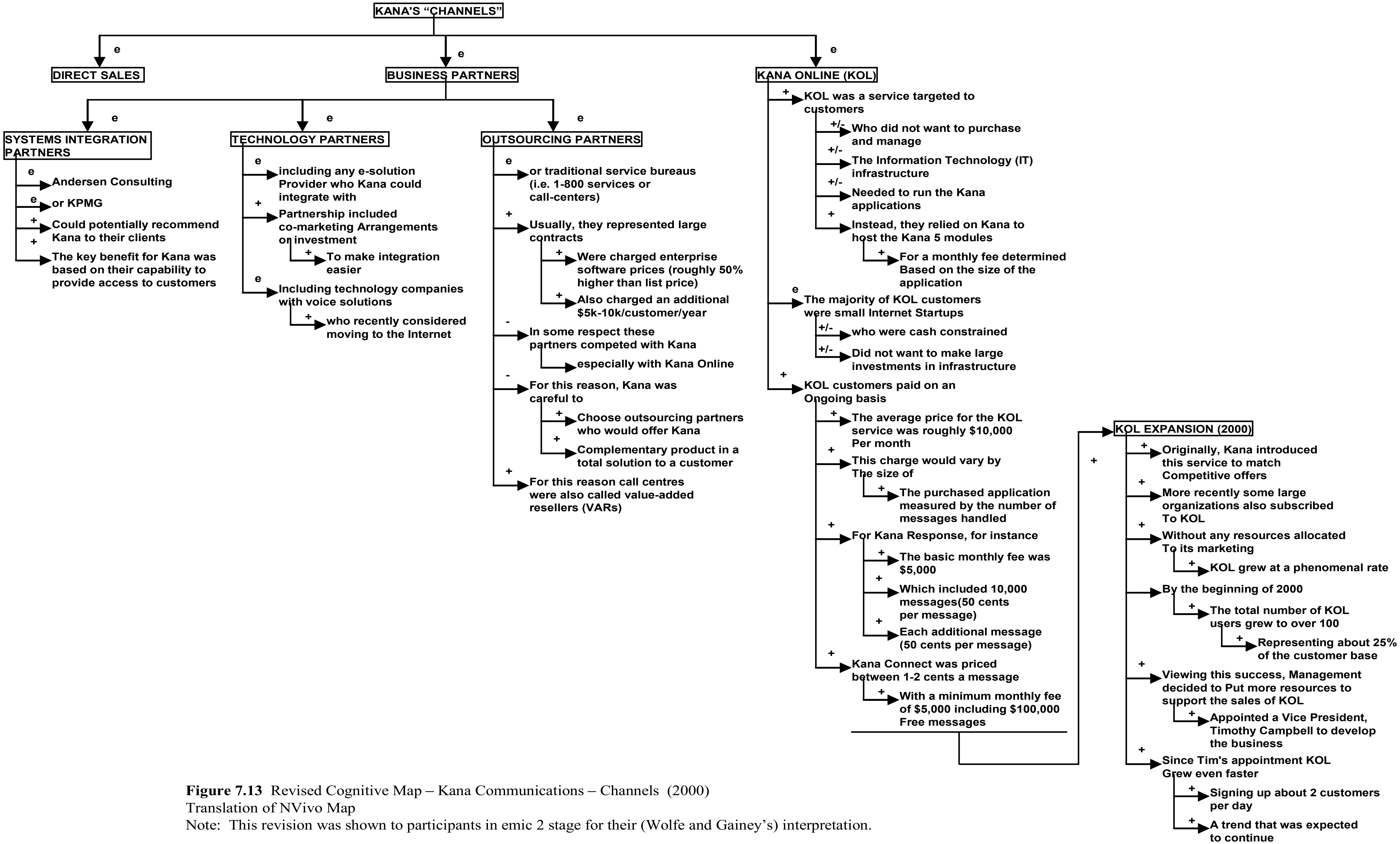


Figure 7.13 Revised Cognitive Map – Kana Communications – Channels (2000)
Translation of NVivo Map
Note: This revision was shown to participants in emic 2 stage for their (Wolfe and Gainey's) interpretation.

Kana's three Channels as of early 2000 are identified as:

- Direct Sales Force
- Business Partners
- Kana Online (KOL)
- Direct Sales force

Sarvary (2000) discusses Kana's Direct Sales force in a separate section (2000, pp. 5-6). However, this section was excluded from the extract for analysis, partly because it would have increased the extract to 2½ pages in length, and a decision to focus more specifically on Kana's Business Partners and KOL at the expense of the Direct Sales force.

- Business Partners

Kana's Business Partners are categorised into three groups:

- System Integration Partners
- Technology Partners
- Outsourcing Partners

System Integration Partners work with enterprises to design, develop and implement information systems. Typically, they provide consulting services to design business organisation structures and practices, selection of hardware, software and associated technologies to support these designs, and project teams to develop the enterprise applications. Some Systems Integrators may manage portions of or fully outsourced enterprise information systems.

As such, System Integrators play a critical role in recommending and implementing new Internet-based applications. System Integrators may be a "virtual" extension of a small or start-up software house both in marketing and customisation/customer development activities.

Kana identified Andersen Consulting and KPMG as key System Integrators for their product. Companies such as Andersen Consulting and KPMG could access potential customers for Kana's product more directly than Kana, so it was vital for Kana to development strong relationships, communication programs, and technology assistance to them.

Kana was actively seeking *Technology Partners* with applications and technologies that Kana could integrate with. Independent Software Vendors (ISVs) require special services such as System Developers Kits, APIs, and technical support to encourage them to use Kana's product in their application suites.

Kana assisted some of these ISVs with co-marketing arrangements. A driving factor for establishing and extending these ISV arrangements was to offer solutions that complemented Kana's products – and extended Kana's capabilities.

However, Kana was developing its capabilities so quickly that there could be a fine line between establishing a technology partnership based on complementarity and then acquiring a technology company to extend Kana's capabilities.

Outsourcing Partners appear to have a classification peculiar to Kana. Hardware vendors would traditionally supply their products or services to companies for incorporation into their hardware products. The company incorporating the products or services is classified as a Value-Added Reseller (VAR). A parallel analogy for software is an ISV.

However, the classification used here applies to call centres or service bureaus managing outsourced customer communications activities. Kana's products are supplied to call centres or service bureaus, which in turn incorporate them into their systems.

These VARs represented large revenue-earners for Kana, but there was some concern within Kana regarding the status of these VARs as to whether they were customers and/or competitors. Kana was careful to strategically select VARs with complementary solutions to its products.

- Kana Online (KOL)

Kana established Kana Online (KOL) in 1999 to provide a hosted or outsourced online customer communications management service. KOL initially focused on customers who did want to purchase and manage the IT infrastructure plus Kana's product but still needed it for their business. The majority of these customers were small new Internet companies who were cash-constrained and preferred to outsource this activity.

Kana charged on average $US10,000 per month, depending on the number of messages handled, and modules selected by the customer. Kana Response incurred a basic monthly fee of $US5,000 including 10,000 messages.

Over and above the 10,000 messages there was an additional charge of US50 cents per message. Kana Connect incurred a minimum monthly fee of $US5,000 including $US100,000 of free messages.

Kana established KOL to match competitors – it is assumed that "competitors" here are software houses offering competitive products and not necessarily outsourcing partners.

During late 1999 through early 2000 KOL grew rapidly and changed direction, with larger enterprises subscribing to KOL. KOL was growing without marketing resources, probably on the wave of new Internet-related businesses starting up at that time.

Kana's management team decided to put resources into KOL, appointing a Vice President to establish a business division for KOL. KOL continued to grow rapidly after this appointment.

Although outside the extract selected for analysis, Sarvary (2000) drove discussion on KOL further toward the Kana management team, considering whether KOL should reposition itself as an enterprise hosting service – or in "new economy" terminology, an Application Server Provider (ASP),

This strategic option would have major implications for the future development of Kana's channels and its competitive position. Certainly in the first quarter of 2000, an ASP approach looked to be an attractive strategic option for the Kana management.

7.9.3.1 Updated Cognitive Map 3 – Channels (2000)

Interviews with Wolfe (2002) and Gainey (2002) provided substantial extra insights into Kana's business partners and developments with KOL. These insights are incorporated into the updated cognitive map presented in Figure 7.14.

The timeframe for the updated cognitive map is extended from 2000 to mid-2001 (see box 1). This timeframe enables the incorporation of additional strategic channels activities, such as releases of specific Kana products for technology partners, and a full history of the KOL business.

The actual channels of distribution, that is, Direct Sales, Business Partners and Kana Online (KOL) with Business Partner classifications (Systems Integration Partners (SIs), Technology Partners (including ISVs) and Outsourcing Partners (including VARs) are unchanged in the updated cognitive map (see boxes 2 to 7). However, there are significant additions to each of the Business Partner categories.

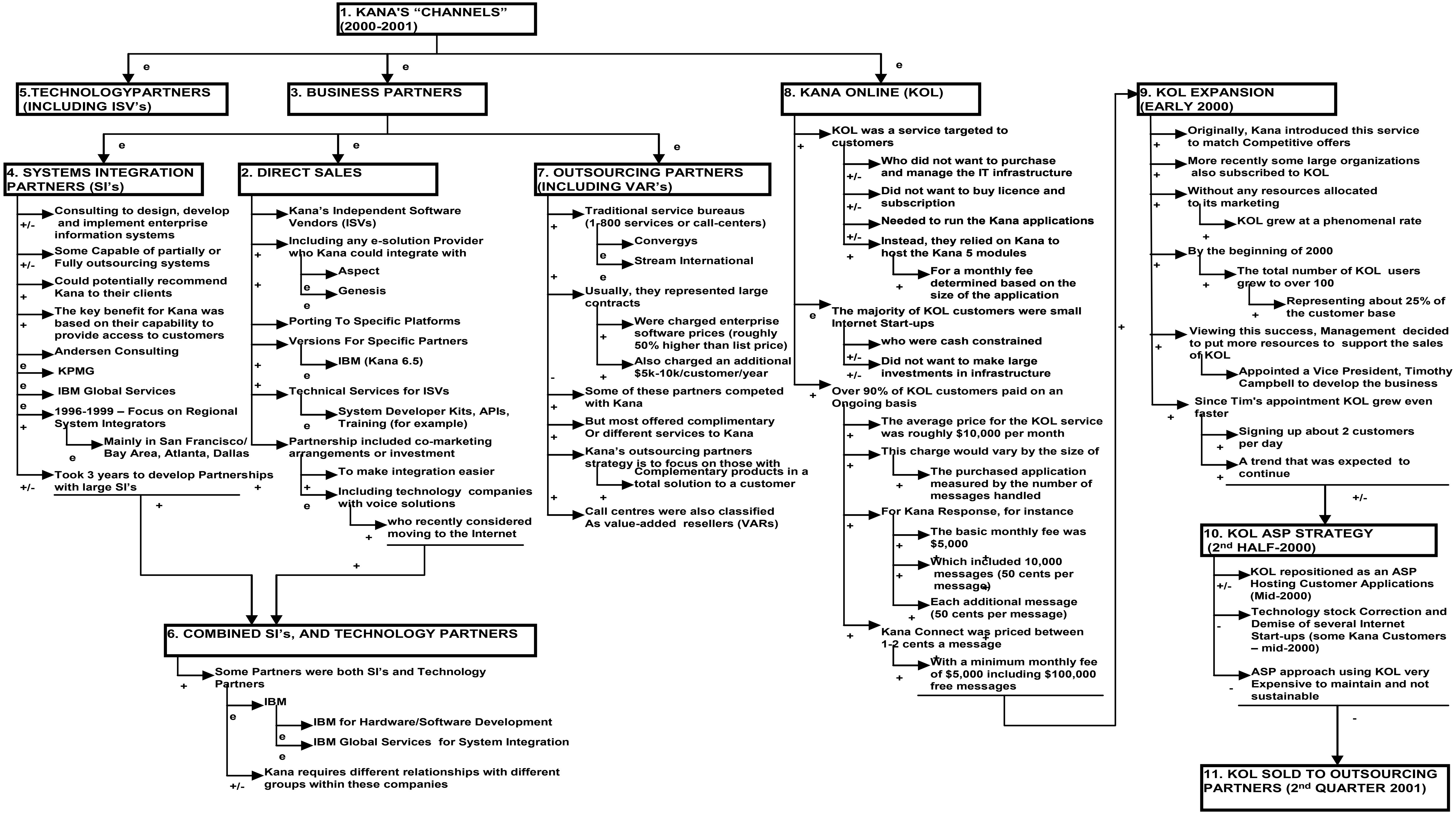


Figure 7.14 Second Revised Cognitive Map – Kana Communications - Channels (2000)
Note: This map represents etic 3 interpretation.

The System Integration Partners (SIs) category (see box 4) is updated to reflect their substantial capabilities in developing and implementing enterprise solutions, and their critical role as decision-makers and influencers for products and services to be selected for such solutions. IBM Global Services is identified as an additional key SI partner in this category.

Comments on Kana's development of SI relationships are included in the updated cognitive map. Kana initially focused on regional SIs in the San Francisco/Bay area, Atlanta and Dallas. Developing relationships with larger SIs was a much larger and longer commitment – typically taking up to three years to formalise and to produce meaningful results for Kana.

The Technology Partners' category (box 5) was updated to reflect Kana's ISV relationships and services provided to ISVs. Kana ported to specific operating systems and platforms to enable ISVs to use Kana's products in their solutions.

In some cases, Kana releases sub-versions of its products to run on specific partner platforms (for example, Kana 7.5 for IBM). These porting activities consumed considerable development resources and required very close interaction with the partner.

Kana also provides System Developer Kits, APIs and training to ISVs to enable them to start development quickly using Kana's products.

Insights into co-marketing arrangements and investment from the original map remained unchanged in the updated cognitive map.

A new section addressing Combined SI and Technology Partnerships was created (see box 6) to reflect Kana's multiple relationships with some companies that carried out both SI activities and as ISVs developing hardware and software.

IBM is an important example in which IBM Corporation develops hardware and software and IBM Global Services is a System Integrator. Kana's challenge is to develop partnerships with several groups within IBM to ensure that Kana's products are used and recommended by these groups.

The Outsourcing Partners category (see box 7) is largely unchanged in the updated map, except that the consensus from interviews is that the VARs were not generally competing, but were more complementary toward Kana.

The Kana Online (KOL) category (box 8) is unchanged in the updated map, except for one comment emphasising that customers of KOL did not want to buy Kana licences and subscriptions, preferring a hosted solution.

KOL's expansion in early 2000 (box 9) was confirmed through the interview stage and no revisions were required for this section. However, two new sections were added addressing Kana's implementation of an ASP strategy (box 10) and ultimate the sale of KOL to outsourcing partners in the second quarter of 2001 (box 11).

Kana's ASP strategy for KOL was implemented in mid-2000. However, at the same time the Technology Stock correction and demise of several Internet start-ups severely impacted KOL's revenues and customer base.

In fact by late 2000, the ASP strategy was not only expensive to maintain but it was determined that the strategy was unsustainable. Subsequently Kana exited this business through the sale of KOL to outsourcing partners in the second quarter of 2001.

The updated cognitive map extends insights from Kana's DSA model and events chronology map into Channels relationships that are vital for Kana's development into a strong vendor of enterprise "e-CRM" and eventually e-commerce solutions.

7.10 Extension Questions

Interviewees for the Kana case study were asked extension questions covering Dominant Logic, Shared Vision, Leverage Points, and Strategic Marketing Principles.

7.10.1 Dominant Logic

The interviewees identified different executives dominating decision-making, or providing dominant logic at different times during 1996-2001.

Mark Gainey and Michael Horvath provided dominant logic for the genesis of the company in 1996-97. Gainey and his executive team with Mike Wolfe were dominant in relation to development of Kana 1, 2, 3 and 4 (1997-99).

However, from his arrival as CEO in summer of 1999 through to stepping down in January 2000, Mike McCloskey provided most of the dominant logic for the acquisitions for 1999 into early 2000.

Jay Wood replaced McCloskey in January 2000 and continued for a short time with the same dominant logic. However, after the merger of Silknet, a combined executive team provided a more collective dominant logic for decision-making.

Although outside the strict timeframe of this Case study, Kana's merger with Broadbase in mid-2001 resulted in a distinct change of direction in decision-making toward incoming Broadbase executives.

7.10.2 Shared Vision

Development and ownership of shared vision also varied through time. According to Gainey (2002) and Wolfe (2002), from 1996 to 1999, Kana's executive team, with inputs from throughout the company, would establish a shared vision and take that out to customers and partners.

There was some variance as to whether Kana actually developed a collectively shared vision with these parties or if told the customers about its products and services and how to implement them!

There was a high level of developing a shared vision for product development and implementation with major business partners. Targeted System Integrators had to "buy into" Kana's shared vision for its products to recommend them.

Those ISVs with potentially high revenues for applications using Kana's products warranted extra attention as Kana worked very closely with some of them to develop specific versions of their products for them.

Kana seems to have developed strong cross-company shared vision with strategic business partners. However, Gainey (2000) noted that such sharing of vision was quite opportunistic. Those ISVs targeted by Kana received a lot of shared vision from Kana, but non-strategic ISVs received very little communication or shared vision from Kana.

7.10.3 Leverage Points

Several leverage points were identified during the interviews. These are outlined in chronological order.

- 7.10.3.1 Emergence of Idea to handle Incoming E-Mails (1996)
- 7.10.3.2 Early Rounds of Funding (1997)
- 7.10.3.3 Kana's Official Launch as a Company (March 1998)
- 7.10.3.4 Emergence of Need for New E-Commerce Applications – Such as Online Customer Communication Management (1998)
- 7.10.3.5 Kana's IPO and Release of Kana 4 (September-October 1999)
- 7.10.3.6 Kana's Merger with Silknet (February 2000)
- 7.10.3.7 Kana's Merger with Broadbase (Mid- 2001)

7.10.3.1 Emergence of Idea to handle Incoming E-Mails (1996)

Initially Mark Gainey was exploring a completely different idea to set up a virtual health club. However, Gainey's discussions with several sporting goods manufacturers unearthed a new challenge – and opportunity – to manage incoming e-mail communication.

This realisation was a leverage point for establishing a new company, then a new market (customer communication management), a new set of customer management systems (e-CRM), and eventually the creation of a central component of e-commerce applications.

7.10.3.2 Early Rounds of Funding (1997)

In order to develop the first version of their application, Gainey and his fledgling executive team had to actively seek out funding in 1997. They were able to successfully raise over $US16 million over four rounds of funding in 1997. This funding enabled Kana to create a strong software development team and to quickly develop Kana 1, released in November 1997.

7.10.3.3 Kana's Official Launch as a Company (March 1998)

Up until March 1998, Kana was really in an incubation stage, in which Kana's staff worked with companies interested in them managing their incoming e-mail.

Kana developed its first application working directly with these customers in 1997, but did not start generating revenue until March 1998. Kana was launched publicly in March 1998, right at a point where interest in Internet-based companies and Internet-based applications were at frenzy levels.

Kana was able to generate revenues and gain sufficient public visibility early enough to grow rapidly, complete a highly successful IPO, and to complete a number of acquisitions – all before the Internet and technology stocks collapsed in mid-2000.

7.10.3.4 Emergence of Need for New E-Commerce Applications – Such as Online Customer Communication Management (1998)

Gainey (2002) indicated that while Kana was growing in 1997-98 an explosion in interest on Internet-based business activity or e-commerce directed a strong focus on related "killer applications".

Interest in online customer communications management applications was so strong that the phones at Kana were sort of ringing of the hook during the holiday seasons (Summer 1998 and Christmas 1998). Kana was in the right place at the right time with a key application for e-commerce activity.

7.10.3.5 Kana's IPO and Release of Kana 4 (September-October 1999)

Kana's IPO on 22 September 1999 was one of the most successful software house launches in history. So successful, in fact that it provided the company with at least $US1 billion to undertake acquisitions and to quickly build up Kana's e-CRM and e-commerce application capability.

The IPO enabled Kana to execute a very large $US4 billion merger with Silknet in February 2000.

7.10.3.6 Kana's Merger with Silknet (February 2000)

Kana's merger with Silknet fundamentally changed the shape philosophy of the company. Kana shifted Kana away from being a specialised start-up software house to a conglomerate with a large installed base seeking to develop a comprehensive set of e-CRM (and eventually e-commerce) solutions. By that time the founders had moved on to set up new software houses, and a different type of management was required to run a larger company.

7.10.3.7 Kana's Merger with Broadbase (Mid- 2001)

The newly merged company faced the management of a greatly increased installed base – but also had to confront the technology stock correction and the demise of many of its customers in late 2000 and early 2001.

Consolidation and restructuring plus a continuing vision of creating enterprise e-commerce solutions shifted Kana toward a merger with Broadbase in mid-2001.

Kana was not the dominant partner in this merger. Although the new company traded as Kana Communication, much of the dominant logic was now emanating from the incoming Broadbase executives.

7.10.4 Strategic Marketing Issues

Seven interconnected strategic marketing issues emerged from the interview stage. These issues include:

- 7.10.4.1 Working with "Lead-Users" on Learning, Development and Implementation of New Applications
- 7.10.4.2 Defining Potential Development and Use of New and Emerging Applications
- 7.10.4.3 Transformation and Repositioning from a Small Software House to a Major Software Infrastructure Provider
- 7.10.4.4 Emphasis on Quality, Specialisation and Solving Specific Online Issues

7.10.4.5 Building Applications versus Acquiring Applications and Skills

7.10.4.6 Vertical versus Horizontal Marketing Issues

7.10.4.7 Roles of System Integrators (SIs) and Independent Software Vendors

7.10.4.1 Working with "Lead-Users" on Learning, Development and Implementation of New Applications

During 1996-98, according to Gainey (2002), Kana spent a lot of time educating interested companies about effectively managing e-mails and online communications. Kana worked closely with these "Lead-Users", directly managing their e-mails, while developing an application to do this systematically.

Kana was able to develop its first application literally in a live production environment and update the application with direct input from the "Lead-Users". Kana was able to map out future product developments such as inbound and outbound customer communications management and online interfaces to these systems, based on working with these "Lead-Users".

7.10.4.2 Defining Potential Development and Use of New and Emerging Applications

Gainey (2002) saw Kana as being at a crossroads in 1998 between offering CRM and new e-commerce applications. Certainly Kana had a "first-mover" advantage with a new application to address e-mail management – but the context for using the application was in a great state of flux.

Customer Relationship Management systems were emerging and needed communications management modules – and especially as these systems were being linked to websites and online business channels.

CRM systems were rapidly redefining themselves to service online business activities. At the same time though, full online business systems were emerging – to become e-commerce systems.

Kana had to stake out some key assessments as to developing – and educating – new enhanced e-CRM systems, or going much further toward integrated e-commerce solutions.

Strategic assessment of the evolution – or revolution – of new applications is a tough challenge. It requires managers and marketers to go beyond conventional product forecasting methods towards scenario planning approaches. These approaches take account of potential usability and product applications – and understanding factors that may enhance or inhibit further product development.

7.10.4.3 Transformation and Repositioning from a Small Software House to a Major Software Infrastructure Provider

Although Kana grew rapidly from 1996 through to mid-1999, it was still a small start-up software house. The IPO process saw Kana shift from a small business mentality toward a software housing seeking to become a large (and dominating) full-service software vendor. Kana's successful IPO in September 1999 provided it with the funding platform to create a large software house, as evidenced by the $US4 billion Silknet deal.

However, by mid-1991 rapid growth had become consolidation and survival, and Kana's next major merger with Broadbase was about maintaining a medium-sized business with a capitalization standing at around $US 60 million in late 2002.

Sometimes such transformation can really move a small business into a bigger business league – but sometimes it can lead to a big but short-lived foray into a high division and then a fall back into a similar division to where the company started.

7.10.4.4 Emphasis on Quality, Specialisation and Solving Specific Online Issues

Both Gainey (2002) and Wolfe (2002) preferred to develop applications and new businesses where applications definitely solved specific online business issues. Extending these concepts to an integrated CRM or full e-commerce application suite was a major debate as Kana grew through a succession of mergers – and different executive teams.

7.10.4.5 Building Applications versus Acquiring Applications and Skills

During Kana's early development (1996-98), Kana's application development was largely internal. In 1999 and early 2000, the executive team was able to plan and quickly execute acquisitions of applications and skills to extend Kana's product functionality.

Kana seemed to possess strong capabilities, not only in targeting strategic and complementary applications, but also rapidly incorporating them into their next product releases.

7.10.4.6 Vertical versus Horizontal Marketing Issues

During 1996-99 Kana specialised in providing a horizontal application for customer communication management. However, Kana did not focus on vertical or industry market segments.

Wolfe (2002) suggested that Kana could have customised its products for specific vertical markets. As Kana moved toward offering integrated e-CRM and e-ecommerce solutions, increased focus on selected vertical markets may become a higher priority for the company.

7.10.4.7 Roles of System Integrators (SIs) and Independent Software Vendors

System Integrators (SIs) and Independent Software Vendor (ISVs) (and other technology partners) are critical partners and channels of communication for emerging software houses. Most start-up software houses have a few sales and marketing staff and perhaps a few business development managers.

Any entity that can add to these capabilities is a valuable resource for the start-up company. SIs can refer and recommend applications for larger systems. If they are sufficiently attracted by the new application they may even employ consultants, designers and even programmers proficient in development and deployment of the application.

Kana was able to build its presence in the market through connecting with regional SIs into the San Francisco/Bay Area, Atlanta and Dallas.

ISVs are vital to start-up as they can incorporate the start-up's applications into their software application, thus extending the scope of the original application into new markets, or across technology platforms. Kana actually released special versions of its products for specific technology platforms so that ISVs developing using those platforms could use Kana's applications.

7.11 Summary

Decision-making issues associated with Kana' rapid development of an idea for e-mail management into a full enterprise e-commerce suite (e-CRM suite) were captured and mapped in a package of DSA models, events chronology maps and cognitive maps and extension questions.

Key insights from the Kana case study include:

7.11.1 Decision-making Associated with Development and Rapid Releases of New Internet-based applications

7.11.2 Decision-making Associated with Extending Broadening Application Capabilities and Suites

7.11.3 Business Growth and Application Development through Rapid and Strategic Acquisitions (and Mergers)

7.11.4 Channels of Distribution and Third-party Partnerships for Internet-based Software Houses

7.11.1 Decision-making Associated with Development and Rapid Releases of New Internet-based Applications

The case study on Kana explored decision-making associated with development and rapid releases of new Internet-based applications for online customer communication management.

Four versions of Kana's application suite were released in three years through a mix of fast internal software development and incorporation of acquired applications to enhance Kana's overall application suite.

Kana's management were focused on rapidly growing its business to a critical mass where they could be sustainable competitors in the "front-office" "e-CRM" portion of enterprise e-commerce solutions.

7.11.2 Decision-making Associated with Extending and Broadening Application Capabilities and Suites

The case study explores the decision-making associated with extending Kana's application capabilities rapidly to handling both outbound and inbound e-mail management, full web-based communication management, online CRM applications, and integrated e-commerce application suites.

Kana was born in 1996 from an idea to handle floods of incoming e-mails. Kana's first application, released in 1998, was very successful and it rode on the wave of emerging e-commerce activity. In fact Kana's application was a key component for emerging CRM and e-commerce application suites and by 1998 interest and demand for the application was growing exponentially.

Kana knew that it needed to grow quickly through acquisitions to develop a comprehensive online customer communication management application suite, but how far should it go toward providing a full e-CRM suite or a fully integrated e-commerce application suite?

Kana's highly successful IPO, close to the peak of the Internet stock in boom in September 1999, enabled it to execute a string of strategic acquisitions to provide a full e-CRM product by early 2000.

7.11.3 Business Growth and Application Development through Rapid and Strategic Acquisitions (and Mergers)

A major component of Kana's application development approach was to bring in additional applications to extend Kana's own applications capabilities. In most instances Kana actually acquired the application provider.

Kana's highly successful IPO set it up to aggressively pursue targeted acquisitions, and to build a critical mass as a larger software vendor.

Kana's merger with Silknet in February 2000 changed the organisation into a large company with a wide range of e-CRM applications. It represented a change from being a small start-up to becoming a large established software house.

That deal may have been the end of the story, except that the collapse of technology stocks in 2000 triggered considerable consolidation and revaluation of Kana such that the merger with Broadbase in 2001 ensured Kana's survival – but as a small to medium software house rather than a large software vendor. Kana has recorded modest but steady growth from mid-2001 through to late 2002.

7.11.4 Channels of Distribution and Third-party Partnerships for Internet-Based Software Houses

Part of the DSA package explores Kana's channels of distribution and its key business partners – and highlights their very high strategic value to Kana.

System Integrators and Technology Partners (ISVs) are fundamental to Kana for referral and recommendation in enterprise computing solutions, and for developers requiring e-CRM applications in their solutions.

Kana Online (KOL) was a particular challenge as a channel for Kana. KOL grew rapidly as Internet start-up companies looked to a hosting service for their online customer communication management.

KOL offered this service and extended it into an Application Service Provider (ASP) model. However, the model was not sustainable in the face of the collapse in Internet stocks in 2000, and KOL was sold in 2001.

During Kana's early days, there seemed to be substantial shared vision between Kana and its "Lead-Users" and early business partners, specifically around addressing e-mail management. However, such shared vision over time became more opportunistic as

Kana worked very closely with selected SIs and ISVs to develop specific applications or versions tailored for them.

Kana, in its current form with BroadBase, works closely with targeted enterprise customers and channels partners to provide a comprehensive e-CRM application suite. Although beyond the timeframe of this case study, Kana seems to be heading toward offering highly customised vertical e-CRM solutions, which is similar to other software houses in this project (for example, Trilogy and Zaplet).

7.12 Conclusions

The Kana case study highlights the decision-making associated with rapid application development with emphasis on extension to a full e-commerce enterprise suite and a strategy of mergers and acquisition.

Conclusions are presented in the sections below:

7.12.1 Theoretical Issues (or Questions) for the Kana Case Study

The Kana case study explores the decision-making issues associated with rapid product development through mergers and acquisitions. The case study was not as explicit about application development cycles as with other cases, such as NetDynamics. Nevertheless, Kana sustained rapid application development in "Internet Time" through releasing seven full versions of its application in four years.

The Kana case study also explored the decision-making and justification for the selection of the development of channels to service a diverse range of customers, but with emphasis on en emerging enterprise customer base.

The storytelling method used for analysis was effective at identifying key decision-making issues, events, and linkages associated with the dynamic evolution of Kana's applications, from the conception of an idea to address an unmet need for e-mail management, right through to the development of a comprehensive e-CRM application suite – in tandem with the rapid development of a new business.

Both methods provided mapping and validating of initial accounts of decision-making, followed up with subsequent revisions of those accounts using a systematic emic/etic representation system.

7.12.2 Kana Case Study Limitations

The Kana case study is based on a complete set of multiple interviews with participants involved in the foundation and early development of Kana.

The case study may have benefited from additional insights from current Kana employees who were also with Kana during 1996-2000. A subset of Kana employees from companies acquired or from merger partners could have provided valuable insights for the case study.

7.12.3 Contributions from the Kana Case study to Strategic Sensemaking

The Kana case study contributes to strategic sensemaking through application of DSA, augmented by cognitive mapping of rapid application development.

The Extended DSA approach maps insights into the "scalability" of dynamic decision-making associated with rapid expansion of software applications.

For the Kana case, a major aspect of the decision-making was the identification and incorporation of third-party applications into an emerging suite of applications as part of a larger vision to rapidly take a simple idea (the management of incoming e-mail) and turn it into a full e-commerce application suite (a full e-CRM application).

The Kana case study applied extended DSA to channel selection and development. A distinct concept that emerges from extended DSA in the case is the acknowledgement of both channels of distribution and channels of development.

7.12.4 Contributions from the Kana Case Study to Practice

The Kana case study provides insights into both rapid application and new organisation development. The case addresses Kana's experiences with new business development, including initial business establishment and early funding, setting up for an IPO, and mergers and acquisitions in a quest to become a major enterprise e-commerce application provider.

The Kana case study provides distinct insights into the management of application development using mergers and acquisitions. By 2000, Kana was a consolidated entity that had digested at least three software houses over 12 months – and its merger partner Silknet itself was a product of at least three acquired or merged software houses in the previous two years. And that was before Kana/Silknet merged with Broadbase in 2001.

Practical insights into channels and their selection are offered in the Kana case study that are interesting to explore, given the very high merger and acquisition focus for Kana in 1999-2000.

A recurring theme in the overall research project is the importance of System Integrators as extended developer and sales and marketing resources for small, but growing software houses. Kana was very conscious of the strategic selection of SIs, particularly for expansion throughout key regions within the U.S.A.

"Channels of Development" as a concept has theoretical and practical application. Specific services and programs were created for development partners. Kana created distinct sub-versions for platforms and operating systems of key computer vendors. Kana also created System Developer Kits (SDKs) for its ISVs and developer communities.

The Kana case study, similar to the NetDynamics case study (and to a lesser extent, the Intuit QuickBooks case study), is focused on start-up business activity. Interviews and insights are drawn from entrepreneurs focused on company creation, but who are also focused on the creation of new companies.

However, given that in 2003 Kana is only five year old as a company, it may still be too early to fully understand the issues associated with transitioning a start-up software house into an established business, capable of supporting large enterprise customer bases.

7.12.5 Kana Case Study Endnote

Kana released Kana 7 in mid-2001 with comprehensive "e-CRM" and e-commerce application suites.

The merged Kana Communications/Broadbase entity consolidated its applications and businesses as Kana Inc. during mid-2001.

Kana repackaged its applications into the Kana iCARE Family launched on 24 September 2001. "Applications in the repackaged application suite include:

- KANA Contact Center™ – a full customer service application for contact centers.
- KANA IQ™ – Bringing together a self-service customer application that may work with or along contact center agents
- KANA ResponseIQ™ – E-mail management with knowledge management application
- KANA Response® – Kana's established e-mail management system
- KANA Marketing™ – a multi-channel marketing automation application

- KANA iCARE Analytics™ – Analytic CRM application".

(Adapted from Hohn, 2001).

In line with other technology stocks, Kana trades at prices a long way below the multibillion dollar valuations of early 2000, with a market capitalization of approximately $US46 million (NASDAQ valuation at close of market 13 February 2005).

Nevertheless, Kana is a major supplier of "e-CRM" solutions competing strongly against other CRM providers including: Oracle, Siebel Systems, E.piphany and Vignette. Kana commenced release of the eighth version of its iCARE suite with Kana IQ 8 in September 2003.

Mapping Implemented Strategies of Bringing Internet-based Software Applications to Market Using Storytelling Research Methods

by

Hugh M Pattinson

A thesis submitted in partial fulfilment of the requirements for the degree of

Doctor of Philosophy (PhD)

University of Technology, Sydney

2005

VOLUME II

Chapter 8: Case Study – Intuit QuickBooks

Abstract

The Intuit QuickBooks case study explores multiple iterations of strategic decision-making over time through the application of two storytelling methods for analysis – an **advanced hermeneutic framework**, and an **extended form of decision systems analysis (DSA)** incorporating **cognitive mapping**.

A well-established software house with strict internal development standards and processes is faced with having to transition its decision-making simultaneously on two fronts. It has to open up its development processes to accommodate third-party applications and services, while at the same time creating an online business and developer presence.

Three iterations of online strategy executed between 1998 and 2002 are analysed. Intuit QuickBooks started with a strategic assumption that a new software industry based on the Internet would emerge underwritten by advertising revenues – similar to early commercial television business models.

Intuit QuickBooks experimented with a portal strategy in 1998, but shifted to the development of an online environment with developer and business services in 1999-2000. A virtual community for both developers and users was created under the QB Economy umbrella in 2001-02.

Findings from the case study include the challenges associated with moving from internal product-focused decision-making to an outward services focus; the management of third parties in application development and online communities; and the reconceptualising of applications into online environments for both development and service delivery.

8.1 Background

Intuit Inc. of Mountain View California provides accounting applications for personal and small-business users through its QuickBooks and Quicken divisions.

Intuit was founded in 1983 and is the oldest software house in the research project. Intuit has well-established application development groups, cycles and delivery processes.

Intuit was included in the final case study list for the project as it met the initial criteria:

- Software has to be “beta”-tested and commercially introduced
    - The case study addresses an application with standardised annual releases dating as far back as 1992
- Case-studies were written by well-known authors
    - Professor Rajiv Lal is regarded as an expert in retailing, promotion and marketing management. His most recent work has focused on the impact of using the Internet as a channel of distribution on a retailer's pricing, merchandising and branding strategy.
- The application development process studied must be over a timeframe of more than 12 months
    - the original case study covers a timeframe of just over three years from 1996-99, with reference to Intuit’s history going back to 1983, and to 1992 for the creation of the QuickBooks
- Names of case study companies and participants can be disguised if requested
    - Permission was gained from Tim Villanueva and Terry Hicks, both for a combination of face-to-face and digitally recorded telephone interviews and the use of that information in the research project, without disguise
- Applications must be Internet-based
    - The case addresses the transition of Intuit QuickBooks application into an extended online application incorporating services. Application development and distribution using the Internet are addressed in the case study.

Additional case study selection criteria included:

- Size: Large
- Success: Yes
- Disruptive Technology: Yes, QuickBooks is regarded as a disruptive technology as it enables small businesses to manage their own accounting systems
- Harvard Business School Case: "Intuit QuickBooks", Kochikar & Lal (2000).

The case study traces Intuit from its launch of QuickBooks in 1992 through to its efforts to develop an online presence for QuickBooks and Quicken in 1999-2002.

The Initial DSA model, events chronology map and the three cognitive maps for this case were developed using extracts drawn from Kochikar & Lal's (2000) Harvard Business School Case study, "Intuit QuickBooks".

Kochikar & Lal (2000) present a brief introduction on the foundation of Intuit in 1983, and then "fast-forward" to the launch of QuickBooks in 1992, focusing on Intuit's attempts to develop an online strategy for its Quicken and QuickBooks products during 1998 to 2000.

Kochikar & Lal (2000) provide a detailed description of the product development and product development processes at Intuit QuickBooks, which is sufficient for the development of a representative DSA model and to undertake further analysis on application development and delivery.

The outline and chapter structure for the Intuit QuickBooks case study are presented in Figure 8.1.

The DSA model and maps in this case were updated, plus extension questions on dominant logic, shared vision, key leverage points and strategic marketing issues were addressed through interviews with one of the executives responsible for the development and execution of Intuit QuickBooks' online strategies in 1998-1999, plus the current Vice President of Development at Intuit Inc. (see Villanueva 2002; Hicks 2002).

8.2 Theoretical Issues (or Questions) for the Intuit QuickBooks Case Study

The case study examines the decision-making associated with an established software house attempting to position its applications and its company in a new online environment. At least two changes in strategy to move into an online environment are observed and further analysed through DSA models and cognitive maps.

Following on from the overall decision-making associated with moving into an online environment is the transition of application development from an internal organisation focus to inclusion of ISVs and third-party developers and service providers. The case study provides a sharp contrast to other case-studies in the project where applications are literally "born online" or "born in the Internet".

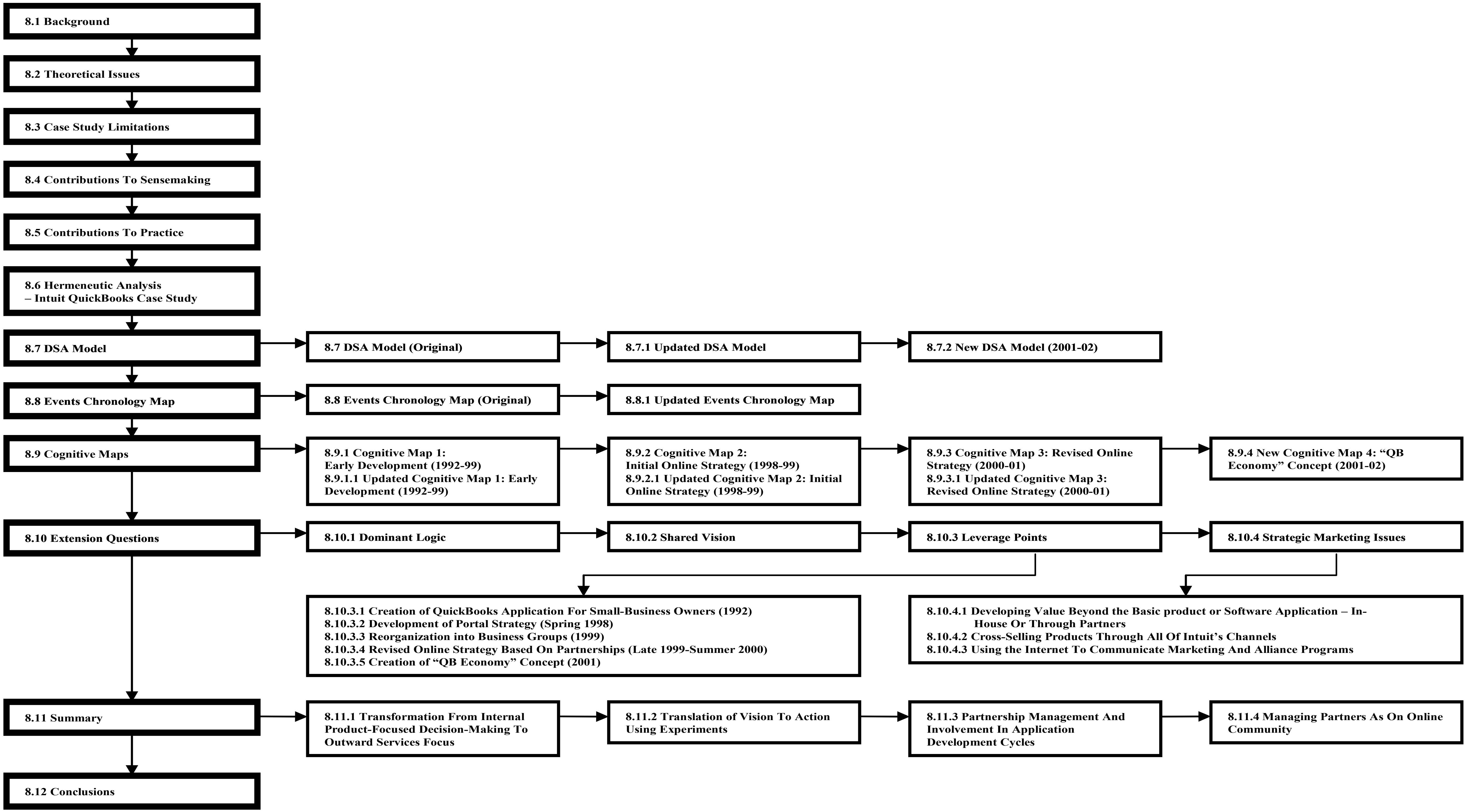


Figure 8.1 Intuit QuickBooks Case Study – Chapter Structure.

The storytelling methodology used for analysis in the case study – an advanced hermeneutic analysis, including an extended form of **decision systems analysis** (DSA) incorporating **cognitive mapping** – were effective in identifying key decision-making issues, events, and linkages.

The methodology provided mapping and validating of the initial accounts of decision-making, followed up with subsequent revisions of those accounts using a systematic emic/etic representation system.

The case study was extended through interviews to provide significantly greater insights into third-party involvement with Intuit QuickBooks than in the initial scope of the case study.

New DSA models and cognitive maps provide further insights into the logic associated with the setting up and management of new channels of distribution (and development) specifically related to Intuit QuickBooks' own application development and that of Intuit QuickBooks' ISVs.

8.3 Intuit QuickBooks Case Study Limitations

The Intuit QuickBooks case study benefited from a complete set of multiple interviews with multiple respondents who were directly involved in the original case study on the company. However, the new DSA models and new cognitive maps which were developed from interview material were not validated.

Given that there is a significant focus on third-party developers and service providers in the Intuit QuickBooks case study there are limitations associated with obtaining no interviews or inputs from them.

An extension of the research methodology from an internal exploration of decision-making toward the inclusion of selected third-parties would be useful, although a clearly defined process for identifying and qualifying appropriate third-parties would be required.

8.4 Contributions from the Intuit QuickBooks Case Study to Strategic Sensemaking

The Intuit QuickBooks case study explored decision-making issues, events, and linkages using the storytelling method developed for the project. An advanced hermeneutic analysis provided several levels of emic/etic representation of decision-making, highlighting key case study issues. The Hermeneutic framework incorporated an extended form of Decision Systems Analysis (DSA) highlights specific subsets of decision-making, through mapping of decision-making from original accounts with revisions, through subsequent contact and discussions with original decision-makers. The extended DSA also explored changes over time in shared vision and new application development.

The Intuit QuickBooks case study applies an extended DSA approach using DSA models, events chronology maps plus cognitive mapping to decision-making for application development and product strategy over time.

Extended DSA provides insight into three strategic revisions to both Internet application development and to establishment of a computer-mediated environment for business-to-business services and support.

The Intuit QuickBooks case study provides mapping of decision-making associated with an increasing inclusion of third-party applications and services in application development cycles.

The case study highlights a fundamental transformation to strategic thinking to reconceptualise “application” (or product) development for computer-mediated environments to extend application development into a set of services with the potential for new and different value-added online products and services.

8.5 Contributions from the Intuit QuickBooks Case Study to Practice

Although the World-Wide Web has been in commercial use for around ten years, challenges remain for established software houses to develop applications for that environment. There are additional issues with established software houses transitioning toward providing applications and services in an online (World-Wide Web) environment.

Intuit QuickBooks is a classic example of a software house established for more than 20 years with well-defined organisational structures, and strictly defined application development cycles.

Intuit QuickBooks modified its organisational structure and development cycles to transition quickly into developing applications for an online environment. The case study highlights a number of iterations in decision-making for Intuit QuickBooks to transition its application, plus associated applications and services, into an online environment.

Insights into these iterations are valuable for software houses that are yet to commit to developing Web-based applications, but also for those software houses that may be in the early stages of transition into online environments.

The Intuit QuickBooks case study challenges executives to rethink what actually constitutes an “online product” or application. Alliances and partnerships are critical for producing a full online application offering – to provide additional supporting information services beyond basic applications.

8.6 Hermeneutic Analysis – Intuit QuickBooks Case Study

The Intuit QuickBooks case study offers an application of the proposals described in Chapter 1 for advancing hermeneutic research in B2B decision-making.

Five levels of hermeneutic analysis were developed to address decision-making issues, events, and linkages in the Intuit QuickBooks case study. These levels are presented in Exhibits 8.1-8.5.

The original case study by Kochikar & Lal (2000) serves as the etic 1 report of the case study – although there are elements of emic 1 reporting in the form of direct quotes from decision-makers in the account. Quotes and perspectives from decision-makers were encapsulated in the Level 1 emic analysis in Exhibit 8.1.

Key case study issues as articulated by Kochikar & Lal (2000) were added to the emic 1 data in the Level II analysis and presented as etic 1 data in Exhibit 8.2. These issues included:

- Intuit QuickBooks was a successful application with more than three million users and over one million upgrades and new versions sold each year
- Intuit QuickBooks was slow to move into online business
- Intuit needed to develop an online strategy focused on providing online services to complement the QuickBooks offline package
- First online strategy to add a small business channel to Quicken.com was not successful
- Revised online strategy included a channels framework to evaluate potential service providers and developers and to tier them into different levels of relationship with QuickBooks.

The Level II analysis also explored mental models and decisions/actions associated with these issues.

Exhibit 8.1 Level I Analysis of Intuit QuickBooks Case.

Emic 1 view:
- QuickBooks' future success depended on its ability to quickly and effectively generate incremental, recurring revenues by harnessing the Internet to offer value-added services to its installed base of customers
- (Villanueva) championed the transformation of Quicken from a solely desktop product to a hybrid product with a strong online component
- To develop online strategy balancing QuickBooks to effectively compete without compromising the values and culture of the company (rigorous development standards and strict development and release cycles)
- Cook (Intuit founder) believed that ad-supported Internet would trigger explosive growth in the software industry, just as ad-supported television triggered explosive growth in the media industry.
- Developed a simple framework to help review the partnership proposals objectively (part of revised online strategy)
- Challenges with developing tiered channels strategy for prospective partners.

↓

1983-98 (Offline)/1998-99 (Online). Initial mental model:
- Intuit Inc established in 1983 – focus on financial software industry
- Intuit releases QuickBooks – an accounting package for small business owners (1992)
- Fast rise in sales through the Internet but no specific website or onsite strategy for QuickBooks (5% by 1999)
- Experimentation with channels on existing Quicken.com site
- Shift to advertising model similar to media industry to drive a development of a new software industry
- Revision to focus on developing third-party relationships to create an extended set of online services for QuickBooks

Decisions-actions:
- Developed accounting package for small business owners; Accepted sales through Internet during late 1990's; Initial online strategy to offer services related to QuickBooks through a small business channel on Quicken.com (1998); Revised online strategy to focus on Value-Added eServices, and online extensions to QuickBooks through browser access; Development of framework for partner evaluation ranging from "light" to "deep" relationships with QuickBooks (1999)

Exhibit 8.2 Level II Analysis of Intuit QuickBooks Case

Etic 1 view:
- Intuit QuickBooks was a successful application with more than 3 million users and over 1 million upgrades and new versions sold each year
- Intuit QuickBooks were slow to move into online business
- Intuit needed to develop an online strategy focused on providing online services to complement QuickBooks offline package
- First online strategy to add a small business channel to Quicken.com was not successful
- Revised online strategy included a channels framework to evaluate potential service providers and developers and to tier them different levels of relationship with QuickBooks

Emic 1 view:
- QuickBooks's future success depended on its ability to quickly and effectively generate incremental, recurring revenues by harnessing the Internet to offer value added services to its installed base of customers
- (Villanueva) championed the transformation of Quicken from a solely desktop product to a hybrid product with a strong online component
- To develop online strategy balancing QuickBooks to effectively compete without compromising the values and culture of the company (rigorous development standards and strict development and release cycles)
- Cook (Intuit founder) believed that ad-supported Internet would trigger explosive growth in the software industry, just as ad-supported television triggered explosive growth in the media industry.
- Developed a simple framework to help review the partnership proposals objectively (part of revised online strategy)
- Challenges with developing tiered channels strategy for prospective partners

1983-1998 (Offline)/1998-1999 (Online). Initial mental model:
- Intuit Inc established in 1983 – Focus on Financial Software Industry
- Intuit releases QuickBooks – an accounting package for Small Business Owners (1992)
- Fast rise in sales through the Internet but no specific website or onsite strategy for QuickBooks (5% by 1999)
- Experimentation with Channels on existing Quicken.com site
- Shift to Advertising Model similar to Media Industry to drive a development of a new software industry
- Revision to focus on developing 3rd party relationships to create an extended set of online services for QuickBooks

Decisions-actions:
- Developed accounting package for small business owners; Accepted sales through Internet during late 1990's; Initial online strategy to offer services related to QuickBooks through a small business channel on Quicken.com (1998); Revised online strategy to focus on Value-Added eServices, and online extensions to QuickBooks through browser access; Development of framework for partner evaluation ranging from "light" to "deep" relationships with QuickBooks (1999)

Exhibit 8.3 Level III Analysis of Intuit QuickBooks Case

Etic 1 view:
•Intuit QuickBooks was a successful application with more than 3 million users and over 1 million upgrades and new versions sold each year
•Intuit QuickBooks were slow to move into online business
•Intuit needed to develop an online strategy focused on providing online services to complement QuickBooks offline package
•First online strategy to add a small business channel to Quicken.com was not successful
•Revised online strategy included a channels framework to evaluate potential service providers and developers and to tier them different levels of relationship with QuickBooks

Etic 2 view:
•Initially an experimentation view with online business – e.g. collect orders, then try out a portal approach
•Small Business Channel on Quicken.com really not reach their small business customers who were not regular users of the Internet
•Revised Online strategy included a rigorous partner selection framework – partly to accommodate issues with stringent quality controls within Intuit Inc

Emic 1 view:
•QuickBooks's future success depended on its ability to quickly and effectively generate incremental, recurring revenues by harnessing the Internet to offer value added services to its installed base of customers
• (Villanueva) championed the transformation of Quicken from a solely desktop product to a hybrid product with a strong online component
• To develop online strategy balancing QuickBooks to effectively compete without compromising the values and culture of the company (rigorous development standards and strict development and release cycles)
•Cook (Intuit founder) believed that ad-supported Internet would trigger explosive growth in the software industry, just as ad-supported television triggered explosive growth in the media industry.
•Developed a simple framework to help review the partnership proposals objectively (part of revised online strategy)
•Challenges with developing tiered channels strategy for prospective partners

1983-1998 (Offline)/1998-1999 (Online). Initial mental model:
•Intuit Inc established in 1983 – Focus on Financial Software Industry
•Intuit releases QuickBooks – an accounting package for Small Business Owners (1992)
•Fast rise in sales through the Internet but no specific website or onsite strategy for QuickBooks (5% by 1999)
•Experimentation with Channels on existing Quicken.com site
•Shift to Advertising Model similar to Media Industry to drive a development of a new software industry
•Revision to focus on developing 3rd party relationships to create an extended set of online services for QuickBooks
Decisions-actions:
•Developed accounting package for small business owners; Accepted sales through Internet during late 1990's; Initial online strategy to offer services related to QuickBooks through a small business channel on Quicken.com (1998); Revised online strategy to focus on Value-Added eServices, and online extensions to QuickBooks through browser access; Development of framework for partner evaluation ranging from "light" to "deep" relationships with QuickBooks (1999)

The third level of analysis, as presented in Exhibit 8.3, is an etic 2 representation based on the current researcher's summarisation of event milestones and the emic 1 sensemaking views identified in the data in the Kochikar & Lal (2000) case study. Subsequently to developing Exhibit 8.3, emic 2 and etic 2 interpretations, along with DSA and event maps, provided revisions to Exhibit 8.3.

Kochikar & Lal's (2000) account focuses on Villanueva and Dan Nye's decision-making associated with developing and revising an online strategy for the Intuit QuickBooks application. The account was sufficient for the development of an initial (etic 2) representative DSA model and to undertake further analysis on application development and delivery.

The initial DSA, event chronology, and cognitive maps reported below offer details supporting Exhibit 8.3. Etic 2 perspectives are mainly based from the initial DSA model, initial events chronology map, and initial cognitive maps.

Key etic2 issues included:

- Initially an experimentation view with online business – for example, collect orders, then try out a portal approach
- Small Business Channel on Quicken.com did not really reach their small business customers who were not regular users of the Internet
- Revised Online strategy included a rigorous partner selection framework – partly to accommodate issues with stringent quality controls within Intuit Inc.

The etic 2 DSA model and maps for the case were updated following extensive questioning of the accuracy and completeness of the original Kochikar & Lal's (2000) case – the collected data represents emic 2 material and was added during the fourth level of analysis (See Exhibit 8.4).

Thus, additional (emic 2) data were collected for etic 3 description and interpretation of the decision-making process as reported in the original case study and for a period of three years beyond that reported by Kochikar & Lal. Emic 2 data consists of responses from interviews with the then Director of Business Development for QuickBooks, and the current (mid-2002) VP of Development at Intuit Incorporated (see Villanueva 2002a & 2002b; Hicks 2002a & 2002).

Exhibit 8.4 Level IV Analysis of Intuit QuickBooks Case

Etic 1 view:
•Intuit QuickBooks was a successful application with more than 3 million users and over 1 million upgrades and new versions sold each year
•Intuit QuickBooks were slow to move into online business
•Intuit needed to develop an online strategy focused on providing online services to complement QuickBooks offline package
•First online strategy to add a small business channel to Quicken.com was not successful
•Revised online strategy included a channels framework to evaluate potential service providers and developers and to tier them different levels of relationship with QuickBooks

Emic 2 view:
•Developing Value Beyond the Basic product or Software Application – In-House Or Through Partners
•Importance Cross-Selling Products Through All Of Intuit's Channels
•Using the Internet To Communicate Marketing And Alliance Programs
•Channels Selection Process for Revised online Strategy was more opportunistic than a clearly defined framework
•Development of a complete Online Developer and Services Provider Community associated with Intuit Inc – and QuickBooks
• QB Economy Umbrella Branding for QuickBooks Online Services

Emic 1 view:
•QuickBooks's future success depended on its ability to quickly and effectively generate incremental, recurring revenues by harnessing the Internet to offer value added services to its installed base of customers
• (Villanueva) championed the transformation of Quicken from a solely desktop product to a hybrid product with a strong online component
• To develop online strategy balancing QuickBooks to effectively compete without compromising the values and culture of the company (rigorous development standards and strict development and release cycles)
•Cook (Intuit founder) believed that ad-supported Internet would trigger explosive growth in the software industry, just as ad-supported television triggered explosive growth in the media industry.
•Developed a simple framework to help review the partnership proposals objectively (part of revised online strategy)
•Challenges with developing tiered channels strategy for prospective partners

Etic 2 view:
•Initially an experimentation view with online business – e.g. collect orders, then try out a portal approach
•Small Business Channel on Quicken.com really not reach their small business customers who were not regular users of the Internet
•Revised Online strategy included a rigorous partner selection framework – partly to accommodate issues with stringent quality controls within Intuit Inc

1983-1998 (Offline)/1998-1999 (Online). Initial mental model:
•Intuit Inc established in 1983 – Focus on Financial Software Industry
•Intuit releases QuickBooks – an accounting package for Small Business Owners (1992)
•Fast rise in sales through the Internet but no specific website or onsite strategy for QuickBooks (5% by 1999)
•Experimentation with Channels on existing Quicken.com site
•Shift to Advertising Model similar to Media Industry to drive a development of a new software industry
•Revision to focus on developing 3rd party relationships to create an extended set of online services for QuickBooks
1998-2002 Extended mental model:
•Branding combined application and online services (2001-2002)
•Multilevel Developer and Service Partner Relationships – At the Intuit Corporate Level
•Emphasis on Strong Interaction and Communications With Key Partners
Decisions-actions:
•Developed accounting package for small business owners; Accepted sales through Internet during late 1990's; Initial online strategy to offer services related to QuickBooks through a small business channel on Quicken.com (1998); Revised online strategy to focus on Value-Added eServices, and online extensions to QuickBooks through browser access; Development of framework for partner evaluation ranging from "light" to "deep" relationships with QuickBooks (1999); Creation of "QB Economy' umbrella branding (2001); Redevelopment of Intuit Developer Network for all Intuit Partners (2001-2002)

Exhibit 8.5 is a fifth level of analysis with the addition of etic 3 data summarising emic 2 interpretations of mental models and events covered in the original case study as well as the work completed for the etic 2 interpretation – including the DSA, event chronology, and cognitive maps developed for the etic 2 interpretation.

The revised DSA, event chronology, and cognitive maps presented in subsequent sections below follow from the emic 2 interpretations, and these maps are part of the etic 3 interpretation.

The new data from the emic 2 and etic 3 rounds of interpretation validated etic 2 data and highlighted new insights related to:

- Transformation from internal product-focused decision-making to outward services focus
- Translation of vision to action using experiments
- Partnership management and involvement in application development cycles
- Managing partners as an online community
- Revised online strategy was revised with a more corporate emphasis
- Deeper segmentation of QuickBooks Applications in 2003
- Finally a true online version of the QuickBooks Application (2003)!

Overall, the five-level hermeneutic framework for the Intuit QuickBooks case study provided a strong and comprehensive framework for capturing and extending dynamic sensemaking for Intuit QuickBooks' (and later Intuit Inc.) decision-makers as they experimented with a number of online strategies for the QuickBooks application.

8.7 DSA Model

The original DSA model was developed from a review of the whole Kochikar & Lal (2000) case study, but mainly from sections on product development and the product development process (see Kochikar & Lal 2000, 2-3).

The account is focused on QuickBooks development but reflects overall application development standards for Intuit Inc., including the Quicken application. The original DSA model is presented in Figure 8.2 and is representative of QuickBooks application development during 1998-2000.

By 1998, there was a well-established annual application development cycle for QuickBooks. The cycle described by Kochikar & Lal was for versions of the QuickBooks application for 1998, 1999 and 2000.

Exhibit 8.5 Level V Analysis of Intuit QuickBooks Case

Etic 1 view:
•Intuit QuickBooks was a successful application with more than 3 million users and over 1 million upgrades and new versions sold each year
•Intuit QuickBooks were slow to move into online business
•Intuit needed to develop an online strategy focused on providing online services to complement QuickBooks offline package
•First online strategy to add a small business channel to Quicken.com was not successful
•Revised online strategy included a channels framework to evaluate potential service providers and developers and to tier them different levels of relationship with QuickBooks

Emic 2 view:
•Developing Value Beyond the Basic product or Software Application – In-House Or Through Partners
•Importance Cross-Selling Products Through All Of Intuit's Channels
•Using the Internet To Communicate Marketing And Alliance Programs
•Channels Selection Process for Revised online Strategy was more opportunistic than a clearly defined framework
•Development of a complete Online Developer and Services Provider Community associated with Intuit Inc – and QuickBooks
• QB Economy Umbrella Branding for QuickBooks Online Services

Emic 1 view:
•QuickBooks's future success depended on its ability to quickly and effectively generate incremental, recurring revenues by harnessing the Internet to offer value added services to its installed base of customers
• (Villanueva) championed the transformation of Quicken from a solely desktop product to a hybrid product with a strong online component
• To develop online strategy balancing QuickBooks to effectively compete without compromising the values and culture of the company (rigorous development standards and strict development and release cycles)
•Cook (Intuit founder) believed that ad-supported Internet would trigger explosive growth in the software industry, just as ad-supported television triggered explosive growth in the media industry.
•Developed a simple framework to help review the partnership proposals objectively (part of revised online strategy)
•Challenges with developing tiered channels strategy for prospective partners

Etic 2 view:
•Initially an experimentation view with online business – e.g. collect orders, then try out a portal approach
•Small Business Channel on Quicken.com really not reach their small business customers who were not regular users of the Internet
•Revised Online strategy included a rigorous partner selection framework – partly to accommodate issues with stringent quality controls within Intuit Inc

1983-1998 (Offline)/1998-1999 (Online). Initial mental model:
•Intuit Inc established in 1983 – Focus on Financial Software Industry
•Intuit releases QuickBooks – an accounting package for Small Business Owners (1992)
•Fast rise in sales through the Internet but no specific website or onsite strategy for QuickBooks (5% by 1999)
•Experimentation with Channels on existing Quicken.com site
•Shift to Advertising Model similar to Media Industry to drive a development of a new software industry
•Revision to focus on developing third-party relationships to create an extended set of online services for QuickBooks
1998-2002 Extended mental model:
•Branding combined application and online services (2001-2002)
•Multilevel Developer and Service Partner Relationships – At the Intuit Corporate Level
•Emphasis on Strong Interaction and Communications With Key Partners
Decisions-actions:
•Developed accounting package for small business owners; Accepted sales through Internet during late 1990's; Initial online strategy to offer services related to QuickBooks through a small business channel on Quicken.com (1998); Revised online strategy to focus on Value-Added eServices, and online extensions to QuickBooks through browser access; Development of framework for partner evaluation ranging from "light" to "deep" relationships with QuickBooks (1999); Creation of "QB Economy' umbrella branding (2001); Redevelopment of Intuit Developer Network for all Intuit Partners (2001-2002); Launch of "Right for My Business Strategy (2003); Full online version Of QuickBooks released (2003)

Etic 3 view:
•Etic 2 views validated or extended
•Transformation From Internal Product-Focused Decision-Making To Outward Services Focus
•Translation of Vision To Action Using Experiments
•Partnership Management And Involvement In Application Development Cycles
•Managing Partners As On Online Community
•Revised Online Strategy was revised with a more corporate emphasis
•Deeper Segmentation of QuickBooks Applications in 2003
•Finally a true online version of the QuickBooks Application (2003)!

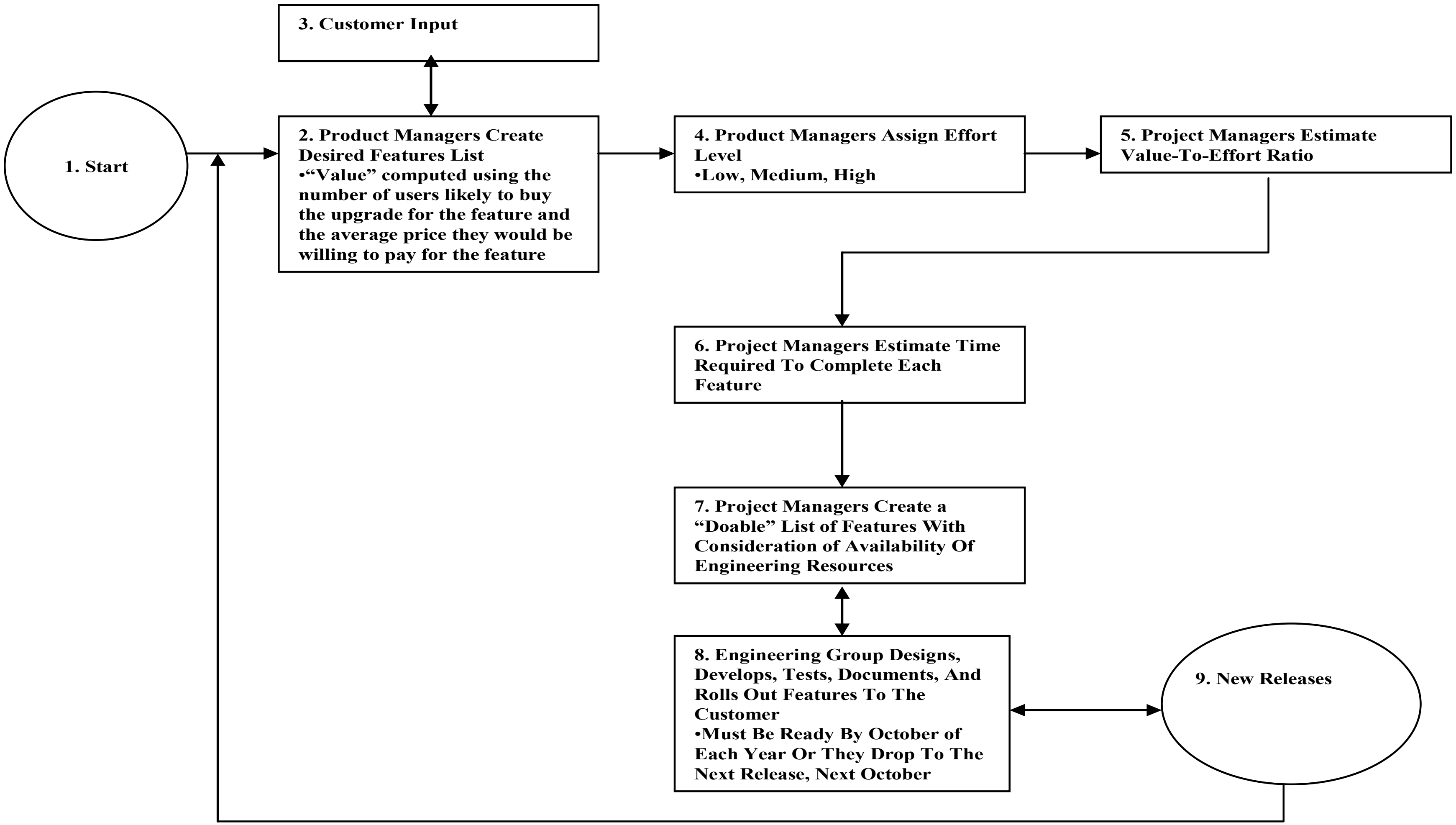


Figure 8.2 Summary DSA Model For Intuit QuickBooks Application Development (1998-2000)
Created from data in Kochikar & Lal (2000)

Product Management for the QuickBooks application creates and prioritises a "Desired Features List" using customer feedback (see boxes 1 and 2). Customer feedback is sourced mainly from thousands of small customers through surveys and electronic communication.

Product Managers also incorporate features planned for the previous version of QuickBooks but eventually dropped from the previous development cycle.

Product Managers perform further prioritisation by effort level into low, medium and high classifications (box 4). Further assessment by value-to effort plus the time required to complete each feature is undertaken by the Project Managers (boxes 6 and 7).

A list is created of "doable" features further prioritised for consideration of the available software engineering resources (box 8).

The Engineering Group team working on the QuickBooks application development undertakes the actual design, development, testing, documentation, and roll-out of the new version of QuickBooks.

The actual time allowed for engineering work is not defined in Kochikar & Lal's (2000) case study, but the new version must be certified for release by October each year, or outstanding unfinished work is dropped into the next version in October of the following year (box 8).

The new version of QuickBooks is formally released in October – November (box 9). Planning for the next application development cycle starts in October – November.

The original DSA model is "Intuit-Centric" with Product Management and Intuit's Engineering planning and delivering new versions of QuickBooks applications.

There is very little input from, and incorporation of ISV applications and third-party services. Customer input is broad as there are thousands of small customers – which is not surprising as QuickBooks (at least in the mid-to-late 1990s) was essentially a business-to-consumer (B2C) application.

Given the QuickBooks and Quicken Groups' initiatives to move online in 1998-999 with emphasis on ISV applications and additional online services, more description on third-party inputs is required to produce a DSA model reflecting these developments.

8.7.1 Updated DSA Model

The interview stages of the case study provided substantial additional detail and insights into Intuit Inc.'s development processes and the evolution of application development. Interviewees included the developer of Intuit QuickBooks' initial and revised online strategies, and Intuit Inc.'s current Vice President of Development (Villanueva 2002; Hicks 2002). An updated Summary DSA model is presented in Figure 8.3.

Planning for each new version of the QuickBooks application commences in October – November in conjunction with the release of the previous version (see box 1).

Selection of third-party ISV applications and services is a now a specific aspect of decision-making in the QuickBooks' application development cycle. Product Managers spend a significant amount of time early in the planning stages for the new version of the QuickBooks application negotiating with third-party applications and services providers (boxes 2 and 3).

The creation of the "desired features list" includes features and services planned for the previous QuickBooks version, but not incorporated in time for that release (box 4). Customer input remains unchanged from the original DSA model (box 5)

Steps associated with the assignment of features by effort level, estimation of value-to-effort ratios, estimation of completion times for features and creation of "doable" feature lists remain unchanged from the original DSA model (boxes 6, 7, 8, 9, 10). However, the final list of "doable" features has to be set by the March prior to the next QuickBooks version release (box 10).

All features must be code complete or ready for testing by June (box 11). Most of the remainder of the cycle is focused on alpha and beta Testing where the new application is subjected to internal QA testing and selected customer testing (box 12).

The new version of the QuickBooks Application is ready for release in October and released in October – November (box 13).

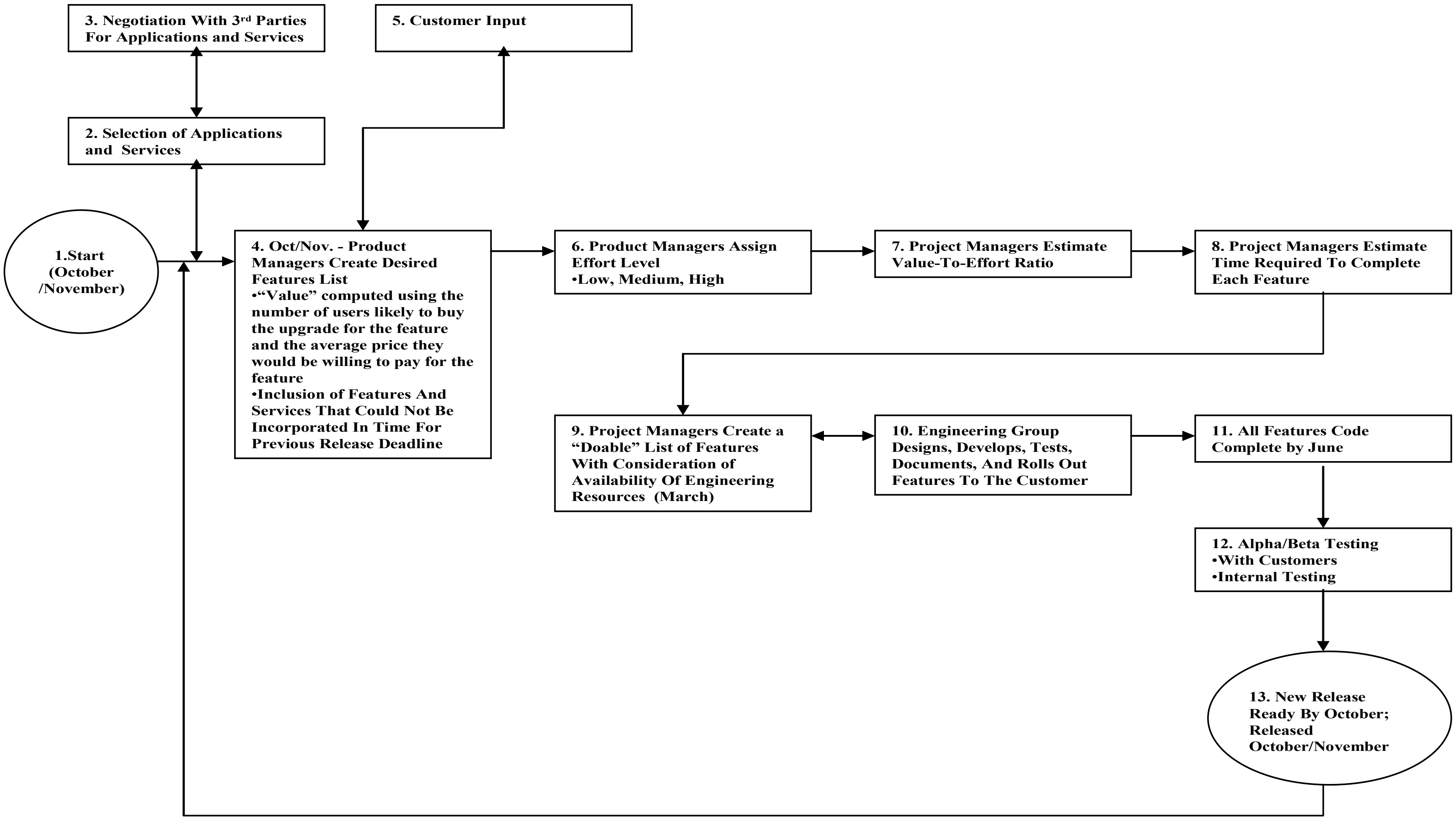


Figure 8.3 Updated Summary DSA Model For Intuit QuickBooks Application Development (1998-2000)
Created from data in Kochikar & Lal (2000), Villanueva (2002); Hicks (2002)

The updated DSA model reflects Intuit's increased involvement with third-party application developers and service providers, and a structured set of milestones for Intuit's annual application development cycles.

Intuit QuickBooks' development processes were the closest to the waterfall or linear development approach, out of all the software houses in the research project.

8.7.2 New DSA Model

Substantial additional material was collected during the interview to warrant the creation of a new DSA model for Intuit Inc.'s application development cycles in 2001-03. Intuit launched the "QB Economy" concept in 2001 with strong emphasis on providing additional application functionality and supporting services for the QuickBooks application.

A vignette was prepared encapsulating insights into, and details of the "QB Economy", to be used to prepare both a new DSA model and cognitive map (see Exhibit 8.6).

The new DSA model for QuickBooks Application Development during 2001-03 is based on QuickBooks 2002 and is presented in Figure 8.4.

Exhibit 8.6 Vignette – Intuit QuickBooks "QB Economy" Concept

Leveraging the "Quickbooks" Brand into a Community of Partners

During 2001, Intuit's Product Development Group established new and extensive relationship programs for its third-parties. Intuit recognised that strong customer, developer and service provider awareness of the "QuickBooks" brand-name could be leveraged to establish a community. The Intuit Developer Network was launched under the "QB Economy" label.

Extension from a Product-focused strategy toward a "Business Environment" approach epitomised by the "QB Economy" label, prompted Intuit to rapidly expand the range of applications and services to fit with the QuickBooks application.

Exhibit 8.6 Vignette – Intuit QuickBooks "QB Economy" Concept (Cont'd)

Vertical applications could extend QuickBooks' appeal to specific SME groups, for example, Retail Applications (including POS systems), Professional Taxpayers and Tax Agents. Vertical services could provide information to support or enhance QuickBooks customised applications.

Horizontal applications could extend QuickBooks capabilities for most SMEs and some large corporation, for example, E-Commerce solutions, Merchant Accounting (Credit Card Services), interfaces between SME and Financial Services Providers, and management of Local, State and Federal Sales and Goods and Services Tax requirements.

The Intuit Developer Network was a vehicle for the developers of QuickBooks in particular to quickly incorporate new features and services while Intuit could focus internally on upgrading the core QuickBooks product and on partnership management.

An "Open" Software Development Approach

Intuit classified its partner relationships by level of service and application integration with the Quickbooks application. Three levels were categorised as:

1. User Interface – Heavily Integrated Partnership both at the application development and service provision levels
2. Data Exchange – Sharing of data between applications and services.
3. Lite Relationships – Standalone Applications and Services to QuickBooks. Selling to the QuickBooks Customer Base as an "Out of box Experience (OOBE)"

For the User Interface and Data Exchange levels, Intuit extended its Developer packaging to include QuickBooks System Developer Kits (SDKs) and capability to export and import data between targeted Services and Applications. Intuit is planning to establish comprehensive ISV programs to accommodate several thousand developers and partners by 2003.

During 2001/02 Intuit shifted toward Application Development Cycles with early incorporation of third-party applications and services. During 1998-2000, selection of partners was regarded as the third or fourth step within formulation of online strategy for QuickBooks. By 2001, selection of partners' applications and selection had risen to be the first step in overall QuickBooks Business and Product Development Strategy.

Exhibit 8.6 Vignette - Intuit QuickBooks “QB Economy” Concept (Cont’d)

The Development Cycle of QuickBooks 2002 highlights the increasing significance of third-party applications, partners and services. By the mid-1990s Intuit had decreed a clearly defined twelve-month Product Development Cycle commencing in October- November each year. Major Product features for QuickBooks plus targeted ISV applications and Partner Services are incorporated into the annual Product Development Cycle. QuickBooks’ “Off-Cycle” Sub-Releases contain product patches and updates, plus specific ISV porting versions, and may be released in approximately four monthly increments.

1. September-October 2001:
    - Planning for new product release
    - Define services and features for new product release
    - Identify third-party applications and services for new release and commence negotiations with them
    - Pick up deferred features and services from previous release
2. November 2001 – February 2002
    - Third-party application and service partners signed up
    - “Off-cycle” QuickBooks sub-release for developers (every four months – about Jan 2002)
    - SDKs and service requirements finalised
3. March-April 2002
    - Final assessment of features, applications and services against “QB Economy” criteria
    - Internal coding for QuickBooks commences
    - Coding commences in ISVs with heavy application integration to QuickBooks application
4. May-August 2002
    - “Off-Cycle” QuickBooks sub-release for developers (every four months – About May 2002)
    - Coding complete and frozen for QuickBooks product by June 2002
    - Coding continues with QuickBooks ISVs applications including client/server synchronisation with QuickBooks application

Exhibit 8.6 Vignette - Intuit QuickBooks “QB Economy” Concept (Cont’d)

5. September-November 2002

- “Off-Cycle” QuickBooks sub-release for developers (Every four months – About September 2002)
- QuickBooks 2003 announced (September/October 2002)
- Late materials, applications and services added (October 2002)
- Any outstanding applications and services as at October 2002, deferred to next product release
- QuickBooks 2003 released and shipped in November 2002
- Planning commences for QuickBooks 2004

This vignette was prepared from: Hicks (2002a & 2002b).

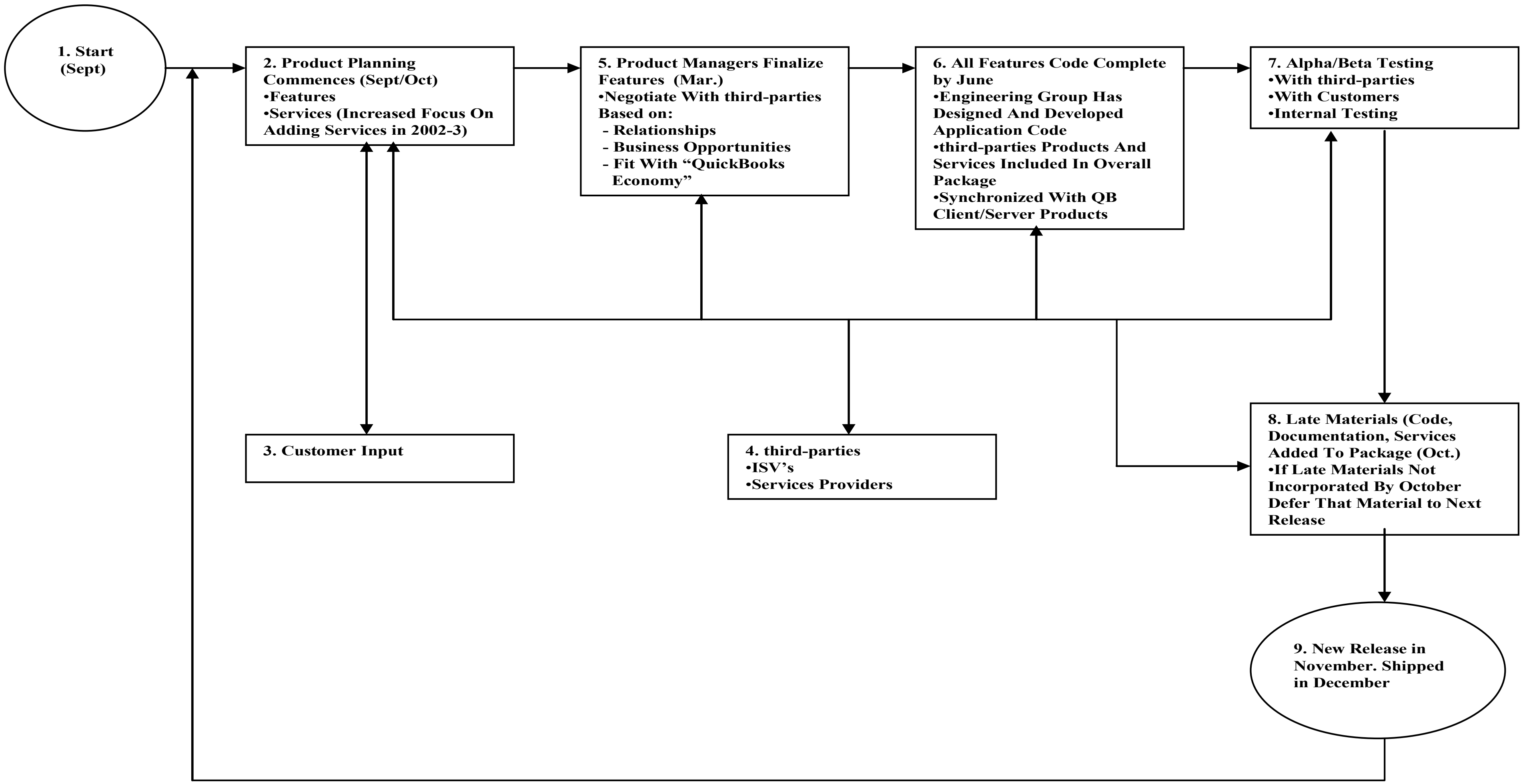


Figure 8.4: New Summary DSA Model For Intuit QuickBooks Application Development (2001-03)
Created from data in Kochikar & Lal (2000); Villanueva (2002); Hicks (2002).

Planning for QuickBooks 2002 commenced in September 2001, about one month earlier than previous versions. Third-party-Party ISVs and Services Providers feature in all aspects of the development cycle (box 4). From September through to March, Product Managers define and negotiate third-party applications and services for the new QuickBooks version (box 5).

The selection of third-party applications and services is based on the type and length of the relationships, potential business opportunities, and fit with the "QuickBooks Economy" concept. These criteria will be further explored in a new cognitive map addressing the "QuickBooks Economy".

By June all coding for all features must be completed to the extent that the application code for the new version of QuickBooks has been designed and developed to the point where alpha/beta testing can commence (box 6).

Third-party party applications and services must be included in the overall package to the stage where they are directly synchronised with QuickBooks' client/server applications. Normally, no third-party applications or services would be added beyond this stage of the development cycle.

Alpha/beta Testing is conducted from June to just prior to product release (box 7). "Customer" testing for a consumer (or small-business) application such as QuickBooks means that a subset of interested customers may try out a beta version of the new application and provide feedback.

Third-parties such as ISVs and developers who incorporate QuickBooks applications into their solutions are also likely to be actively involved in testing the application, with some involved in a "parallel" or "mutual" beta-testing process where they are acting as beta testers for QuickBooks while beta-testing their own applications themselves and with their customers.

QuickBooks' strict requirement for closing off development in October is maintained in the new Application Development DSA model (box 8). Training and preparation for the marketing launch of the new version is also undertaken in October.

The new version of Intuit QuickBooks is launched in November and typically ships in December (box 9). There is an overlap period between the release of the new version of QuickBooks and planning for the next version between September and December of each year.

The new DSA model for development in 2001-03 reflects high the involvement of third parties all the way through the development process. Intuit QuickBooks' management must factor in decisions related to third-party applications and services at the commencement of application planning.

A large proportion of Intuit QuickBooks management time and decision-making during October to March of each application development cycle is allocated to negotiating with third-parties for preferred applications and services. Programs to encourage third-party involvement with Intuit QuickBooks are a major priority for the QuickBooks management team and for Intuit's development executives.

8.8 Events Chronology Map

Kochikar & Lal's (2002) account of Intuit QuickBooks' history provides brief comments on events from Intuit's creation in 1983 through to 1998, as the focus of their case study is on Intuit QuickBooks' efforts to transition into an online environment during 1998-999.

An events chronology map covering the period 1983 to 1990 was created (see Figure 8.5), based on Kochikar & Lal (2000). Additional symbols were employed for the maps including the flowchart symbol for a delay or problem and a six-pronged box for a solution to the delay or problem.

Scott Cook and Tom Proulx set up Intuit in 1983 with a focus on an emerging niche in the financial software industry – providing a PC-based accounting package that could be used by individuals (and by extension for some small businesses – see box 1).

QuickBooks was launched in 1992, specifically packaged for small business owners (box 2). Intuit grew rapidly through the 1990s, developing formal organisational structures for sales, marketing and product development.

Intuit's development processes followed strict quality controls and product lifecycles (box 3). Intuit's structures were similar to established "B2C" software houses developing and marketing applications directly for consumers.

Intuit provided avenues for online purchasing in the late 1990s, but these were not part of any formal strategy until 1998. Nevertheless, 5% of all QuickBooks sales were through the Internet by 1998 (box 4).

The QuickBooks division of Intuit needed to clearly articulate a strategy for not only selling QuickBooks online, but also having an online presence for the division.

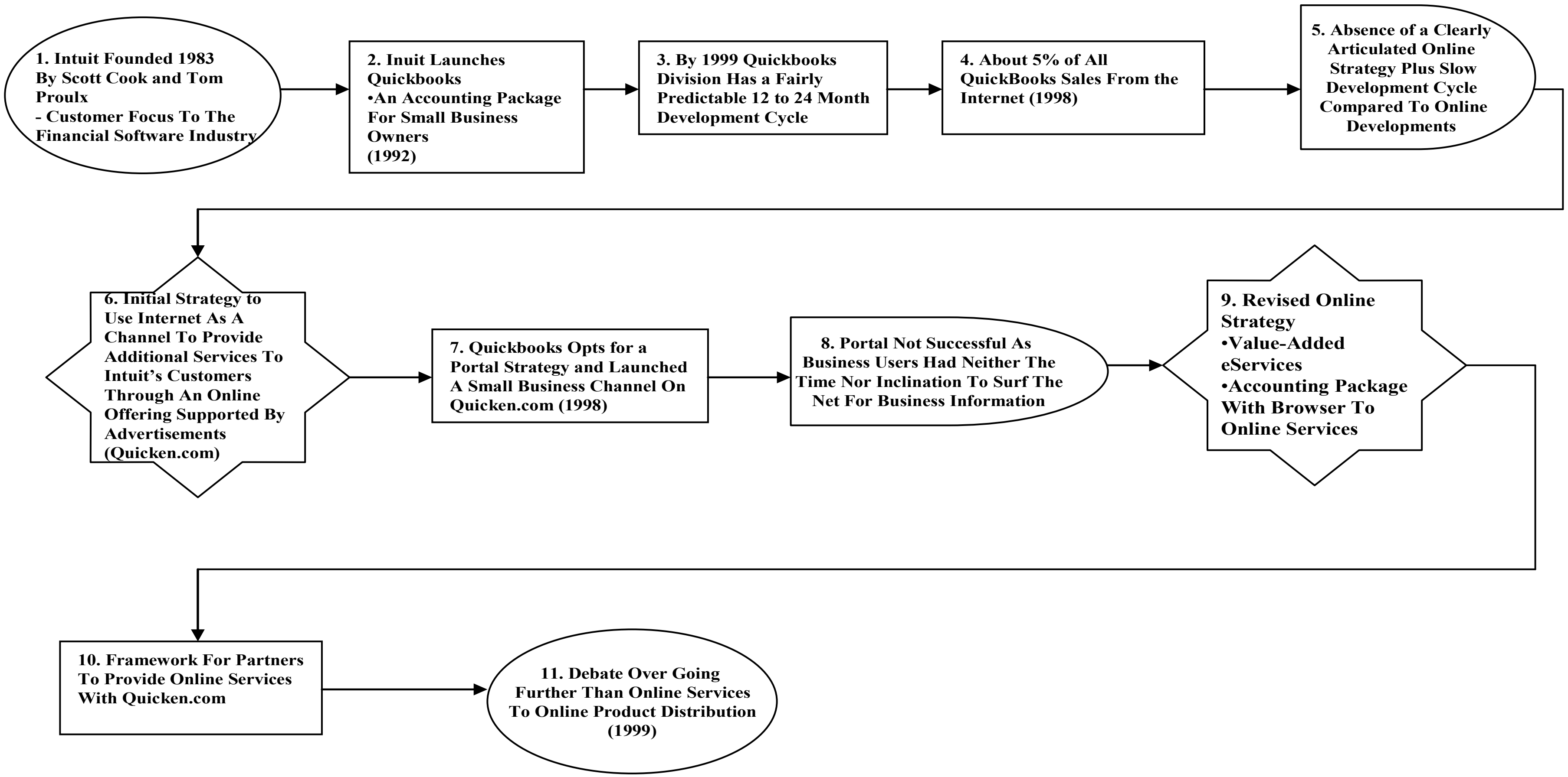


Figure 8.5: Events Chronology Map – Intuit QuickBooks
Created from data in Kochikar & Lal (2001); Villanueva (2002); Hicks (2002)

However, in order to rapidly move into an online environment, the QuickBooks division would also have to persuade Intuit to relax its strict and long development cycles (box 5).

The initial online strategy as devised by Nye and Villanueva was to set up a channel within the Quicken.com website, to provide additional services to Intuit's customers (not just Quickbooks or Quicken customers – see box 6).

Scott Cook was enthusiastic about setting up a business model for an online presence similar to that used by commercial television and radio, whereby Intuit's services and offerings would be underwritten by advertising revenues.

A common approach for development of online environment in the late 1990s was a portal strategy through which services could be packaged through channels within a website. The QuickBooks division developed a Small Business Channel to operate on the Quicken.com website in 1998 (box 7).

The Small Business Channel on Quicken.com was not successful, as QuickBooks discovered that small business users at that time were not substantial users of the World-Wide Web. Also, in circumstances where small business users had Web access, they did not have the time or inclination to surf it for business information (box 8).

The QuickBooks division believed that a revised online strategy focused specifically on providing value-added services for small businesses was required (box 9).

In retrospect, the QuickBooks application should have included an extension on its menu to enable users to go out to the Web to at least Intuit's website and probably out to other services. This functionality should have been present as early as 1995 or 1996 in the application – but that was not unusual for software houses that were not "born in the Internet".

Nevertheless, the QuickBooks division realised that there was a lot more to do than just add links to the Web, and expanded its decision-making toward working with third-parties to provide new functionality and services.

Nye and Villanueva developed a four-stage framework for qualifying third-party application and service providers, to provide services through the existing Small Business Channel on Quicken.com (box 10).

Kochikar & Lal's (2000) case rounds out the revised Online Strategy with discussion on whether QuickBooks should expand its services into actual online distribution in 1999 (box 11).

During the mid-1990s claims were made that the Web would revolutionise distribution of software (and information encapsulated in software files) as applications would be transferred and even sold through downloading copies of applications from Websites.

In practice low-speed bandwidth to most users reduced this capability down to the distribution of small-sized files, usually less than 5 MB. Web-browsers, plus some application patches and updates could be downloaded. Full applications were too large to download and delivery on CD-ROM was the preferred mode of distribution.

The bandwidth situation was relevant for small-businesses who, in the late 1990s were mainly connected to the Web via telephone modems with some instances of slow ISDN connections. In fact, consumers with cable TV/Internet connections were able to access broadband speed much earlier than small businesses in the late 1990s and the early years of the current decade.

Using CD-ROMs as a mode for application distribution remains strong to the present day, although increasing uptake of broadband connections for small business has swung the pendulum toward distribution by direct download – a trend likely to continue upward for the next three to five years.

Over a two year period, the QuickBooks Division appears to have rapidly increased the speed and scope of its decision-making when compared with its 20-year history.

Within the context of moving the QuickBooks business online, there was dispensation from Intuit's management to experiment with using a new business model, and then to refine that model for a specific group of customers (small business users). Key aspects of this experimentation will be explored through selected cognitive maps.

8.8.1 Updated Events Chronology Map

The events chronology map was updated after the interview stage to cover an extended timeframe of 1983 to 2002. There were significant revisions and corrections in the updated events chronology map, presented in Figure 8.6.

Additional insight into the early development of Intuit's first application, Quicken, was incorporated in the map, which underlines the company's focus on consumer-oriented financial management packages (boxes 1, 2 and 3).

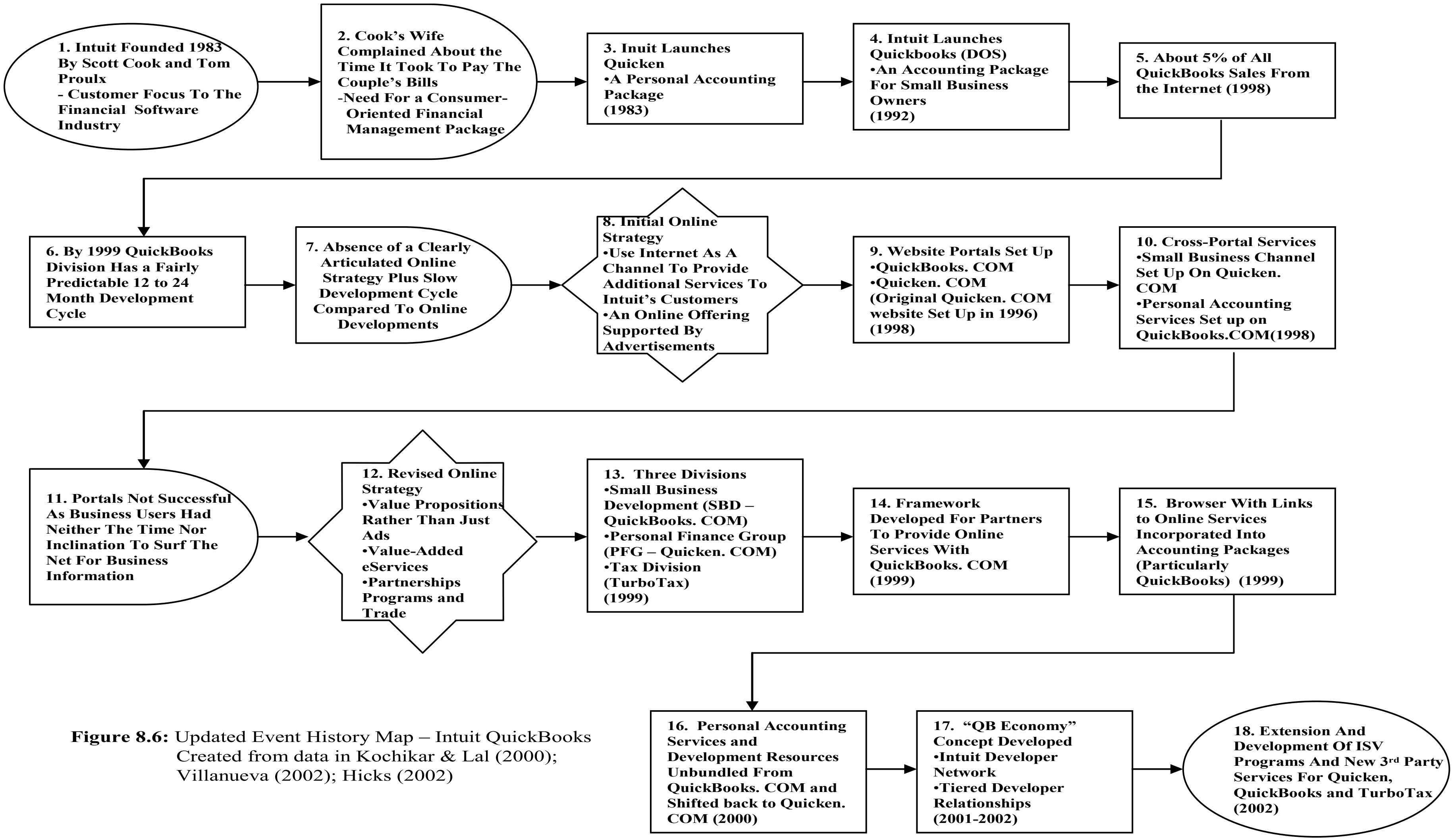


Figure 8.6: Updated Event History Map – Intuit QuickBooks Created from data in Kochikar & Lal (2000); Villanueva (2002); Hicks (2002)

Sections covering events from 1992 through the development of an initial online strategy remain unchanged (boxes 4, 5, 6, 7, 8). However, the initial online strategy is revised to indicate that the Quicken.com website was set up in 1996 and two separate websites, QuickBooks.com and Quicken.com were set up in 1998 (box 9).

“Cross-Portal” services were offered, that is, Small Business Channel on Quicken.com and Personal Accounting Services on QuickBooks.com (box 10), indicating some coordination of services across the QuickBooks and Quicken Divisions of Intuit.

Reasons for the portals not being successful remain unchanged (box 11), but there is further insight into the revised online strategy (box 12). Villanueva and Nye wanted more than just advertising and promotions online, and focused on value propositions. The propositions were in the form of value-added eServices developed through partnership programs and trading arrangements.

In 1999, Intuit reorganised into three divisions (box 13). The Small Business Development (SBD) Division runs the QuickBooks product line plus QuickBooks.com. The Personal Finance Group (PFG) manages the Quicken product line and its online presence through Quicken.com. The Tax Division runs the TurboTax product line and its online service, TurboTax.com. Each division is linked to Intuit’s corporate website Intuit.com.

Villanueva and Nye devised the partnership qualification framework for QuickBooks.com and the SBD Division in 1999 (box 14). This framework is explored further through cognitive maps in this case study.

The bundling of a Web-browser with each major Intuit product was introduced in 1999. Online services were tailored and linked to each major Intuit product. This initiative appears to be a significant catalyst for Intuit as a corporation to transition toward focusing on third-party developers and service providers.

In 2000, cross-portal personal accounting services and development associated with that activity were shifted out of SBD back to the PFG (box 16). At that time application development was focused back into each Division.

However, by 2001, a third-generation online strategy emerged with both a corporate and divisional contribution. Intuit established the Intuit Developer Network with a range of tiered Developer and Service Provider relationships and programs. The Intuit Developer Network was initially set up and trialled in the SBD with QuickBooks as part of an umbrella “QB Economy” concept (box 17).

During 2002, the QB Economy concept plus the Intuit Developer Network approach was expanded to the PFG and Tax Divisions (box 18). Intuit at the corporate level is active in extending existing and developing new ISV programs, which are then tailored for Intuit's divisions and product lines.

The revised events chronology map provides substantially increased insight with a clearer understanding of the development of online applications and supporting services for Intuit as a corporation and its main products moving into online environments.

Extension of the map to 2002 unearths further insights into online strategy development, indeed to a point where the emergence of a third-generation online strategy can be discerned.

Several issues emerging from the original and updated events chronology maps are explored further though cognitive maps.

8.9 Cognitive Maps

Cognitive maps created for the Kana case study were designed to complement DSA models and events chronology maps through additional exploration of selected decision-making issues and contexts.

All cognitive maps were created using the following procedures:

1. Selected extracts from Kochikar & Lal's (2000) case study were extracted to text files
2. One text file version of each extract was processed through the TACT package to produce a complete word-frequency list
3. Word frequencies were reduced through one or two iterations to ranked lists of words with frequencies ranging from over one to four words
4. A second version of the text file was input to NVivo
5. Ranked words were input to NVivo and coded as nodes in association with the text file
6. The overall extract was deconstructed into selected phases and sentences and coded directly as nodes through the InVivo tool within NVivo
7. The nodes associated with each extract were input to the Model Explorer and mapped using links incorporating codes of association plus addition node "items"
8. Several iterations of dynamic mapping were undertaken with the final cognitive map exported from NVivo to PowerPoint

9. Each PowerPoint version of the cognitive map was reformatted into a more readable and project-consistent format
10. The revised PowerPoint cognitive maps were presented to interviewees and validated or modified
11. The updated PowerPoint cognitive maps incorporated changes after the interview process.

Three sections were extracted from Kochikar & Lal's (2000) case study for further text analysis and cognitive mapping to complement the analysis presented in the original DSA model and original events chronology map.

The three initial cognitive maps created for the Intuit QuickBooks case study addressed the following themes:

8.9.1 Early Development (1992-99)
8.9.2 Initial Online Strategy (1998-99)
8.9.3 Revised Online Strategy (1999-2000)

During the interview process, a requirement was identified for an additional cognitive map reflecting updated insights from 2001-02:

8.9.4 "QB Economy" Concept (2001-02)

The first cognitive map is an extended insight into Intuit QuickBooks prior to formalising an online strategy. The second and third cognitive maps are also sequential snapshots reflecting two changes in online strategy. The additional cognitive map highlights insights from what is effectively a third online strategy as executed in 2001-02.

8.9.1 Cognitive Map 1 – Early Development (1992-99)

Development details for the first cognitive map include:

- Extract section, size and exhibit:
    - "The QuickBooks Story" section, Kochikar & Lal (2000, pp. 3-4); 1 page; Exhibit 8.7
- TACT word frequency lists and comments on key words:
    - First pass: 27 words with frequency > 2
    - Second & final pass: 18 words with frequency > 3
    - Final ranked word list: Table 8.1
    - There was a diverse range of words in the final list with the top words being related to QuickBooks and Intuit, "product", "develop"

"customer", and "channel" – providing indicators on key themes in the extract on Intuit QuickBooks' history, product development and channels of distribution.

- NVivo ranked word, sentence and phrase node count = 136
- Association of concepts coding used: +, -. +/-
- NVivo PowerPoint cognitive map: Figure 8.7
- Reformatted PowerPoint cognitive map: Figure 8.8

The NVivo cognitive map covers insights into the development and launch of the QuickBooks product in 1992, and goes forward to the late 1990s highlighting QuickBooks' product features, Intuit's product development process, and QuickBooks' channels of distribution.

Exhibit 8.7 Text Extract – Intuit QuickBooks – Early Development (1992-99).

The QuickBooks Story

In 1992, Intuit launched QuickBooks, an accounting package for small business owners. The impetus for the creation of QuickBooks came from a surprising source: the sales data for Quicken indicated that a growing number of small businesses purchased the product even though it was not built for the small-business segment. Subsequent market research confirmed that there was an unquestionable need for an easy-to-use accounting package within the small-business segment. The existing accounting packages were complex and required extensive knowledge of double-entry accounting systems.

501-054 Intuit QuickBooks

Like all other Intuit products, QuickBooks was built to be a consumer product. Individuals with basic computer skills and no accounting knowledge could both install and use the product. QuickBooks converted familiar tasks such as writing checks and filling invoices and order entry forms into accounting entries. QuickBooks also analysed these entries and generated reports that tracked key business drivers. For instance, a small-business owner could generate reports to find out which customers or products were most profitable. Additionally, QuickBooks could be used to track time, cost jobs, and develop more accurate estimates for future tasks.

Exhibit 8.7 Text Extract – Intuit QuickBooks – Early Development (1992-99) (Cont'd).

The QuickBooks Division adhered to Intuit's stringent product development, quality assurance, and customer service guidelines. Although the development process was complex, over time it had evolved into a science. By 1999, the process was fairly predictable and had a structured cycle of about 12 to 24 months. Once new products were released, they generally were not modified (except to fix bugs) until the next scheduled product upgrade. A rigorous, comprehensive quality assurance process was integrated deeply into the development process, minimizing bugs and optimizing the development cycle.

Retail distribution channels such as software outlets, computer superstores, office and warehouse clubs, and general mass merchandisers were used to provide visibility and accessibility at places where small business owners shopped. Original equipment manufacturers (OEMs) were used strategically to source new customers even though their discounted pricing structure made them an insignificant revenue source.

Direct distribution channels such as direct-mail, telephone solicitations, direct-response newspaper and magazine advertising, and television and radio advertising were used to encourage existing customers to purchase software upgrades and related products and services such as office supplies and checks. Intuit's customer database became a valuable tool for drawing additional revenues through cross-selling products and services through this channel. Company sales records showed that almost 50% of the registered Quicken and QuickBooks customers had also purchased other products and services from Intuit, making direct sales a significant portion of total desktop software revenue.

Exhibit 8.7 Text Extract – Intuit QuickBooks – Early Development (1992-99) (Cont'd).

Electronic ordering and delivery - with the advent of the Internet, electronic ordering and delivery became both a convenient and cost-effective means of distributing QuickBooks to the customers. Sales through this channel grew steadily and amounted to about 5% of all sales in 1998. This clearly indicated that an increasing number of QuickBooks customers had access to the Internet and were willing to purchase products and services online.

Well-oiled development processes and distribution channels made QuickBooks a huge success, with nearly 50% growth in revenues each year. By 1998, QuickBooks owned 85% share of retail sales in its segment, delivering nearly 20% of Intuit's total revenues.

Source: Kochikar & Lal 2000, pp. 3-4.

Table 8.1 Intuit QuickBooks – Early Development (1992-99) – Ranked Word List

Anchor Concepts	Frequency
quickbooks	15
product	12
customer	8
intuit	7
use	6
sales	6
develop	6
channel	6
accounting	6
service	5
revenue	5
process	5
distribute	5
direct	5
business	5
such	4
small	4
purchase	4

Source: data analysis performed using TACT (1997)

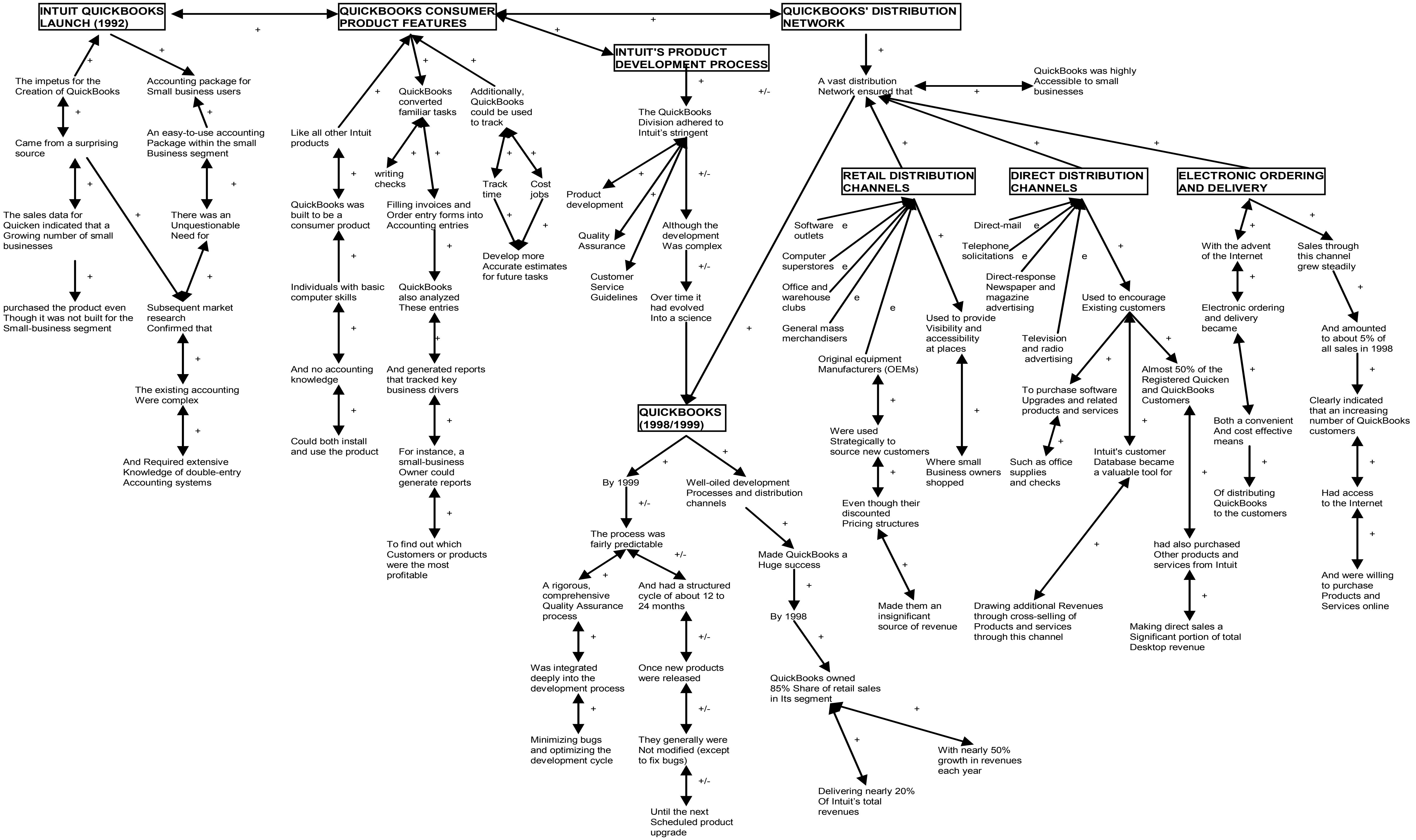


Figure 8.7 First Cognitive Map (NVivo Model) – Intuit QuickBooks – Early Development (1992-99)
Created from data in extract (Exhibit 8.7)

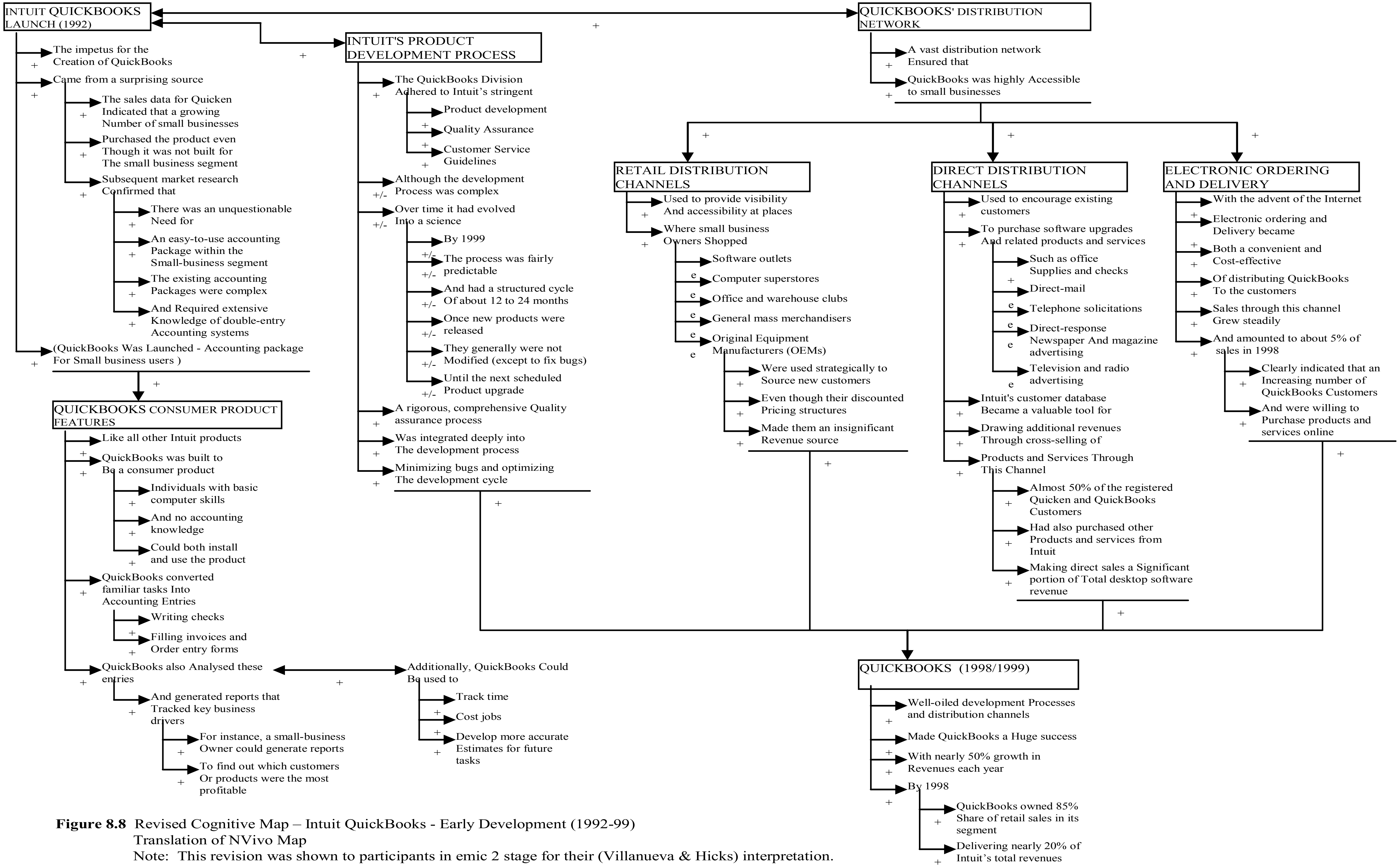


Figure 8.8 Revised Cognitive Map – Intuit QuickBooks - Early Development (1992-99)
Translation of NVivo Map
Note: This revision was shown to participants in emic 2 stage for their (Villanueva & Hicks) interpretation.

In the early 1990s, Intuit's main application was Quicken, which was geared for the personal financial management market. However, the application indirectly extended to small business financial management, evidenced through a growing number of sales to small business owners.

Quicken was easy to use, while existing accounting and financial management packages designed for SMEs were complex and required extensive accounting knowledge to use them.

Intuit addressed this unmet need for an easy to use product for the small-business market through the launch of QuickBooks in 1992. QuickBooks was designed to be a consumer product, to be sold through the same channels of distribution as Quicken. It was also designed for small business managers with basic computer skills and no specialised accounting knowledge.

QuickBooks could handle basic business transaction recording and activities such as writing cheques, preparing and producing invoices and order entry forms.

QuickBooks could go further than basic accounting activities to generate business reports and track the most profitable customers or products. QuickBooks could also provide some basic cost management features, such as costing jobs and tracking times for jobs and set up estimates for future tasks.

Over the next 5-6 years Intuits' established strict product development processes and refined its distribution network. These built on standards developed by Intuit for Quicken from 1983 to 1992.

Stringent standards for product development, quality assurance and customer service were required to produce and deliver high-quality products for consumers with basic computing skills and minimal accounting knowledge. Intuit's accumulated experience enabled them to organise product development into 12 to 42-month cycles, with strict rules on not modifying or adding features until subsequent releases.

One significant factor enabling Intuit to maintain such high development standards was that the QuickBooks application used minimal third-party applications. The rigorous and comprehensive quality assurance process could be heavily monitored directly into Intuit's own development teams.

In the late 1990s, QuickBooks distribution network was tuned toward reaching small business customers through three main channels; including retail distribution, direct distribution, electronic ordering and delivery.

Retail distribution channels for QuickBooks were an extension of Intuit's Quicken channels using software outlets, computer superstores, relevant office and warehouse clubs and general mass merchandisers – all channels where small business owners shopped.

QuickBooks supplied its product to PC and small computer system vendors in OEM relationship arrangements. These relationships provided strong strategic visibility for Intuit and its QuickBooks product when small business owners purchased PCs or small servers for their operations.

The direct distribution channels were also a logical extension of those used for the Quicken product. It seemed that Intuit's customer database was expanded to accommodate QuickBooks' customers, and several basic direct marketing programs were run but tailored differently for Quicken and QuickBooks customers.

Indeed, as almost 50% of registered Quicken and QuickBooks customers had purchased other Intuit products (most likely Quicken or QuickBooks or TurboTax), cross-selling to existing Intuit customers was a strategic direct-selling activity.

This channel was used for selling to existing QuickBooks and Intuit customers rather than for outbound direct marketing to prospective customers. Direct marketing programs were coordinated through various avenues including direct-mail, telemarketing, direct response newspaper and magazine advertising, and TV and radio advertising.

Electronic ordering and delivery had commenced for mainly Quicken products in 1996, but within two years, 5% of QuickBooks sales could be attributed to customers ordering the product via the Internet (but not downloading through it). E-commerce functionality enabled convenient, fast and cost-effective ordering of QuickBooks, but slow bandwidth severely held back using the Internet for product delivery in 1998-99.

However, a strong growth in online ordering indicated that an increasing number of small business customers were becoming more proficient at using the Internet for business activity and were confident enough to order, and in some cases receive the QuickBooks product through the Internet.

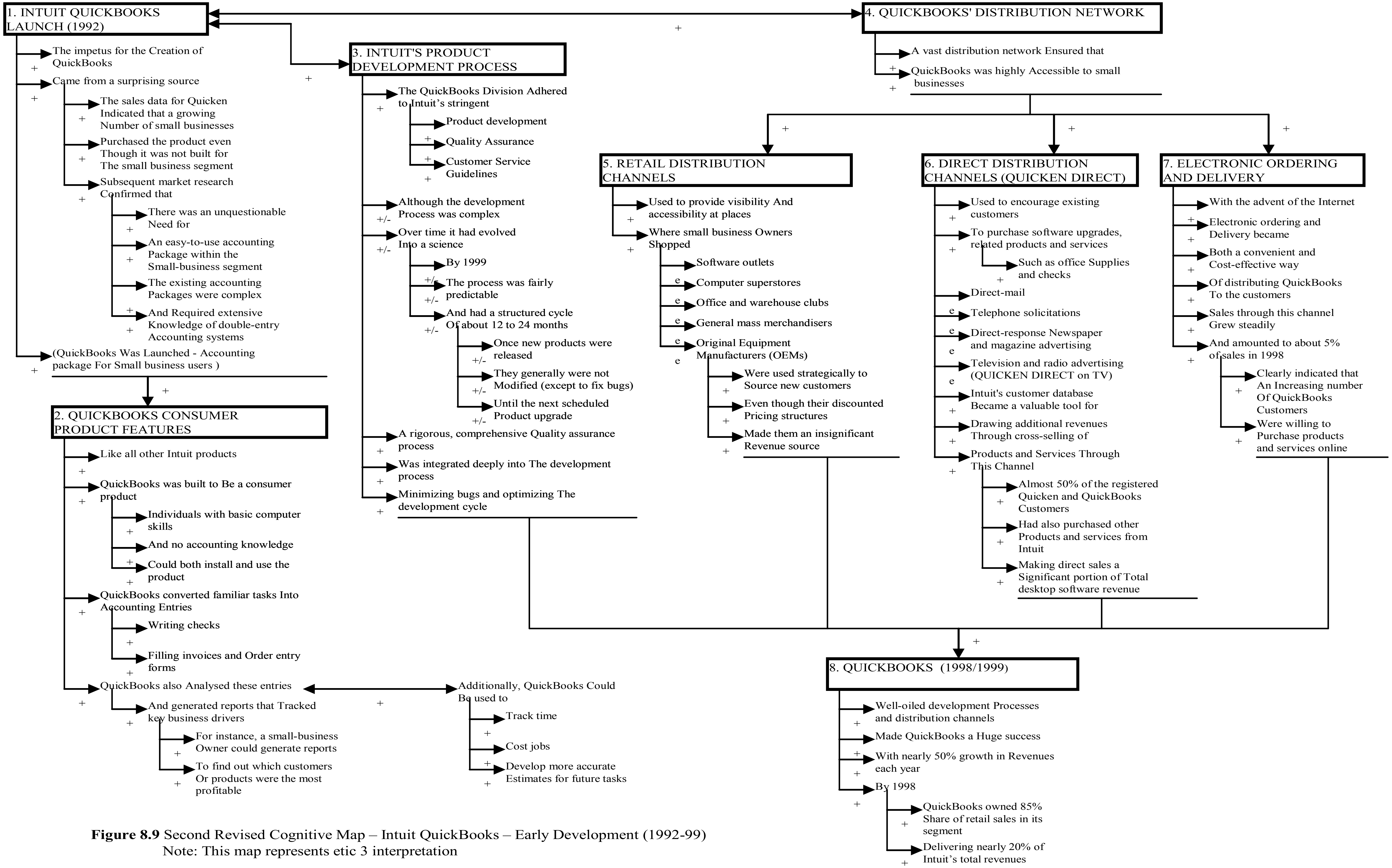


Figure 8.9 Second Revised Cognitive Map – Intuit QuickBooks – Early Development (1992-99)
Note: This map represents etic 3 interpretation

By late 1998 – early 1999, established development processes and distribution channels had set up the QuickBooks Division with a market share in its segment for the QuickBooks product of 85%, yet with revenue growing at 50% each year, and the Division delivering nearly 20% of Intuit's total revenues.

Prior to widespread acceptance of the Internet by small business (that is, from about 1994 to 2000), Intuit QuickBooks had established a strong market position.

8.9.1.1 Updated Cognitive Map 1 – Early Development (1992-99)

The cognitive map was revised through direct editing in PowerPoint with inputs from interviews with Villanueva (2002) and Hicks (2002). Overall, the original map was validated with some only minor updates. The updated map is presented in Figure 8.9.

Sections covering Intuit QuickBooks launch (box 1), QuickBooks' consumer product features (box 2), and Intuit's product development process (box 3) were validated without changes (except for a minor indentation of points in for the development process).

Channels of distribution for retail distribution (box 5) and electronic ordering and delivery (box 7) were validated without modifications.

The direct distribution channels Section (box 6) was modified to highlight the channel as being for Quicken Direct, rather than specifically for QuickBooks. According to Villanueva (2002), this was the correct context for the account given for the HBS case study by Kochikar & Lal (2000).

There was a minor indentation of points related to purchasing software upgrades and related products and services. A point was added noting a special Quicken Direct promotion on television for the Quicken product, which could also be extended to the QuickBooks product.

QuickBooks (1998-99)(see box 8) was validated without modifications.

Overall, the updated cognitive map was validated, except that Intuit's direct distribution channels in 1998-99 were really focused on Quicken and not QuickBooks, and this was also probably true for electronic ordering and delivery at that time.

8.9.2 Cognitive Map 2 – Initial Online Strategy (1998-99)

Development details for the second cognitive map include:

- Extract section, size and exhibit:
    - "Initial Online Strategy" section, Kochikar & Lal (2000, pp. 4-5); 1 ¼ page; Exhibit 8.8
- TACT word frequency lists and comments on key words:

    - First pass: 27 words with frequency > 2
    - Second & final pass: 18 words with frequency > 3
    - Final ranked word list: Table 8.2
    - Most of the highly ranked words highlighted an emphasis on Intuit *QuickBooks* to *develop* a *business strategy* for using the Internet to *deliver* a *service* to *customers* (words in italics were ranked highly in the ranking list).
- NVivo ranked word, sentence and phrase node count = 144
- Association of concepts coding used: +, -. +/-
- NVivo PowerPoint cognitive map: Figure 8.10
- Reformatted PowerPoint cognitive map: Figure 8.11

The cognitive map provides a cognitive description of Intuit's logic for looking at the Internet as a new channel and the development of an online strategy, with emphasis on developing customer-based solutions including value-added services.

Exhibit 8.8 Text Extract – Intuit QuickBooks – Initial Online Strategy

Initial Online Strategy

The emergence of the Internet offered Intuit a new channel to its customers. Scott Cook envisioned using this channel to provide additional services to Intuit's customers through an online offering supported by advertisements. "Ads when done right can be a very powerful revenue source. If we aggregate our customers online, we would be the chosen Internet site for all companies that want to reach our installed base."

Intuit QuickBooks 501-054

Cook believed that ad-supported Internet would trigger explosive growth in the software industry, just as ad-supported television triggered explosive growth in the media industry. "Advertisements made it possible for [the ordinary folks] to have access to extraordinarily produced, good quality programs for free (through public television) by creating two kinds of customers for the media industry: the audience (users) and the advertisers (payers).

Exhibit 8.8 Text Extract – Intuit QuickBooks – Initial Online Strategy (Cont'd)

Advertisers became the primary revenue source." Drawing on this parallel, Cook was excited about the potential of financing some of the development costs of QuickBooks through ad revenues. "Today 100% of the cost of QuickBooks is borne by the customer. In the future, we can use ads (revenues) to lower the cost to the user. By creating highly affordable products, we could not only grow the market but also increase market share of our products. We could become the AOL of small businesses."

This vision and Quicken.com's success as an advertisement-supported financial service portal for individuals led QuickBooks to adopt a portal strategy. In spring of 1998, QuickBooks launched a small business channel on Quicken.com. Although ad inventory on this channel was sold out almost instantly, the portal strategy proved inappropriate for the small business segment. While individuals surfed Quicken.com actively for financial information and advice, small-business owners had neither the time nor the inclination to surf the net for business information. Villanueva acknowledged that the small-business channel on Quicken.com had a "tech-focus" and not a "customer-focus." "We were excited by the technical possibilities and had overlooked our customer needs."

In order to develop a customer-focused solution, QuickBooks Division created an internal taskforce to examine ways to use the Internet to serve small businesses. This taskforce recommended using the Internet to provide value-added services such as inventory management, employee benefits administration, customer relationship management, and online banking to small businesses. The findings also indicated that small businesses were willing to pay for these services. To act upon these findings and to develop a comprehensive online strategy, Nye created a Business Development organisation and recruited Villanueva to lead it.

Business development at QuickBooks: Villanueva's team was charged with the responsibility of determining the appropriate strategy for providing value-added services over the Internet: Should these services be built in-house or should they be acquired through partnerships? How should they be integrated into the QuickBooks product? How should these services be priced?

Exhibit 8.8 Text Extract – Intuit QuickBooks – Initial Online Strategy (Cont'd)

The formation of Villanueva's group suggested a significant digression from QuickBooks' traditional, customer-centric, make-incremental-improvements-to-a-successful-product strategy. Villanueva knew that the battle for scarce engineering resources would severely test the commitment of the organisation to enter the online market. To be successful, Villanueva's group had to respond to market trends and rapidly develop and deploy several new services. This not only required a change in the product-development mindset but also a willingness to make tradeoffs between upgrades and new features.

Opportunities to partner: Just as Villanueva was pondering over these issues, several companies approached QuickBooks with attractive partnership proposals. While some of these companies were young Internet start-ups eager to be associated with a strong brand, others were established players who were anxious to reach QuickBooks's installed base. These companies also provided a wide gamut of services, including online banking, online postal services, and online procurement. Although these opportunities were compelling, Villanueva had to resolve the partner or-not-to-partner debate within QuickBooks before he could draw on these opportunities.

Source: Kochikar & Lal 2000, pp. 4-5.

Table 8.2 Intuit QuickBooks – Initial Online Strategy – Ranked Word List

Anchor Concepts	Frequency
Quickbooks	12
customer	10
service	10
business	9
online	9
Villanueva	8
internet	7
develop	6
small	6
strategy	6
channel	5
companies	4
create	4
market	4
product	4
provide	4
revenue	4
should	4

Source: data analysis performed using TACT (1997)

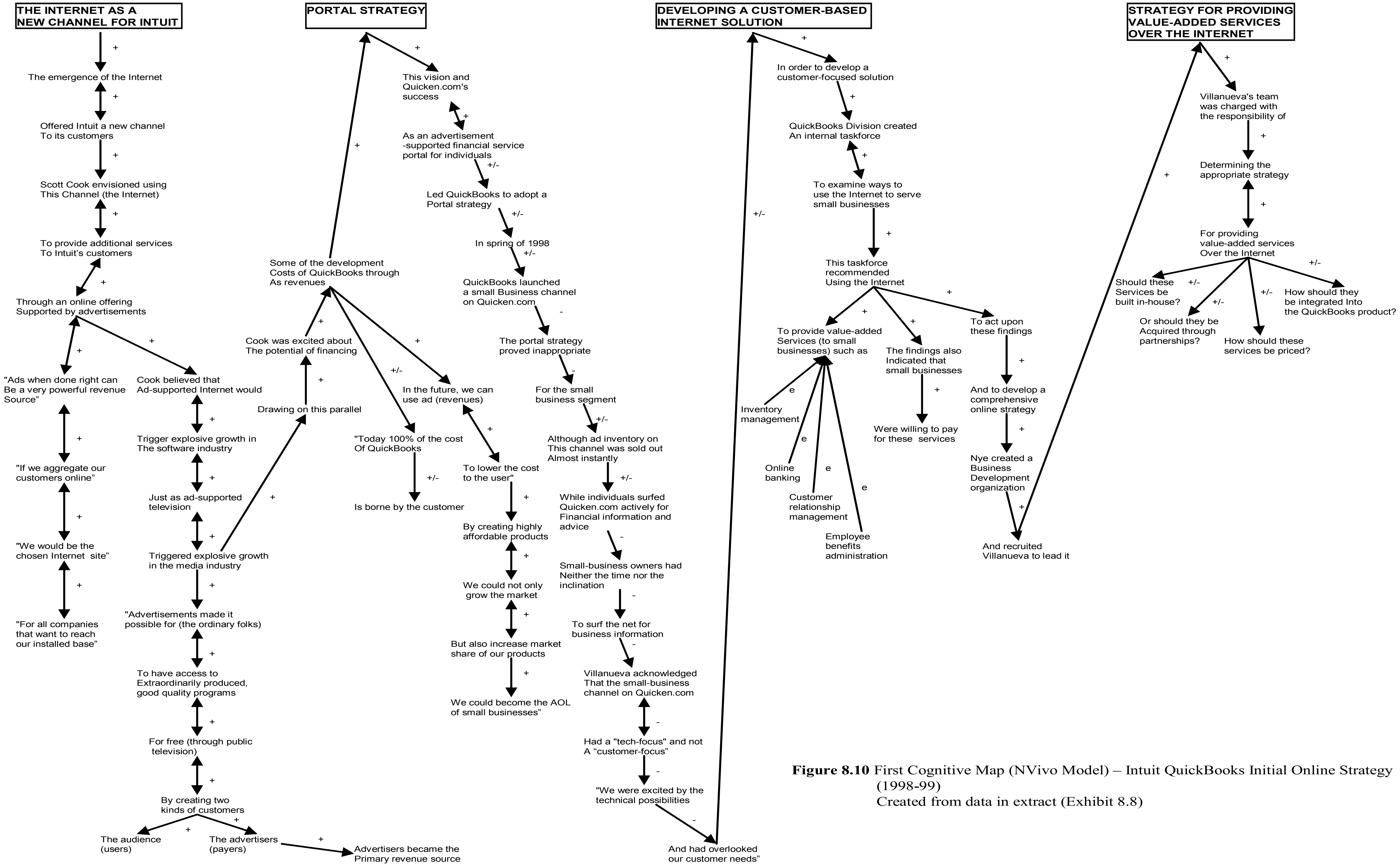


Figure 8.10 First Cognitive Map (NVivo Model) – Intuit QuickBooks Initial Online Strategy (1998-99)
Created from data in extract (Exhibit 8.8)

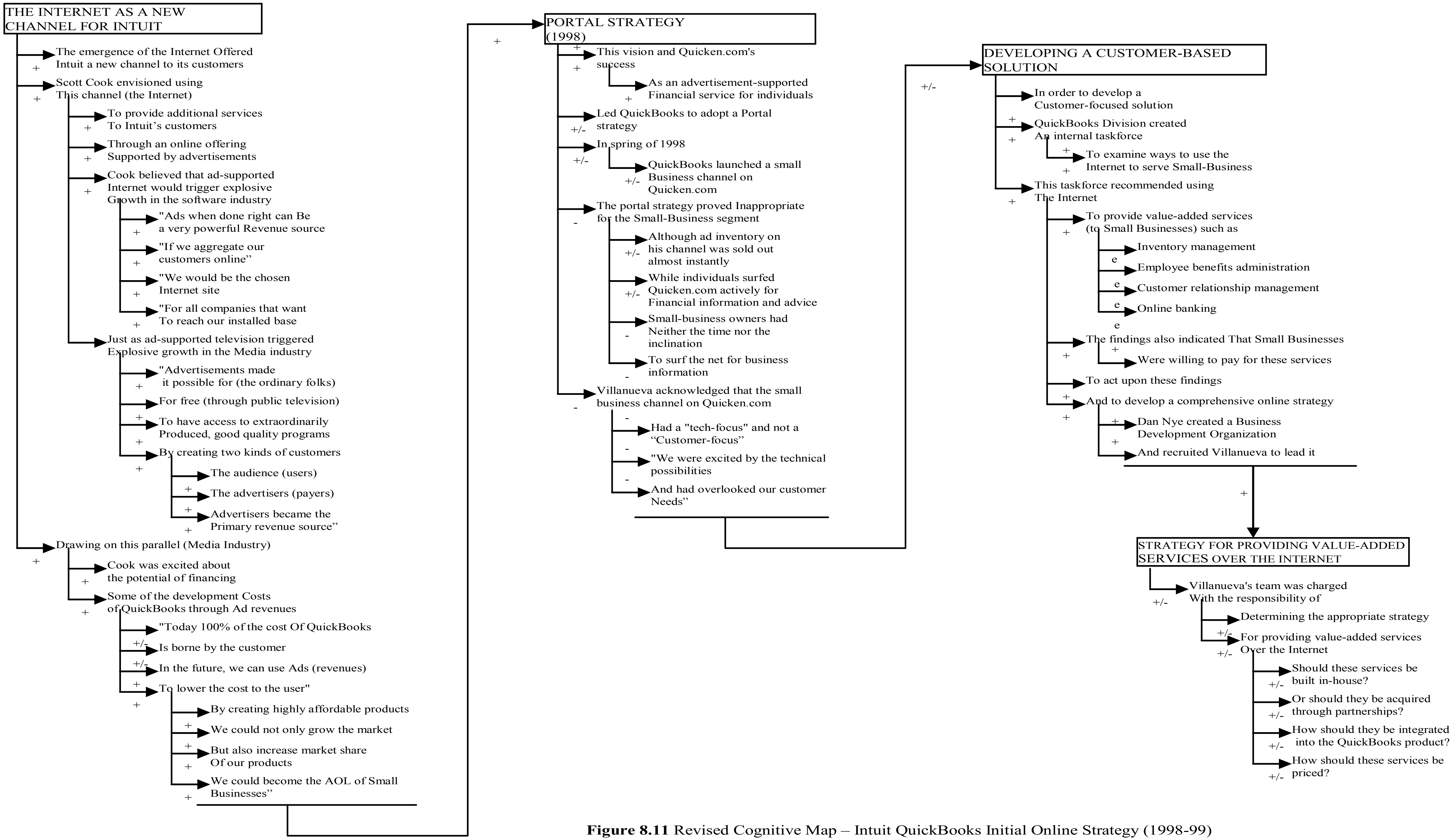


Figure 8.11 Revised Cognitive Map – Intuit QuickBooks Initial Online Strategy (1998-99)
Translation of NVivo Map
Note: This revision was shown to participants in emic 2 stage for their (Villanueva & Hicks) interpretation.

By 1998, Intuit's executive management could see the Internet as an emerging channel to its customers – but also a new business model for that channel.

Intuit's co-founder Scott Cook saw the Internet as an opportunity for Intuit to provide a range of additional services to its customers, underwritten by advertising.

Cook noted that the ad-supported television environment triggered explosive growth in the media industry, enabling viewers to see good quality programs for free, with advertisers being the primary revenue source for the industry. The approach, applied to ad-supported Internet, could trigger rapid growth for the software industry.

Cook could also see the Internet as an opportunity to create an aggregation of online customers under an "Intuit Online" umbrella.

In order to start the new approach, Cook (and his executive team) considered the potential for financing some the development costs of QuickBooks through online advertising revenues.

Initiating such a strategy would represent a transformation for Intuit, which was highly focused on internal product development with minimal interaction with third-party application and service providers. In this new online world the third-parties, as advertisers, would be the primary revenue source for Intuit.

Cook hoped that online advertising revenues could not only lower the cost of QuickBooks (and Quicken) to the customer, but that the new revenue streams could also be used to position Intuit as a portal for small businesses, similar to AOL being a portal for personal information provision.

Development of portals underwritten by advertising or referral revenues was a common online strategy in the late 1990s and was best exemplified by Yahoo!'s approach to developing a comprehensive range of personal information services at that time.

Intuit also believed that it had been successful in developing an advertisement-supported personal financial service through Quicken.com from 1996-98. Intuit could experiment with extending this expertise to developing an equivalent online small business service.

The QuickBooks Division adopted a portal strategy whereby it launched a small business channel on the Quicken.com website in the spring of 1998.

However, the portal strategy did not work quite as the QuickBooks division expected. Advertisers took up slots and placements quickly on the new channel, in anticipation of high hit-rates as surfers moved through Inuit's online offerings – and individuals surfed Quicken.com actively for financial information and advice.

What was not anticipated was that as a group, small business owners did not have the time or inclination to surf the World-Wide Web for business information. Furthermore, looking back at 1998, it is highly likely that most small business users would have possessed basic PC skills – which would have not included using the World-Wide Web for searching and retrieving business information.

Villanueva recognised that the small-business channel on Quicken.com was technically innovative but a long way ahead of small-business customers' capabilities and needs.

In order to develop an online customer-focused solution for the QuickBooks' product, the QuickBooks Division created an internal taskforce.

The taskforce came up with a number of recommendations for using the Internet with the QuickBooks product. Value-added services that could be provided through the Internet for small businesses included: inventory management, employee benefits administration, CRM, and online banking. The taskforce believed that small businesses were willing to pay these services.

In response to the taskforce's findings, Dan Nye created a Business Development Organisation with Villanueva to lead it. The new organisation was charged with determining the appropriate strategy for providing value-added services over the Internet.

Several issues needed to be addressed in formulating the strategy, many of which were quite radical steps for the QuickBooks Division. Decisions covering the development of services in-houses versus acquiring them through partnership would be critical in the revised strategy.

Furthermore, the level of integration of services into the QuickBooks products would require an additional set of decision-making dimensions for QuickBooks' executive management and supporting product and project management teams. In particular, new decision-making processes for the QuickBooks Division, covering pricing of the new services had to be established.

The map concludes by highlighting key decision-making questions in developing a comprehensive online strategy for the QuickBooks division.

8.9.2.1 Updated Cognitive Map 2 – Initial Online Strategy (1998-999)

The cognitive map was revised mainly for extra insights, subsequent to interviews with Villanueva (2002) and Hicks (2002), and is presented in Figure 8.12.

The first section of the map covering "The Internet as a new channel for Intuit" (box 1) was validated without modifications, providing a clear account of the ad-supported logic for Intuit moving into the Internet, as articulated by Scott Cook.

There were a number of additional insights and explanatory notes incorporated into the portal strategy section (box 2). Intuit's experience with online information management actually extended back to 1993, with the development of an online system for private banking using bulletin boards. This approach was not successful, probably because few customers were users of bulletin boards at that time.

Scott Cook's vision of an ad-supported Internet approach for Intuit seemed to be encouraged by the success with Quicken sales through the Quicken.com website, although that early success was not based on providing online financial services or providing advertising on the Quicken.com site.

In the spring of 1998, Intuit created a consumer finance portal strategy, but the QuickBooks Small Business Division (SBD) not only launched a small business channel on Quicken.com, but also set up a separate website for QuickBooks (QuickBooks.com).

Reasons for the portal strategy being inappropriate were not modified in the updated cognitive map. An example of financial information and advice that individuals sought while surfing Quicken.com was stock quotes.

Villanueva (2002) provided additional insights into the small business channel on Quicken.com: At the time "everyone thought we needed a portal" and "there was an overemphasis on Ad Revenue". These thoughts reinforce a common view around 1998-2000 on developing online business models using portals supported by online advertising revenues.

The section on "developing a customer-based solution (box 3) remained unchanged from the original map except that the timing for creation of the taskforces and reporting its findings was during the summer of 1999.

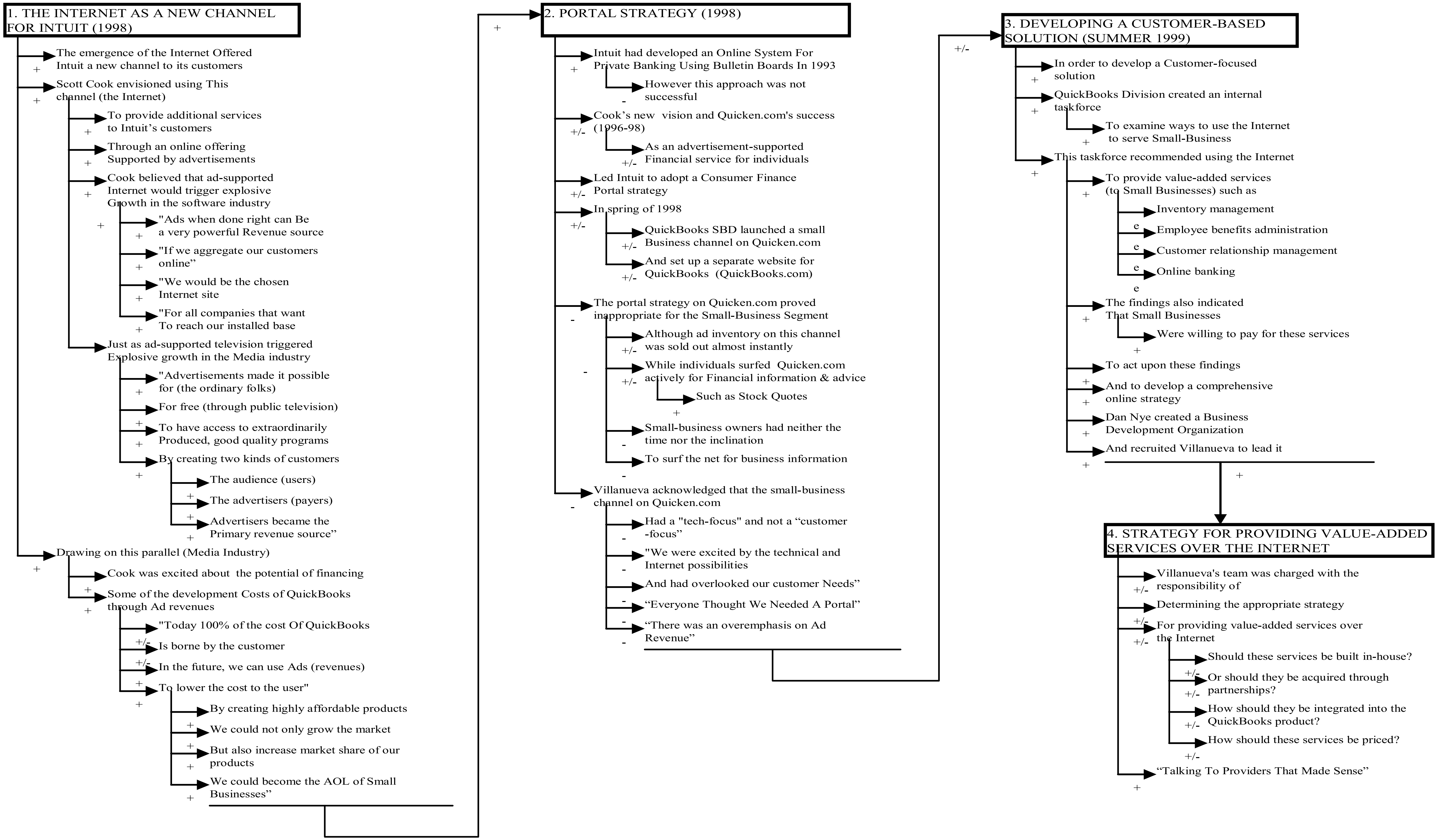


Figure 8.12 Second Revised Cognitive Map – Intuit Quickbooks Initial Online Strategy (1998-99)
Note: This map represents etic 3 interpretation

The section on "strategy for providing value-added services over the Internet" (box 4) was unchanged except for an additional insight from Villanueva. He indicated that the appropriate strategy was about "Talking to Providers That Made Sense" (Villanueva 2002). "Sense" here means that relationships between QuickBooks SBD and providers would logically complement or extend the QuickBooks product.

The revised cognitive map largely validates the original map with some extra insights specific to QuickBooks SBD.

8.9.3 Cognitive Map 3 – Revised Online Strategy (1999-2000)

Development details for the third cognitive map include:

- Extract section, size and exhibit:
    - "Revised Online Strategy" section, Kochikar & Lal (2000, pp. 7-9); 1 ½ page; Exhibit 8.9
- TACT word frequency lists and comments on key words:
    - First & final pass: 19 words with frequency > 4
    - Final ranked word list: Table 8.3
    - The ranked word list highlights themes relating to *QuickBooks build*ing *business* and *revenue* with *partners* and *partnerships* for *online services* (italicised words were ranked highly in the ranked word list).
- NVivo ranked word, sentence and phrase node count = 195
- Association of concepts coding used: +, -. +/-, e
- NVivo PowerPoint cognitive map: Figure 8.13
- Reformatted PowerPoint cognitive map: Figure 8.14

The map highlights Kana's channels of distribution with emphasis on business partners, plus further insights into Kana Online (KOL).

The map explores issues related to setting up and selecting partners for online applications and services to work with the QuickBooks product. A formal four step framework for partner assessment and selection is described and analysed in the map.

Exhibit 8.9 Text Extract – Intuit QuickBooks – Revised Online Strategy

Revised Online Strategy

Villanova envisioned a hybrid product that combined the desktop accounting package with a suite of value-added "eServices." To ensure seamless access to these online services, the product would include an Internet browser, which would be installed along with the accounting package. Users without Internet access could continue to use it as a standalone accounting package. However, those with Internet access could get the added benefit of valuable, small-business-focused, online services.

Both Nye and Villanueva acknowledged that most online services were immature and would likely fail QuickBooks's stringent quality standards tests. Balancing the need to get to market before the competition with the need to protect the QuickBooks brand would be their greatest challenge in formulating an effective online strategy. In order to overcome this challenge successfully, Nye developed a simple framework that would help him review the partnership proposals objectively.

Nye's framework: Nye's framework had four steps. First, a set of desired features would be prioritised based on their usefulness to small businesses and economic viability. "We are in the business of making small businesses successful," said Nye. "However, we cannot build everything that [small businesses] desire because we are constrained by the resources-engineering, financial, and so forth-that we have. To be successful, we need to build features that are both desirable and make economic sense. The services must be attractive enough to promote usage and build greater loyalty to the QuickBooks brand."

Second, the "build-vs.-buy" would be made using well-defined criteria. According to Nye, QuickBooks would benefit from buying services that met the following criteria:
Small businesses valued the service: Service belonged to a new and emerging space, and a high-quality player was willing to partner, QuickBooks benefited (had potential of deriving recurring revenues) from playing in that market, and QuickBooks would not lose valuable time and money trying to catch up with the frontrunners that had made significant headway in that market.

Exhibit 8.9 Text Extract – Intuit QuickBooks – Revised Online Strategy (Cont'd)

Third, the appropriate level of integration between the partner's products/ services and QuickBooks would be determined. "We can hedge our bets by taking a tiered approach to partnering and integration," reasoned Nye.

> If we believe in the value of the service, but are uncertain about the long-term viability of the partnership, we could build a "light" partnership, wherein we would only provide a referring link from our product to the service provider's site and not lend the QuickBooks name to the service. If we value the service and support the partner strongly, we could co-brand the referring link, lending the QuickBooks seal of approval to the service. For instance, we could say "online postage brought to you by QuickBooks." Only when we are convinced that the partnership provides superior, long-term benefits, and promotes upgrades and brand loyalty, would we build a "deep" partnership. Here, we would not only co-brand but also integrate the service seamlessly into QuickBooks.

Villanueva agreed with Nye that the branding and integration would be two of the important decision factors in finalizing partnership deals. However, he believed that the quality of the customer experience on the partner site would impact the QuickBooks brand even in a 'light' partnership. "Our users trust our brand and will assume that we recommend any service we link to from our product," said Villanueva, pointing out that stringent due-diligence would be required even for "light" relationships.

Finally, a list of potential partners would be created and the appropriate deal structure for each would be established. According to Nye, there were two possible revenue components to each deal: (1) Slotting Fee-an up-front payment made by the partner to QuickBooks to get access to small businesses, and (2) Revenue Share-a cut in the ongoing transaction revenues.

Exhibit 8.9 Text Extract – Intuit QuickBooks – Revised Online Strategy (Cont'd)

Villanueva surmised that the level of integration would determine the interplay between these two deal levers. "In light partnerships, since the long-term viability is questionable, we must reduce our risk by demanding a high slotting fee. Whereas, in deep partnerships we could settle for lower up-front fee in return for a greater revenue share," he said.
"The deals may not be so black-and-white. What if we decide to build a 'deep' relationship with a young start-up? Since the start-up by its nature is risky, should we expect both a large slotting fee and a significant revenue share?" wondered Nye.

Both Nye and Villanueva knew that, despite the detailed criteria, several uncertainties would remain. For instance, quality of the code, robustness of a partner's organisation, and fit between the two cultures would be issues that had to be determined on a case-by-case basis. Moreover, in order to attract the best partnerships deals in the future, they had to find a balance between generating demonstrable recurring revenues for QuickBooks with making their partners successful. Both recognised that they were likely to make mistakes in the beginning. However, they were determined to make as few mistakes as possible and learn quickly. Sobered by this realization, Nye and Villanueva reviewed the two main deals before them.

Source: Kochikar & Lal 2000, pp. 7-9.

Table 8.3 Intuit QuickBooks – Revised Online Strategy – Ranked Word List

Anchor Concepts	Frequency
service	14
Quickbooks	13
Nye	12
partnership	10
online	9
partner	9
build	7
deal	7
revenue	7
Villanueva	7
business	6
our	6
small	6
both	5
brand	5
integrate	5
make	5
product	5
two	5

Source: data analysis performed using TACT (1997)

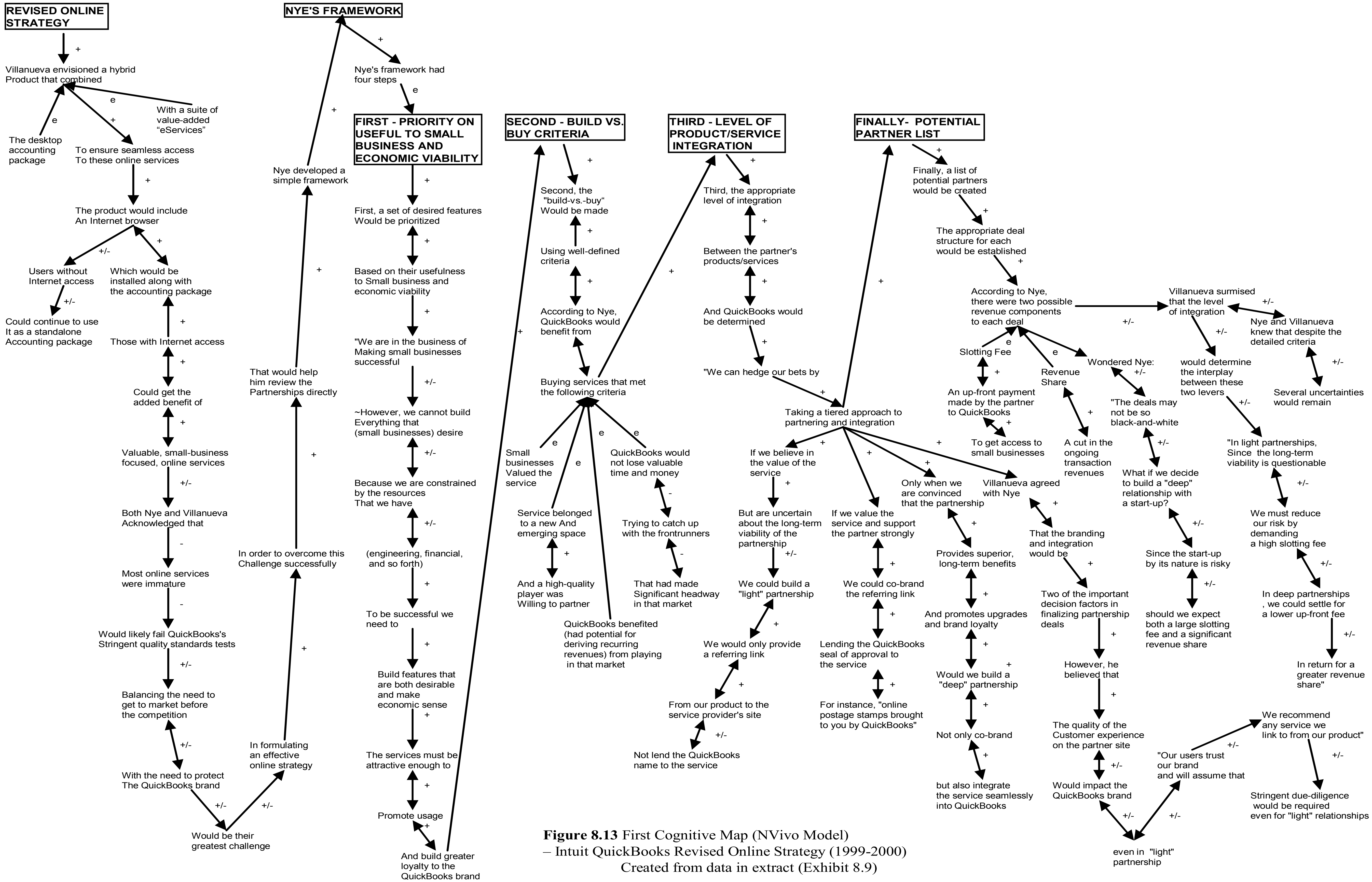


Figure 8.13 First Cognitive Map (NVivo Model)
– Intuit QuickBooks Revised Online Strategy (1999-2000)
Created from data in extract (Exhibit 8.9)

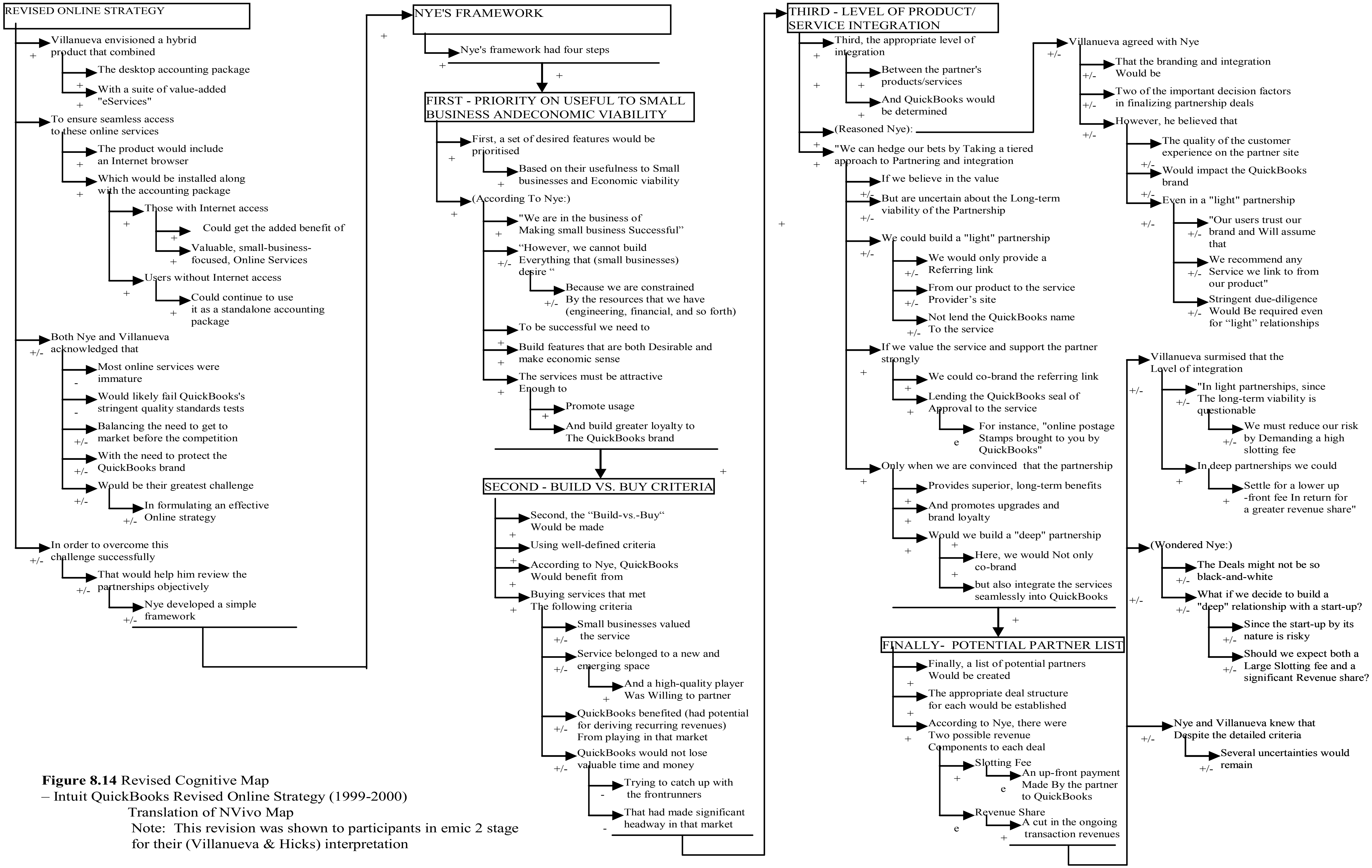


Figure 8.14 Revised Cognitive Map
– Intuit QuickBooks Revised Online Strategy (1999-2000)
Translation of NVivo Map
Note: This revision was shown to participants in emic 2 stage for their (Villanueva & Hicks) interpretation

According to Kochikar & Lal (2000), Villanueva put forward a concept combining the QuickBooks desktop accounting package with a suite on online value-added “eServices”.

Translated practically, the concept would mean that the next version of QuickBooks (probably QuickBooks 2001) would be sold in standard form on a CD-ROM. However, when installed with the Web browser option it would enable the user to access a wide range of tailored small business services. Of course if the user did not have Internet access, QuickBooks would still work as a stand-alone accounting package.

While this concept seemed to be quite plausible, Kochikar & Lal (2002) pointed out that Nye and Villanueva realised that most online services were mature from a development and quality assurance perspective – and particularly when viewed against Intuit QuickBooks’ very strict development processes.

Nye and Villanueva faced a challenging management issue – to overcome a very structured decision-making mindset within the SBD and Intuit based on maintaining high quality products to protect Intuit QuickBooks’ brand and image, in order to bring a new service to the market before their competition.

A minimum requirement to address the prevailing management mindset was for Nye to set up a framework to objectively assess and qualify potential partners for online applications and services. Nye set up a four-step framework for partner assessment and selection.

The first step of the framework was to prioritise a set of features based on both their usefulness to small business and economic viability. Nye added insight to this step with a strong business-to-business oriented comment (unusual for a company heavily driven up to that time by B2C activity) that “We are in the business of making small business successful”. However, Intuit QuickBooks could not build everything for that market because of constrained resources.

Although Nye emphasised that features should make economic sense, he really wanted features that would promote usage of the QuickBooks product and build greater brand loyalty to QuickBooks.

The second step of the framework was to decide on whether chosen features could be built in-house or had to be brought in. At this stage, though, the focus is still broadly on the feature or the services associated with the feature and not yet down to specific application or service providers.

Four criteria were put forward in step two, focused on small businesses perspective of the service and direct value for QuickBooks.

The first criterion, that small businesses should value a service, is obvious. The second criterion, that the service is in a new and emerging space represents a new opportunity for QuickBooks – but only with the additional qualifier that it could find a willing and high quality partner to develop and run the service.

The third criterion was based on QuickBooks gaining potential recurring revenues from offering the service. The final criterion for step two was to protect QuickBooks from losing valuable time and money, while trying to develop services to catch up with others who had launched the service in the marketplace. QuickBooks may benefit from working with partners who had already borne the pain of developing, trialling and running online services to small businesses.

The third step of the framework was to assess the appropriate level of integration with the QuickBooks product. Both Nye and Villanueva agreed that branding and integration of services would be important in the actual finalising of partnership deals. Villanueva went further to emphasise that the quality of customer experience on the partner's website would impact on the QuickBooks brand.

Understanding an extended multidimensional perspective for online customer activity is one of the key concepts for e-business marketers to have emerged from the advent of the Internet for business. Not only do customers surf up and down and across a website but they will go out through set-up (and, if the links are set up properly, back to the first website).

From the customer's (or surfer's) viewpoint both the initial provider and their extended providers in the same surfing stream can be seen as one organisation, providing an extended image of products, services and even branding.

Villanueva recognised that even with fairly minor low-level partnerships, some form of qualification and assessment would be required for them with regard to the quality of their websites. QuickBooks' endorsement of those relationships would have implications for QuickBooks' own image and branding.

Nye proposed a three-tiered approach to partnering and integration. "Light" partnerships were appropriate where QuickBooks believed there was value in the partnership (as qualified in the previous two steps of the overall framework), but there was significant uncertainty about the long-term viability of the relationship. In a "lite" relationship QuickBooks would only provide a referring link from the QuickBooks product to the service provider's site, but not lend the QuickBooks name to the service.

The second tier of partnership was appropriate where QuickBooks valued both the service and the partner strongly. QuickBooks would co-brand the referring link and endorsed the service. An example from around 2000 was "online postage brought to you by QuickBooks".

The third tier was a "deep" partnership where QuickBooks could see superior, long-term benefits that promoted QuickBooks product upgrades and brand loyalty. QuickBooks would co-brand the service, but also work with the partner to integrate the service seamlessly into the QuickBooks product and online environment.

The final step in the overall framework was to create a list of potential partners. A deal structure would be established for each potential partner based on revenues through either a slotting fee (an upfront fee from the partner to QuickBooks or a revenue share (a cut in the ongoing transaction revenues), or a combination.

However, both Nye and Villanueva, on reflecting about these arrangements, were not so sure that partnerships could be so clearly categorised for revenue dealing and partnership tiers.

Villanueva thought that in higher-risk "lite" partnerships, a higher slotting fee would reduce the risk for QuickBooks, but in "deep" partnerships QuickBooks could settle for lower slotting fees, while opting for a higher revenue share.

However, Nye was not convinced that partnership deals would be so "black-and-white". He raised the scenario of a start-up services provider where QuickBooks wanted a "deep" relationship, where perhaps a high slotting fee and a high revenue share might be appropriate.

Given the immaturity of online services providers and the huge scope for new Internet-based applications at the time, this scenario was much more plausible then both Nye and Villanueva realised.

Nevertheless, both Nye and Villanueva recognised that despite having devised a detailed framework for partner assessment and selection, there were still significant uncertainties with developing and executing the revised online strategy which now relied so heavily on partnerships.

Overall, the map demonstrates a stream of justified logic channelled into a structured framework for decisions associated with selecting partners for the revised online strategy. It also highlights that, despite the development of a structured decision-making approach, concerns with high uncertainty over developing new products and services for and in a new online environment could undermine or cause major changes to that approach.

8.9.3.1 Updated Cognitive Map 3 – Revised Online Strategy (1999-2000)

Interviews with Villanueva (2002) and Hicks (2002) validated the original cognitive without modifications except for an additional insight that the second step of the framework that is, deciding on building versus buying services was often based on opportunistic spending. The updated cognitive map with numbered sections is presented in Figure 8.15.

There were, however, some questions emerging from the interview as to the real formality and structure of the four-step framework. The actual qualification and selection may have been in practice less formalised and more opportunistic, although that did not necessarily detract from the selection of appropriate partners.

Decision-making may have been more adaptive and improvised than the structure suggested in the map. Nevertheless, the map does contain logic and description for intended decision-making on partner selection.

8.9.4 New Cognitive Map 4 – "QB Economy" Concept (2001-02)

During the interview stage for the Intuit QuickBooks project it became apparent that not only a new DSA model for QuickBooks application development in 2002 would be desirable but suggestions were put to create a new cognitive map addressing the "QB Economy" concept.

Development details for the final cognitive map include:

- Extract section, size and exhibit:
    - "Leveraging The "QuickBooks" Brand into a community of partners" section from the "Intuit QuickBooks "QB Economy" Concept" vignette, ½ page, Exhibit 8.10

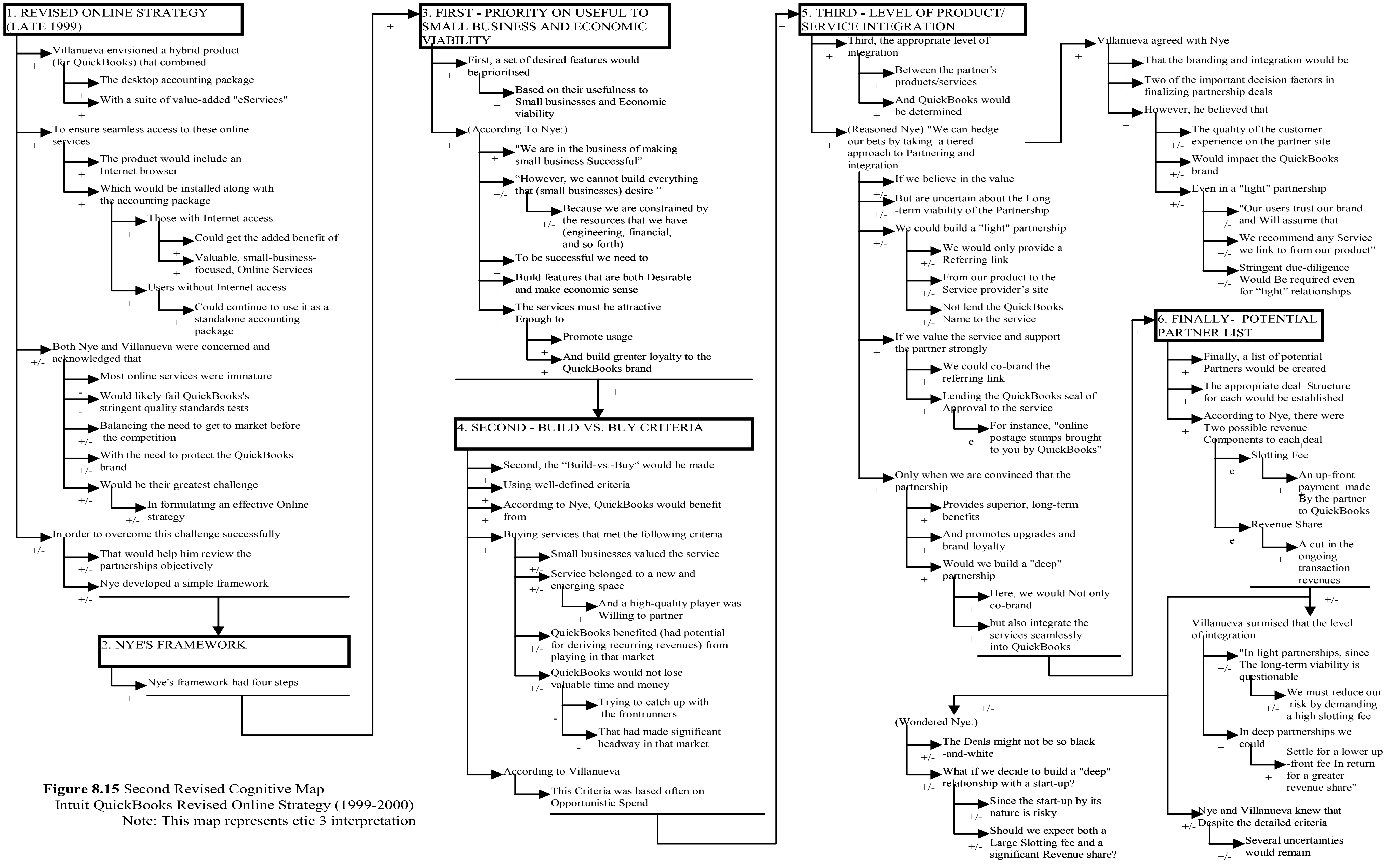


Figure 8.15 Second Revised Cognitive Map – Intuit QuickBooks Revised Online Strategy (1999-2000)
Note: This map represents etic 3 interpretation

- TACT word frequency lists and comments on key words:
    - First & final pass: 17 words with frequency > 1
    - Final ranked word list: Table 8.4
    - The ranked word list highlight themes related to *Intuit QuickBooks* working with *developer*s and *partner*s on *product*s, *application*s and *service*s (italicised words mainly in singular form were ranked highly in the ranked word list).
- NVivo ranked word, sentence and phrase node count = 55
- Association of concepts coding used: +, -. +/-, e
- NVivo PowerPoint cognitive map: Figure 8.16
- Reformatted PowerPoint cognitive map: Figure 8.17

The "QB Economy" concept was launched during 2001, with the aim of leveraging the QuickBooks brand into a community of partners (box 1). The concept could be seen as a culmination of Scott Cook's vision to aggregate online customers as articulated in the cognitive maps for Intuit QuickBooks' Initial Online Strategy.

Exhibit 8.10 Extract from Vignette – Intuit QuickBooks "QB Economy" Concept

LEVERAGING THE "QUICKBOOKS" BRAND INTO A COMMUNITY OF PARTNERS

During 2001, Intuit's Product Development Group established new and extensive relationship programs for its third-parties. Intuit recognised that strong customer, developer and service provider awareness of the "QuickBooks" brand-name could be leveraged to establish a community. The Intuit Developer Network was launched under the "QB Economy" label.

Extension from a product-focused strategy toward a "business environment" approach epitomised by the "QB Economy" label, prompted Intuit to rapidly expand the range of applications and services to fit with the QuickBooks application.

Vertical applications could extend QuickBooks' appeal to specific SME groups, for example, retail applications (including POS systems), professional taxpayers and tax agents. vertical services could provide information to support or enhance QuickBooks customised applications.

Horizontal applications could extend QuickBooks capabilities for most SMEs and some large corporation, for example, e-commerce solutions, merchant accounting (credit card services), interfaces between SME and financial services providers, and management of local, state and federal sales and goods and services tax requirements.

The Intuit Developer Network was a vehicle for the developers of QuickBooks in particular to quickly incorporate new features and services, while Intuit could focus internally on upgrading the core QuickBooks product and on partnership management.

Table 8.4 Intuit QuickBooks – "QB ECONOMY" Concept – Ranked Word List

Anchor Concepts	**Frequency**
Quickbooks	8
service	7
intuit	7
application	5
developer	5
could	5
product	3
economy	3
sme	3
tax	3
new	2
extend	2
community	2
label	2
management	2
network	2
vertical	2

Source: data analysis performed using TACT (1997)

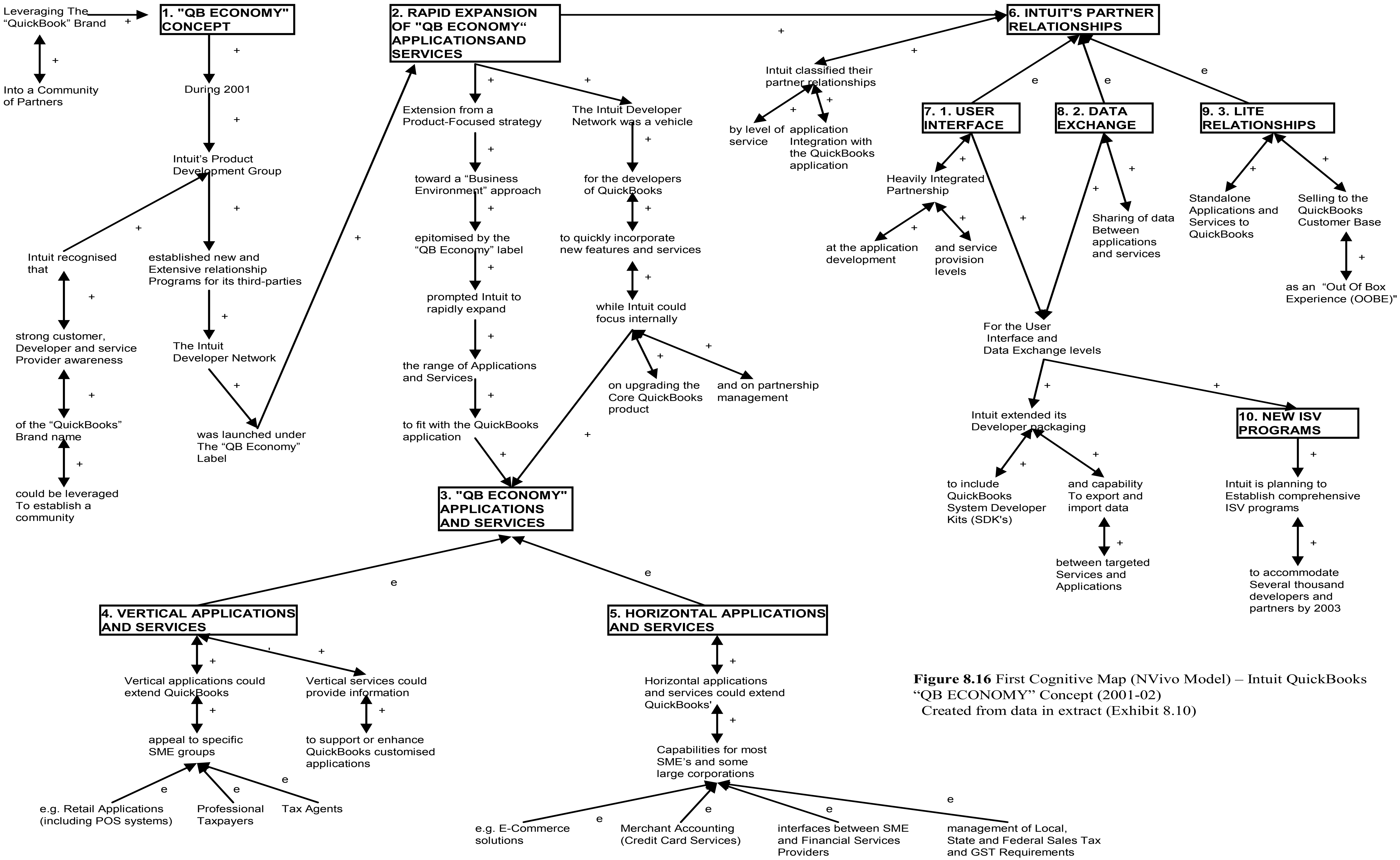


Figure 8.16 First Cognitive Map (NVivo Model) – Intuit QuickBooks "QB ECONOMY" Concept (2001-02)
Created from data in extract (Exhibit 8.10)

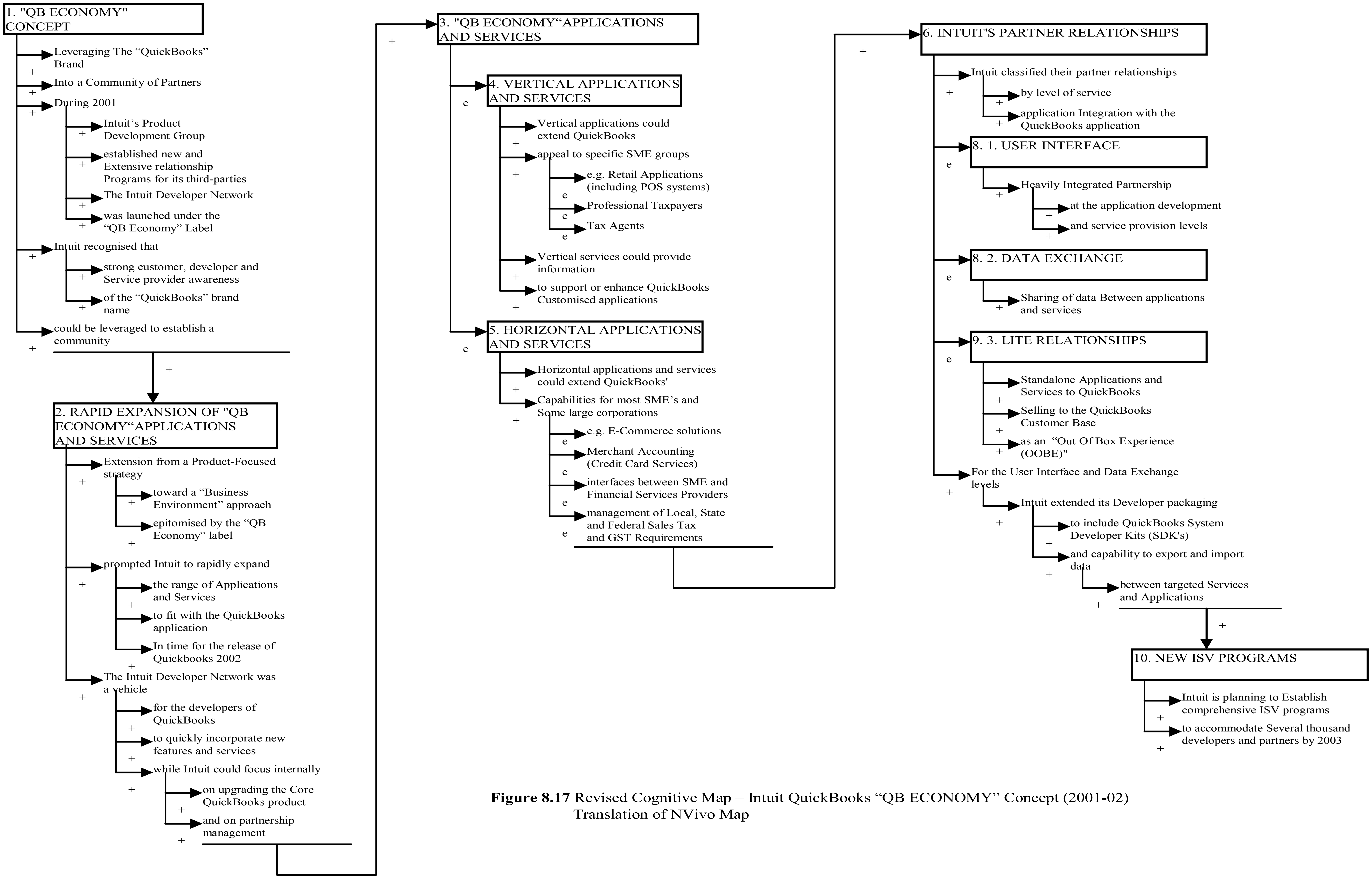


Figure 8.17 Revised Cognitive Map – Intuit QuickBooks "QB ECONOMY" Concept (2001-02)
Translation of NVivo Map

Intuit Corporation's Product Development created the "Intuit Developer Network" program which included several new and extensive programs for third-parties working with Intuit, and which was also positioned under the "QB Economy" label. The Intuit Develop Network approach incorporates experience gained by the QuickBooks SBD through its revised online strategy into a more broad corporate partnership program.

In particular, the recognition of strong customer, developer and service awareness of the QuickBooks brand name could be extended out of the QuickBooks SBD into the whole Intuit Developer Network.

During 2001-02 there was a rapid expansion of "QB Economy" applications and services (box 2). The "QB Economy" concept in turn, seems to have accelerated decision-making process toward factoring in more issues related to the partner community than just a strict internal product-focused set of processes.

Intuit's Product Development Group and the QuickBooks SBD added a range of applications and services to fit with the QuickBooks applications in time for the release the QuickBooks 2002 product (October - November 2001).

As distinctions between QuickBooks applications and ISV and service providers own applications and services became increasingly blurred, the Intuit Developer Network would be the main vehicle to enable these third parties to continue development and enhancement with Intuit. Intuit's own development and management teams could then focus on upgrading the core QuickBooks product and concentrated more on partnership management.

Applications and services for the "QB Economy" (box 3) can be classified into vertical and horizontal groups. Vertical applications and services (box 4) can extend QuickBooks appeal to special SME groups such as retailers, professional taxpayers and tax agents. These applications and services could support or enhance QuickBooks customise applications.

Building up vertical applications and services may over time lead to creation or refocusing of marketing, sales and product management toward vertical or industry market groupings. However, Intuit and the QuickBooks SBD had not moved to such a structure in 2001-02.

Horizontal applications and services (box 4) are used to extend QuickBooks' product capabilities across most SMEs and some large corporations. Operational areas in these organisations addressed by QuickBooks include e-commerce solutions, merchant accounting (management and operation of credit card services), links between SMEs and financial service providers, and management of Sales Tax requirements at various levels of government.

Intuit's approach to managing partner relationships (box 6) reflects an evolution from the partnership tiers created by Nye and Villanueva in 1999-2000 for the QuickBooks SBD. There are still three partnership tiers, but more precise classification by level of service and application integration with the QuickBooks (and Intuit's other) application.

The "User Interface" level (box 7) is characterised by a heavy integration of applications and services directly with the QuickBooks application.

High levels of management and application development information and product sharing between QuickBooks SBD and partners at this level is required at all stages of the QuickBooks application development cycle. Such interaction is also required for the ongoing development and maintenance of an online "QuickBooks Economy" environment.

At the "Data Exchange" level (box 8) there is data exchange between applications and services, and while heavy directing and application integration is not required, compatibility and consistency is still necessary. There is communication between QuickBooks SBD management but mainly at the project teams level to ensure that information be transferred between relevant applications and services.

"Lite Relationships" (box 9) do not generally integrate with the QuickBooks product but may be sold to the QuickBooks customer base (online or offline) with QuickBooks' endorsement. These applications and services would be used at the customer's risk and supported by the third-party – hence QuickBooks terms them as an "Out of box Experience" (OOBE).

Negotiations with these parties may be undertaken independently of the QuickBooks application development cycle, although there would very likely be attempts to still position Lite Relationships with upcoming QuickBooks' versions.

Intuit provides extra resources for user interface and data exchange developers. System Developer Kits (SDKs) are provided for these partners, echoing several other software houses in this project. Since around 1999-2000, Internet-based software developers have focused on making life a lot easier for developers to quickly install their application and start further development using SDKs.

SDKs now include source code, new patches and fixes, a collection of the suppliers plus other partners programs and tools, plus program, and some training documentation. SDKs in 2003 can be seen – just like the evolution with QuickBooks - as a combination of offline applications with online updates and support services.

Intuit also provides the capability to test the export and import of data for targeted applications and services.

Intuit is now evolving its Intuit Developer Network with more comprehensive ISV programs (box 10). These programs are similar to the marketing and application porting programs provided by Sun Microsystems for UNIX in the early 1990s, DEC for its alpha chip technology, also in the early 1990s and Microsoft for Windows through the 1990s and into the current decade.

Intuit is ambitious about developing a comprehensive range of partnerships – to accommodate up to several developers and partnerships by late 2003.

The "QB Economy" concept cognitive map appears to herald the emergence of a third online strategy, which clearly evolves from the revised strategy, but now emphasising high levels of partnership integration and involvement beyond just the QuickBooks SBD through to the whole Intuit corporation.

Decision-making has evolved from just the selection of partners through to shared decision-making at all levels of QuickBooks' application development cycles and is a parallel reflection of the blurring of QuickBooks' and partners' applications and services in an online environment.

Decision-making in relation to partnerships has become quite structured, and the number and combinations of partners are increasing and becoming more complex. Indeed, the Intuit Corporation has undergone a seismic shift from where it stood in 1998 as being an internally focused developer, to in 2001-02 becoming a virtual corporation with a significant offering of online applications and services.

8.10 Extension Questions

A set of additional questions was put to the interviewees for the case study, covering dominant logic, shared vision, leverage points, and strategic marketing issues. These questions were asked during the initial interviews and followed-up with further validation and commentary in the follow-up interviews.

8.10.1 Dominant Logic

The interviewees agreed that for the scope of the original case study, Scott Cook represented dominant logic in terms of driving his vision for an ad-supporting online strategy.

Tim Villanueva and Dan Nye drove the translation of the vision into strategy and action through the establishment of Quicken.com and QuickBooks.com and the creation of the early online applications and services.

From around 2001, the overall Intuit Product Development Group appears to have taken over Villanueva and Nye's roles in driving and translating dominant logic into action.

It is probably too early to judge how much logic is permeating through into Intuit from partners to the point where it could be regarded as "dominant". Perhaps such an assessment could be more clearly made around 2004-05.

8.10.2 Shared Vision

Two distinct "eras" of shared vision emerge from the Intuit QuickBooks case study. From 1992 to 1998, shared vision was mainly internal to Intuit among the product management, sales and marketing, and development groups. Strict development and quality assurance programs reinforced a strong and consistent shared vision for all of Intuit's products.

The second "era" of shared vision emerges from Intuit's iterations with implementation of its online strategies. Intuit has shifted from an internal product-driven vision to an extended services vision with significant support from strategic partners.

Vision is shared and refined through experts on particular services within Intuit and their strategic partners. These "experts" may go further and provide dominant logic for specific services but not for the overall Intuit umbrella strategy.

The Intuit Developer Network is likely to develop its own shared vision over the next few years, but, like dominant logic, such vision may not be clearly discernable until about 2004-05. Nevertheless, Intuit is working hard through the creation of a wide range of partnership programs to build stronger shared vision through and with its partners.

8.10.3 Leverage Points

Five leverage points were identified during the interviews. These are outlined in chronological order.

8.10.3.1 Creation of QuickBooks Application for Small-business Owners (1992)
8.10.3.2 Development of Portal Strategy (Spring 1998)
8.10.3.3 Reorganisation into Business Groups (1999)
8.10.3.4 Revised Online Strategy Based on Partnerships (Late 1999-Summer 2000)
8.10.3.5 Creation of "QB Economy" Concept (2001)

8.10.3.1 Creation of QuickBooks Application for Small-business Owners (1992)

Intuit identified an opportunity to create a simple financial accounting and management for small businesses. Intuit discovered that a significant number of customers for the Quicken application were actually purchasing it for use in their small business operations. The customers wanted a simple but effective application that did not require advanced accounting or computing skills to install and use.

8.10.3.2 Development of Portal Strategy (Spring 1998)

Intuit recognised that the Internet was a fast growing channel for Quicken and QuickBooks sales in the late 1990s. However, Scott Cook wanted to do more than just use the Internet as an ordering channel for products. Cook envisioned a new industry, similar to the media industry with its business-model based on advertising revenue.

To test the vision, Intuit devised a simple portal strategy, a small business channel with a small range of online services, within the Quicken.com website. The portal strategy was a small experiment, but produced a substantial amount of learning about not only the overall vision but also the context for actualising the vision – and the value of partners to make on online strategy work.

The experiment also demonstrated that customer focus was more important than technical and application excellence and innovation. This point should be qualified in that at the time in 1998, most online applications and services were not mature and required a lot more development.

8.10.3.3 Reorganisation into Business Groups (1999)

While the portal strategy was being implemented, Intuit reorganised into three main business groups, the Personal Finance Group (PFG), the Small Business Division (SBD), and the Tax Division.

Nye and Villanueva continued the roll out of the online strategy in the Small Business Division with the QuickBooks product. QuickBooks.com was significantly upgraded into a distinct online environment for QuickBooks customers, and the channels strategy was refined further through the QuickBooks SBD.

Over time the experience accumulated by the QuickBooks SBD would be extended back into the Intuit Corporation through the creation and expansion of the Intuit Developer Network program.

8.10.3.4 Revised Online Strategy Based on Partnerships (Late 1999-Summer 2000)

Nye and Villanueva's revised online strategy represented a transformation point for Intuit's QuickBooks SBD, but also for the Intuit Corporation. The strategy challenged Intuit to change from being internally focused on product development, toward being focused on services and partners. The strategy challenged Intuit's product development and quality philosophies and practices.

Nevertheless, the new approach was still operationalised into the existing application development cycles, such that the QuickBooks 2000 product was launched within a standard release timetable with an initial set of services, provided through a small group of partners.

Having set up a framework for online services and working with partners, the development of an online branding approach and larger partnership network were logical progressions for Intuit.

8.10.3.5 Creation of "QB Economy" Concept (2001)

Creation of the "QB Economy" Concept and the Intuit Developer Network represented a culmination of learning about online products and services development and delivery that was expanded to the whole Intuit Corporation. It is consistent with Scott Cook's original online vision and, as of late 2003, has been in place for over two years. Expansion of the branding and the developer network is a high priority for Intuit through 2002-03.

8.10.4 Strategic Marketing Issues

Three strategic marketing issues were identified from the interviews. These issues include:

8.10.4.1 Developing Value Beyond the Basic Product or Software Application – In-house or Through Partners

8.10.4.2 Cross-selling Products Through All of Intuit's Channels

8.10.4.3 Using the Internet to Communicate Marketing and Alliance Programs

8.10.4.1 Developing Value Beyond the Basic Product or Software Application – In-house or Through Partners

Intuit's online strategies were built on creating value beyond the basic QuickBooks product by providing links to value-added services provided by partners. As products and services have come more complex in an online environment, Intuit has "stuck to its own knitting", focusing on upgrading its own core applications and relying on partners to provide most value-added services.

Villanueva (2002) highlighted the need for value-added applications and services to be credible within the QuickBooks brand, and to have clear value beyond the QuickBooks core product.

Although the strategic issue of developing applications or services in-house versus through partnerships was an obvious step in Nye's channels assessment framework in 1999, it remains an important issue that has been constantly revisited as Intuit's partnership programs rapidly grow in membership, number and diversity.

Intuit prefers to partner rather than building in-house or ISV/Service acquisitions. Development of an appropriate ISV/Services Provider strategy has evolved such that Intuit's experience in strict application development cycles quality control is now shared with these partners.

Intuit appears to be increasingly committed to a shared and open development approach for the foreseeable future.

8.10.4.2 Cross-selling Products Through All of Intuit's Channels

Intuit is keen to use the "QB Economy" online environment to cross-sell products and services into other channels, such as through retail and direct avenues. Such cross-selling could be similar to Amazon.com acting as an online ordering vehicle for companies such as Toys "R" Us – that is, mixing online and offline strategies.

Would Intuit go as far as actually branding third-party applications in QuickBooks packages for retail or direct marketing programs? Would Intuit provide a share of services revenue back to a retailer recognised to be a source for a QuickBooks sale? Would Intuit allow a third-party service provider with QuickBooks' branding to go on a QuickBooks' retailer's website?

There are several possibilities and permutations for this issue which Intuit's management will ponder over as they pursue cross-selling opportunities.

8.10.4.3 Using the Internet to Communicate Marketing and Alliance Programs

The case study addressed the development and execution of an online strategy for Intuit, but there are still many issues related to using the Internet to communicate with partners.

The Internet can be used to communicate and execute marketing programs through and with partners. It can also be used to provide support, updates and new versions of applications to partners. Intuit has set up the Intuit Developer Network to partly address this issue and will continue to upgrade its partner programs using the Internet as a platform for increasingly sophisticated exchanges of information, applications and services with partners.

8.11 summary

The collection of DSA models, events chronology maps and cognitive maps plus extension questions encapsulates the decision-making issues associated with both a transition to an online presence for Intuit QuickBooks' products, services and branding, and to a different application development cycle incorporating partner inputs at most steps of the process.

The package also highlights a transformation for Intuit from an inward-looking product-focused organisation in the late 1990s to a focus on services and partnerships to provide services with and beyond the Quickbooks product. The interview process enabled extension of the case study timeframe to show a third iteration of Intuit QuickBooks' online strategy toward an umbrella "QB Economy" concept.

Key insights from the case study include:

- 8.11.1 Transformation from Internal Product-focused Decision-making to Outward Services Focus.
- 8.11.2 Translation of Vision to Action Using Experiments

8.11.3 Partnership Management and Involvement in Application Development Cycles

8.11.4 Managing Partners as an Online Community

8.11.1 Transformation from Internal Product-focused Decision-making to Outward Services Focus.

The package of models and maps provides effective insights into a transformation in Intuit's decision-making processes from 1998-2002. In 1998, Intuit had quite structured processes for product development and strict development cycles.

Intuit's extended product development to cover online value-added services radically changed decision-making toward working with partners and understanding products and services that could enhance the QuickBooks' brand online.

8.11.2 Translation of Vision to Action Using Experiments

Scott Cook, one of the founders of Intuit, had a strong vision for the Internet becoming a new industry similar to media with an ad-supported revenue base. He wanted Intuit to be the supplier of information and services to small business within the "Internet Industry".

Tim Villanueva and Dan Nye took the broad vision and tested it through a small experiment for the QuickBooks application – though its sister products' online environment (Quicken.com). The Small Business Channel on Quicken.com was a simple portal experiment that produced significant learning about presenting products and services online.

Lessons from the portal strategy prompted Nye and Villanueva to develop a partner qualification framework and to move online initiatives to QuickBooks' own website. The revised online strategy for Quickbooks.com was developed further with the "QB Economy" concept. Intuit Corporation expanded the QuickBooks partnership initiatives into the Intuit Developer Network corporate partnership program.

8.11.3 Partnership Management and Involvement in Application Development Cycles

The package of models and maps for the Intuit QuickBooks case study highlighted the decision-making associated with the selection of partners to provide applications and services for an online environment.

A significant benefit emerging from the interviews for the case was sufficient information and insights to develop a revised DSA model for development cycles depicting the direct involvement of third-parties throughout those cycles. The new DSA may be an indicator for the future development of online applications and services.

8.11.4 Managing Partners as an Online Community

The original case study focused on the creation and revision of an online strategy for Intuit QuickBooks. New models and maps produced from the interviews for the case highlight Intuit's efforts to manage its partners, but only lightly covering the online management of them.

In an online environment, distinctions between development and delivery of applications and services can become blurred. Intuit has worked hard on delineating the development of its core application QuickBooks from the online applications and services which are linked to the core product through Web-browsers.

Intuit has attempted to manage its partners through the Intuit Developer Network as an online community with a mix of online and offline programs.

However, as the total offering is consolidated through the integration of applications and services, Intuit faces great challenges in managing its large and diverse developer community

8.12 Conclusions

The Intuit QuickBooks case study highlights the decision-making associated with transitioning a whole business toward operating in an online environment over time. Three iterations of online strategy revisions from 1998 through to 2002 were explored using DSA augmented by events chronology and cognitive maps.

Conclusions are presented in the sections below:

8.12.1 Theoretical Issues (or Questions) for the Intuit QuickBooks Case Study

The case study examines decision-making associated with an established software house attempting to position its applications and its company in a new online environment.

Intuit QuickBooks is a software house with very detailed and strict internal application development standards and processes. Moving toward application development incorporating significant amounts of third-party applications was a significant decision-making issue.

Going further to establishing third-party online services in a QuickBooks virtual community was just as much a cultural change for decision-making at QuickBooks as for application development using third-parties.

The case study highlights these transitions as iterative and, to a fair degree, experimental. QuickBooks management learned more about both working with third parties, and online environments through the strategy iterations.

The two extended storytelling methods used for analysis in the advanced hermeneutic analysis and extended DSA were effective at identifying the key decision-making issues, events, and linkages associated with the dynamic evolution of Intuit QuickBooks' applications, in terms of the transitions to a more open and partner oriented application development approach and experimentation with a number of online strategies for the application.

Both methods provided mapping and validating of the initial accounts of decision-making, followed up with subsequent revisions of those accounts using a systematic emic/etic representation system – including the development of new DSA models and maps for new application development approaches in 2002-03.

8.12.2 Intuit QuickBooks Case Study Limitations

A complete set of multiple interviews with multiple respondents who were directly involved in the original case study on the company was completed for the Intuit QuickBooks case study.

However, the new DSA models and new cognitive maps created from the interview material were not validated.

The case study may have benefited from gaining insights from selected third-party developers and service providers on Intuit's views and the actual implementation of their online strategies. A clearly defined processed for identifying and qualifying appropriate third-parties would be required.

8.12.3 Contributions from the Intuit QuickBooks Case Study to Strategic Sensemaking

The Intuit QuickBooks case study applies an extended DSA approach using DSA models, events chronology maps plus cognitive mapping to decision-making for application development and product strategy over time.

Three strategic iterations are mapped and analysed from 1998 to 2002. The analysis provides insight into both Internet application development and the establishment of a computer-mediated environment for business-to-business services and support.

The Intuit QuickBooks case study provides mapping of the decision-making associated with the increasing inclusion of third-party applications and services in application development cycles.

The case study highlights a fundamental transformation to strategic thinking to reconceptualise "application" (or product) development for computer-mediated environments. "Applications" have to be extended from internal application development into a set of services with the potential for new and different value-added online products and services.

8.12.4 Contributions from the Intuit QuickBooks Case Study to Practice

The Intuit QuickBooks case study provides valuable insights into transitioning a "traditional" software house with established strict internal application development processes and standards into a more "open" application development approach.

The case study highlights the decision-making associated with negotiating and incorporating third-party applications in application development processes.

Perhaps even more substantial insights are offered on transitioning a popular application online. Additional decision-making is required to develop a virtual community with appropriate third-party services within the new community.

The case study challenges traditional software house executives' views of what actually constitutes an "application". For software houses used to providing a boxed package with a simple licence to run an application, stretching executive mindsets to developing a full online virtual community providing developer and business services is a large and difficult challenge.

8.12.5 Intuit QuickBooks Case Study Endnote

In May 2003 Intuit Corporation introduced its "Right for My Business" strategy offering industry-specific versions (or "flavours") of QuickBooks. Industry "flavours" of QuickBooks included:

- Non-profit Organisations
- Healthcare Practices
- Enterprise Solutions (Accountants)
- Contractors

QuickBooks also offered a distinct QuickBooks Online Edition in its 2003 range (see Intuit 2002 and Intuit 2003).

The QB Economy umbrella appears to have been diffused into a more vertical marketing strategy and to the Intuit Developer Network (IDN). IDN has grown rapidly through 2002-03 with distinct marketing & sales, and technical services, marketing & sales services from IDN include marketing programs and support from Intuit. Technical services include SDKs (at V3.0 in late 2003) and supporting developer tools.

IDN has upgraded QuickBase (an information management application created in 1999) to be a standards-based Web application development environment. Intuit also provides access for QuickBase developers to market their applications and services to each other. (For Intuit Developer Network services, see Intuit 2003).

Intuit continues to increase its programs and services for both developers and service providers. QuickBooks has come a very long way from being offered as a single copy of accounting software over the counter at the local retail store for small business users.

Chapter 9: Case Study – Trilogy

Abstract

The Trilogy case study uses two storytelling methods for analysis – an **advanced hermeneutic framework**, and an **extended form of decision systems analysis (DSA)** incorporating **cognitive mapping** – to explore the strategic thought associated with building a software house around a philosophy of "best practice" application development.

The account explores decision-making for turning an ambitious vision (automation of enterprise-based sales and marketing activities) into a focused application (pricing configuration software), and then into a large software house offering enterprise e-commerce suites. Trilogy's management team's decision-making was heavily influenced by a strong perception that the organisation needed to take risks to achieve "critical mass" in anticipation of a convergence of "back-office" and "front-office" applications into one market.

The case study also addresses Trilogy's transition of its applications into Internet environments, plus the transformation of the organisation from a product-orientation to a strictly industry-based business and application development perspective. Trilogy has not compromised its strict application development philosophies, but has incorporated its Fast Cycle Time (FCT) methodologies into its industry-focused business divisions – and now actually offers the methodology as a set of services.

9.1 Background

Trilogy Inc., of Austin Texas provides enterprise software focused on reducing the costs associated with "front-end" or "selling chain activities", including sales and marketing.

Trilogy emerged in 1991 from a group of Stanford University students pursuing the development of software to configure and price computer hardware parts.

Trilogy was included in the final case study list for the project as it met the initial criteria:

- Software has to be "beta"-tested and commercially introduced
    - The case study covers the evolution of Trilogy from its first SalesBuilder configurator application in 1991 through several versions to a full front-office application suite in 1998

- Case-studies were written by well-known authors
    - Professor Robert D. Austin has established expertise in research and executive curriculum development in information technology management, and more generally on the management of knowledge-intensive activities. He has written on these subjects in four books
- The application development process studied must be over a timeframe of more than 12 months
    - the original case study covers a timeframe of Trilogy's company history over just over seven years from 1991 to 1998
- Names of case study companies and participants can be disguised if requested
    - Permission was gained from David Franke and Chris Hyams, both for a combination of face-to-face and digitally recorded telephone interviews and the use of that information in the research project, without disguise. Certain documents related to Trilogy's application development methodology were used with written agreed permission from Trilogy
- Applications must be Internet-based
    - The initial case study addresses application development at Trilogy. The follow-up case study refers specifically to spinning off Internet-based applications into distinct Internet businesses (Dot.com). Revised accounts from respondents, however, clearly describe the transition to Internet-based application development during the timeframe of the case, plus the transition to a comprehensive e-commerce application suite.

Additional Case study selection criteria included:

- Size: Small/Medium
- Success: Yes
- Disruptive Technology: No, however, Trilogy's application enabled substantial cuts in sales and marketing costs and revolutionised quoting systems
- Harvard Business School Cases: "Trilogy (A)", Austin (1998); "Trilogy (B)", Mandel & Austin (2000).

The case study for the project traces Trilogy from its inception from an idea to address a specific marketing problem through to competing with large ERP organisations in the late 1990s and is extended to cover a new phase of industry-focused application development and delivery in 2001-02.

The Initial DSA model, events chronology map and the three cognitive maps for this case were developed from extracts drawn from Austin's (1998) Harvard Business School Case study "Trilogy (A)". Austin, with Mandel, wrote an additional short note for the Case study, "Trilogy (B)" addressing the spin-off of several Internet-based on "Dot.com" businesses from Trilogy in 1999-2000 (Mandel & Austin 2000).

The main purpose of Austin's (1998) and Mandel & Austin's (2000) case study was to explore risk-taking in terms of creating and rapidly growing a new business and instilling such a culture in the business, and particularly in software development.

Austin (1998) outlined the creation and rapid development of Trilogy from 1991 through to 1998, while Mandel & Austin (2000) note a list of spin-off businesses out of Trilogy in 1999-2000.

The case is relevant to this project because Austin provides a substantial account of application development and its central role in Trilogy's corporate culture. Trilogy's early applications were developed mainly in C++ but were transitioned across to an Internet-based platform using Java and XML from 1997-99.

Austin's (1998) account of the application is sufficient for the development of a representative DSA model and to undertake further analysis on application development and delivery. The outline and chapter structure for this case study is presented in Figure 9.1.

The DSA model and maps for this case were updated, plus extension questions on dominant logic, shared vision, key leverage points and strategic marketing issues were addressed through interviews.

The interview participants were one of the key players in the HBS case study, who is still with Trilogy, and the current Vice President of Development at Trilogy (see Franke 2002; Hyams 2002).

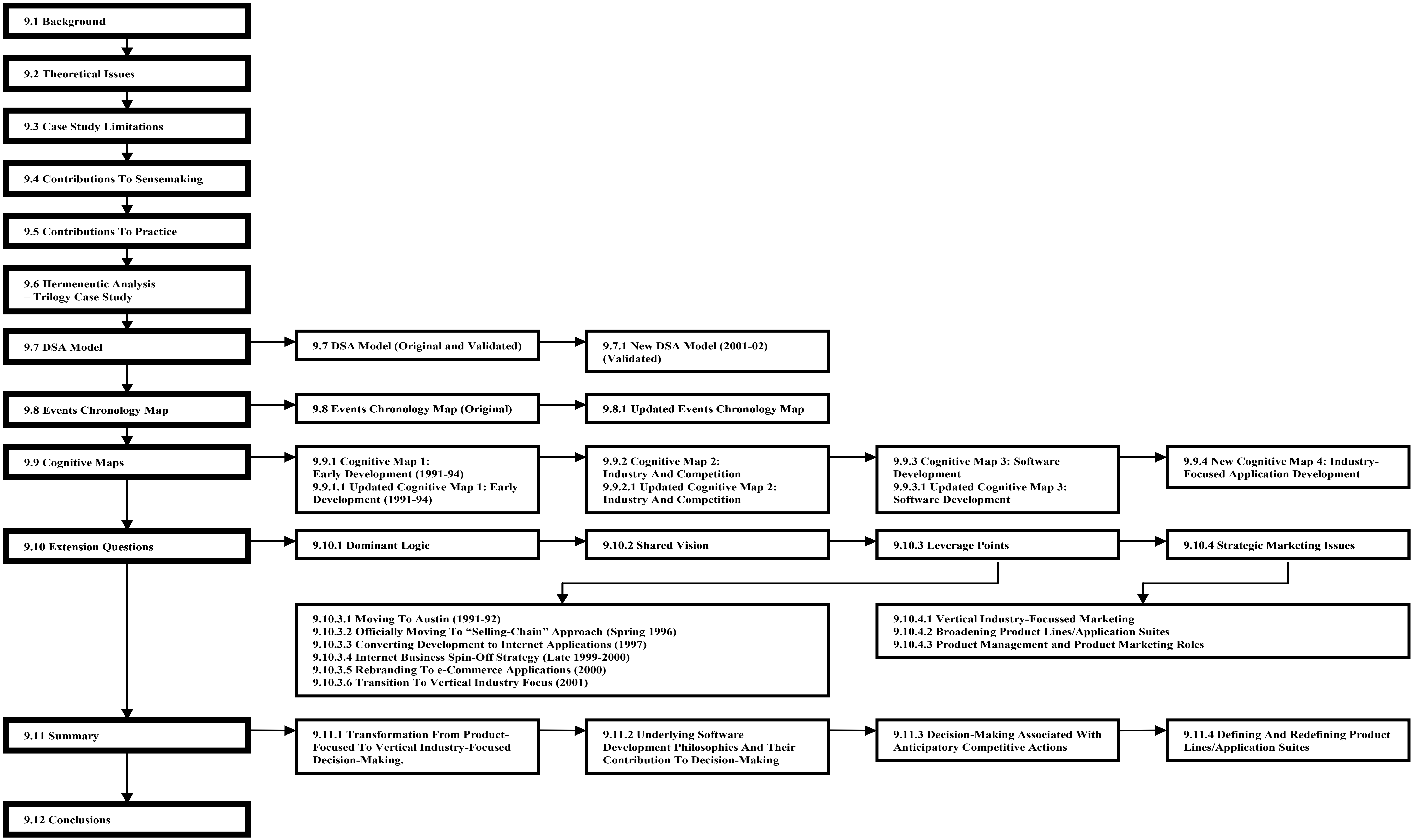


Figure 9.1 Trilogy Case Study – Chapter Structure.

9.2 Theoretical Issues (or Questions) for the Trilogy Case study

The Trilogy case study explores the decision-making associated with creating an organisational culture of "best-practice" application development processes.

The case offers insights into combining a philosophy of "best-practice" with risk-taking in order to grow a company rapidly to prepare for an eventual and imposing competitive showdown with larger and more established corporations.

The case study also provides insights into decision-making from the original conception of the product through to the creation and rapid growth of Trilogy and on to a transition into a provider of vertically-focused (industry-focused) applications.

There are substantial insights into the decision-making associated with the high standards set for product-oriented application development, and how those high standards are translated into an industry-focused application development environment.

Trilogy's strict and high-quality application development standards make it the most internally-focused software house in this project. Decision-making processes associated with such an environment provide an interesting comparison with other software houses in this project.

The storytelling methodology used for analysis in the case study – an advanced hermeneutic analysis, including an extended form of **decision systems analysis** (DSA) incorporating **cognitive mapping** – were effective at identifying key decision-making issues, events, and linkages.

The Methodology provided mapping and validating of initial accounts of decision-making, followed up with subsequent revisions of those accounts using a systematic emic/etic representation system.

Interviews for the Trilogy case study captured substantial insights beyond Austin's (1998) and Mandel & Austin's (2000) case study, particularly on Trilogy's new industry-focused application development cycles. Additional material provided by Trilogy enabled the most comprehensive insights into the actual application development cycle out of all the cases for this project.

9.3 Trilogy Case Study Limitations

The Trilogy case study benefited from a complete set of multiple interviews with multiple respondents who were directly involved in the original case study on the company, to validate the original set of DSA models, events chronology maps and cognitive maps. The new DSA models and new cognitive maps developed from interview material were validated within the timeframe allowed for this project.

There may have been additional benefit in finding more description about the transition to straight Internet-based applications during 1997-99. However, this issue was partly addressed through the collection of additional information on Internet development and Internet business initiatives, during the interview process.

9.4 Contributions from the Trilogy Case Study to Strategic Sensemaking

The Trilogy case study explored decision-making issues, events, and linkages using the storytelling method developed for the project. An advanced hermeneutic analysis provided several levels of emic/etic representation of decision-making, highlighting key case study issues.

The Hermeneutic framework incorporated an extended form of Decision Systems Analysis (DSA) highlights specific subsets of decision-making, through mapping of decision-making from original accounts with revisions, through subsequent contact and discussions with original decision-makers. The extended DSA also explored changes over time in shared vision and new application development.

Creation and validation of a set of DSA models plus cognitive mapping outputs represents an effective mapping of strategic sensemaking. In the Trilogy case study this approach is applied to turning an idea into an application, in conjunction with turning a small start-up business into a substantial and competitive enterprise software provider.

Trilogy had a small period of time in which to grow fast enough to be ready to face severe competition from much larger and more established ERP software houses that could stake out its niche and incorporate it into their broad enterprise-wide systems.

A selection of the cognitive maps provides insights into both Trilogy and its competitors' perceptions of an emergent convergence of niches into larger enterprise-wide application suites.

DSA models and selected maps for the Trilogy case provide unusual insights into a very strong logic for setting innovative application development processes and cycles. Trilogy has quite structured and unusual requirements regarding the skills expected of developers, and has a special system of recruitment and training to instil shared vision and empowered decision-making at the individual developer level in projects.

The mapping of decision-making associated with transitioning from a “best-practice” product-driven business and application development approach to a vertical industry-focused approach, is a an extended contribution from additional insights gained from the interviews stages at Trilogy.

9.5 Contributions from the Trilogy Case Study to Practice

The Trilogy case study provides valuable insights into establishing “best-practice” application development groups. The establishment of a strong shared vision around programming and testing excellence through recruitment and training can be drawn from the Trilogy case.

The Trilogy case study provides some practical insights into transforming a software house from a strong product development focus across to industry-focused business groups and application development.

9.6 Hermeneutic Analysis – Trilogy Case Study

The Trilogy case study offers an application of the proposals described in Chapter 1 for advancing hermeneutic research in B2B decision-making.

Five levels of hermeneutic analysis were developed to address decision-making issues, events, and linkages in the Trilogy case study. These levels are presented in Exhibits 9.1 to 9.5.

The original case study by Austin (1998) serves as the etic 1 report of the case study – although there are elements of emic 1 reporting in the form of direct quotes from decision-makers in the account. Quotes and perspectives from decision-makers were encapsulated the Level 1 emic analysis in Exhibit 9.1.

Key case study issues as articulated by Austin (1998) were added to the emic 1 data in the Level II analysis and presented as etic 1 data in Exhibit 9.2. These issues included:

- High risk-taking to rapidly develop a software house capable of taking very large ERP competitors
- Very strong focus on application development excellence
- Trilogy rated itself as superior to Microsoft for application development
- Its application development approach focused on rapid development in small teams of superprogrammers specially developed by Trilogy to maximise quality and speed of application development

- Trilogy programmers were rewarded highly and encouraged to take risks with defined boundaries.

The Level II analysis also explored mental models and decisions/actions associated with these issues.

Exhibit 9.1 Level I Analysis of Trilogy Case.

Emic 1 view:

•Liemandt identified problems with hardware companies managing their selling and delivery processes

• Sales and Marketing costs typically represented over 40% total computer hardware vendor expenses – and were open to substantial automation

•Development of pricing configurator application for hardware companies

•Expanded to be a complete “front-end” enterprise application

•To be the best application development company in the world

•15 or 20 successful software companies could be built around the talent in Trilogy's development organization

•New philosophies on application development based on small teams of “superprogrammers”

•Deep concern about reaching “critical mass” to compete against large “back-office” ERP providers

↓

1991-98 Initial mental model:

•Focus on automation of sales & marketing activities

•Aim to cut hardware vendor Sales & Marketing Costs by 2 % per year

•Excellence in application development through using small teams of “super programmers” managing whole projects with frequent version builds and automated testing

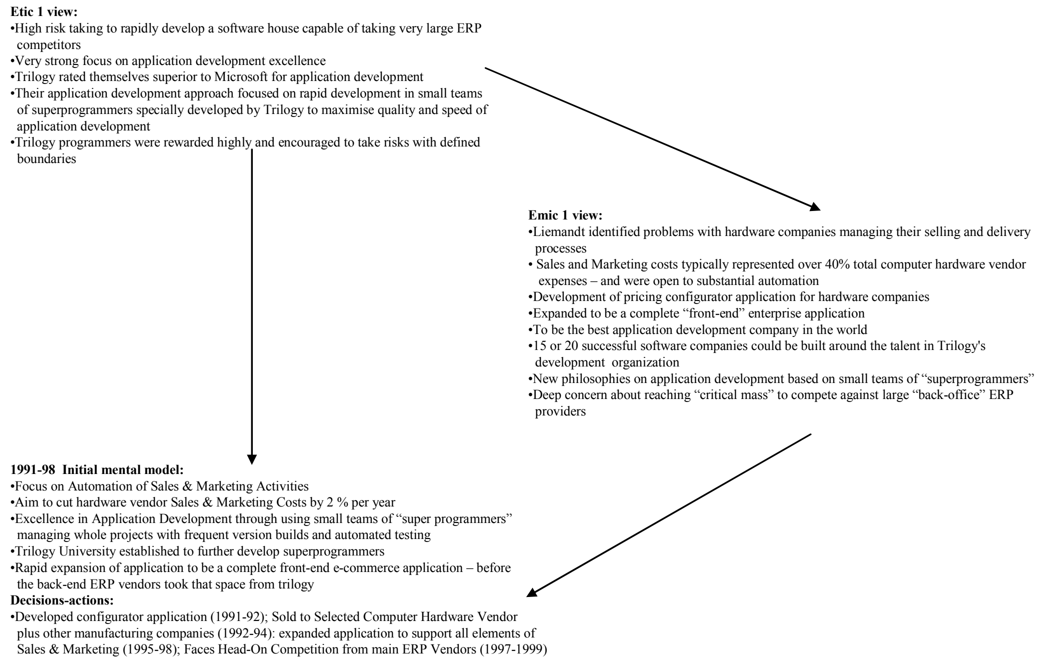
•Trilogy University established to further develop superprogrammers

•Rapid expansion of application to be a complete front-end e-commerce application – before the back-end ERP vendors took that space from Trilogy

Decisions-actions:

•Developed configurator application (1991-92); Sold to Selected Computer Hardware Vendor plus other manufacturing companies (1992-94): expanded application to support all elements of Sales & Marketing (1995-98); Faces Head-On Competition from main ERP Vendors (1997-99)

Exhibit 9.2 Level II Analysis of Trilogy Case.



The third level of analysis, as presented in Exhibit 9.3, is an etic 2 representation based on the current researcher's summarisation of event milestones and the emic 1 sensemaking views identified in the data in the Austin (1998) case study. Subsequently to developing Exhibit 9.3, emic 2 and etic 2 interpretations along with DSA and event maps provided revisions to Exhibit 9.3.

Austin's (1998) account focuses on the decision-making associated with identifying the automation and reduction of sales and marketing costs as a major unmet enterprise need and developing a comprehensive e-commerce application suite over time to address this need.

The account also highlights application development excellence of a core value in Trilogy's corporate culture. The account was sufficient for the development of an initial (etic 2) representative DSA model and to undertake further analysis on application development and delivery.

The initial DSA, event chronology, and cognitive maps reported below offer details supporting Exhibit 9.3. Etic 2 perspectives are mainly based from the initial DSA model, initial events chronology map, and initial cognitive maps.

Key etic 2 issues included:

- A company with an unusually strong emphasis on application development
- Focus on small team application development
- Not initially focused on Internet-based development but shifted quickly toward e-commerce in the late 1990s
- Focused on "anticipatory" competitive actions

The etic 2 DSA model and maps for the case were updated following extensive questioning of the accuracy and completeness of the original Austin's (1994) case – the collected data represents emic 2 material and was added during the fourth level of analysis (See Exhibit 9.4).

Thus, additional (emic 2) data were collected for etic 3 description and interpretation of the decision-making process as reported in the original case study, and for a period of four years beyond that reported by Austin (1998) and three years beyond Mandel & Austin (2000). Emic 2 data consists of responses from interviews with the then principal developer of Trilogy's SalesBuilder application, and the current (as early 2003) VP of Development at Trilogy (see Franke 2002a & 2002b; Hyams 2002a & 2002b).

Exhibit 9.5 is a fifth level of analysis with the addition of etic 3 data summarising emic 2 interpretations of mental models and events covered in the original case study as well as the work completed for the etic 2 interpretation – including the DSA, event chronology, and cognitive maps developed for the etic 2 interpretation.

The revised DSA, event chronology, and cognitive maps presented in subsequent sections below follow from the emic 2 interpretations, and these maps are part of the etic 3 interpretation.

Exhibit 9.3 Level III Analysis of Trilogy Case.

Etic 1 view:
•High risk taking to rapidly develop a software house capable of taking very large ERP competitors
•Very strong focus on application development excellence
•Trilogy rated themselves superior to Microsoft for application development
•Their application development approach focused on rapid development in small teams of superprogrammers specially developed by Trilogy to maximise quality and speed of application development
•Trilogy programmers were rewarded highly and encouraged to take risks with defined boundaries

Etic 2 view:
•A company with an unusually strong emphasis on application development
•Focus on small team application development
•Not initially focused on Internet-based development but shifted quickly toward e-commerce in the late 1990's
•Focused on "anticipatory" competitive actions

Emic 1 view:
•Liemandt identified problems with hardware companies managing their selling and delivery processes
• Sales and Marketing costs typically represented over 40% total computer hardware vendor expenses – and were open to substantial automation
•Development of pricing configurator application for hardware companies
•Expanded to be a complete "front-end" enterprise application
•To be the best application development company in the world
•15 or 20 successful software companies could be built around the talent in Trilogy's development organization
•New philosophies on application development based on small teams of "superprogrammers"
•Deep concern about reaching "critical mass" to compete against large "back-office" ERP providers

1991-98 Initial mental model:
•Focus on Automation of Sales & Marketing Activities
•Aim to cut hardware vendor Sales & Marketing Costs by 2 % per year
•Excellence in Application Development through using small teams of "super programmers" managing whole projects with frequent version builds and automated testing
•Trilogy University established to further develop superprogrammers
•Rapid expansion of application to be a complete front-end e-commerce application – before the back-end ERP vendors took that space from trilogy
Decisions-actions:
•Developed configurator application (1991-92); Sold to Selected Computer Hardware Vendor plus other manufacturing companies (1992-94): expanded application to support all elements of Sales & Marketing (1995-1998); Faces Head-On Competition from main ERP Vendors (1997-99)

Exhibit 9.4 Level IV Analysis of Trilogy Case.

Etic 1 view:
- High risk taking to rapidly develop a software house capable of taking very large ERP competitors
- Very strong focus on application development excellence
- Trilogy rated themselves superior to Microsoft for application development
- Their application development approach focused on rapid development in small teams of superprogrammers specially developed by Trilogy to maximise quality and speed of application development
- Trilogy programmers were rewarded highly and encouraged to take risks with defined boundaries

Emic 2 view:
- Validated Emic 1 views
- Transformation to Vertically Industry-Focused Divisions
- Trilogy application tailored by each Division for their key Customers
- Recruitment of Industry specialists and experts into Divisions
- Move to full front-end e-commerce positioning
- Consolidation away from Internet spin-off businesses back to industry focused enterprise customers

Emic 1 view:
- Liemandt identified problems with hardware companies managing their selling and delivery processes
- Sales and Marketing costs typically represented over 40% total computer hardware vendor expenses – and were open to substantial automation
- Development of pricing configurator application for hardware companies
- Expanded to be a complete "front-end" enterprise application
- To be the best application development company in the world
- 15 or 20 successful software companies could be built around the talent in Trilogy's development organization
- New philosophies on application development based on small teams of "superprogrammers"
- Deep concern about reaching "critical mass" to compete against large "back-office" ERP providers

Etic 2 view:
- A company with an unusually strong emphasis on application development
- Focus on small team application development
- Not initially focused on Internet-based development but shifted quickly toward e-commerce in the late 1990's
- Focused on "anticipatory" competitive actions

1991-98 Initial mental model:
- Focus on Automation of Sales & Marketing Activities
- Aim to cut hardware vendor Sales & Marketing Costs by 2 % per year
- Excellence in Application Development through using small teams of "super programmers" managing whole projects with frequent version builds and automated testing
- Trilogy University established to further develop superprogrammers
- Rapid expansion of application to be a complete front-end e-commerce application – before the back-end ERP vendors took that space from Trilogy

1998-2002 Extended mental model:
- Develop and Spin-off "Dotcom" businesses based on trilogy's applications
- Vertical Industry Focus with application development and delivery focused on Specific industry "roadmaps"

Decisions-actions:
- Developed configurator application (1991-92); Sold to Selected Computer Hardware Vendor plus other manufacturing companies (1992-94): expanded application to support all elements of Sales & Marketing (1995-98); Faces Head-On Competition from main ERP Vendors (1997-99); Rewrote all applications completely for Internet (1997-98); spun-off several Internet businesses (1999-2000); restructured and consolidated into Vertical industry divisions (2001)

Exhibit 9.5 Level V Analysis of Trilogy Case.

Etic 1 view:
•High risk taking to rapidly develop a software house capable of taking very large ERP competitors
•Very strong focus on application development excellence
•Trilogy rated themselves superior to Microsoft for application development
•Their application development approach focused on rapid development in small teams of superprogrammers specially developed by Trilogy to maximise quality and speed of application development
•Trilogy programmers were rewarded highly and encouraged to take risks with defined boundaries

Emic 2 view:
•Validated Emic 1 views
•Transformation to Vertically Industry-Focused Divisions
•Trilogy application tailored by each Division for their key Customers
•Recruitment of Industry specialists and experts into Divisions
•Move to full front-end e-commerce positioning
•Consolidation away from Internet spin-off businesses back to industry focused enterprise customers

Emic 1 view:
•Liemandt identified problems with hardware companies managing their selling and delivery processes
• Sales and Marketing costs typically represented over 40% total computer hardware vendor expenses – and were open to substantial automation
•Development of pricing configurator application for hardware companies
•Expanded to be a complete “front-end” enterprise application
•To be the best application development company in the world
•15 or 20 successful software companies could be built around the talent in Trilogy's development organization
•New philosophies on application development based on small teams of “superprogrammers”
•Deep concern about reaching “critical mass” to compete against large “back-office” ERP providers

Etic 2 view:
•A company with an unusually strong emphasis on application development
•Focus on small team application development
•Not initially focused on Internet-based development but shifted quickly toward e-commerce in the late 1990’s
•Focused on “anticipatory” competitive actions

1991-98 Initial mental model:
•Focus on Automation of Sales & Marketing Activities
•Aim to cut hardware vendor Sales & Marketing Costs by 2 % per year
•Excellence in Application Development through using small teams of “super programmers” managing whole projects with frequent version builds and automated testing
•Trilogy University established to further develop superprogrammers
•Rapid expansion of application to be a complete front-end e-commerce application – before the back-end ERP vendors took that space from Trilogy
1998-2002 Extended mental model:
•Develop and Spin-off “Dotcom” businesses based on trilogy’s applications
•Vertical Industry Focus with application development and delivery focused on Specific industry “roadmaps”
Decisions-actions:
•Developed configurator application (1991-92); Sold to Selected Computer Hardware Vendor plus other manufacturing companies (1992-94): expanded application to support all elements of Sales & Marketing (1995-98); Faces Head-On Competition from main ERP Vendors (1997-99); Rewrote all applications completely for Internet (1997-98); spun-off several Internet businesses (1999-2000); restructured and consolidated into Vertical industry divisions (2001): Trilogy promotes focus on software productivity and attempts to link to ROI through customer programs (2002-03); Trilogy releases FCT Methodology as a product/consulting service (2003)

Etic 3 view:
•Etic 2 views validated or extended
•Transformation To Vertical Industry-Focused Divisions
•Trilogy application development philosophies maintained in restructured Divisions
•Trilogy actually turned their application development philosophy into a Product/Methodology!
• Can Trilogy's’ philosophy of application development excellence be successfully applied to developing and managing industry-specific applications?
•Trilogy continues to battle with the ERP Vendors - sometimes in competition, sometimes in partnership
•New focus by Trilogy on ROI from software productivity encourages measurement of effectiveness of using front-end applications

The new data from the emic 2 and etic 3 rounds of interpretation validated etic 2 data and highlighted new insights related to:

- Transformation to vertical industry-focused divisions
- Trilogy application development philosophies maintained in restructured divisions
- Trilogy actually turned its application development philosophy into a Product/Methodology!
- Can Trilogy's philosophy of application development excellence be successfully applied to developing and managing industry-specific applications?
- Trilogy continues to battle with the ERP vendors - sometimes in competition, sometimes in partnership
- New focus by Trilogy on ROI from software productivity encourages measurement of the effectiveness of using front-end applications

Overall, the five-level hermeneutic framework for the Trilogy case study provided a strong and comprehensive framework for capturing and extending dynamic sensemaking for Trilogy, for both the development of a pricing configurator application into a comprehensive front-end e-commerce application suite, in tandem with the rapid development of a small business into a robust medium-sized enterprise.

9.7 DSA Model

The original DSA model was developed from a review of the whole Austin (1998) case study. The supporting note by Mandel & Austin (2000) was dropped from the analysis as it only addressed a list of new business ventures and not application development and delivery.

The section in Austin’s (1998) case study under “Developing the Product” was the main source for developing a DSA model (see Austin 1998, 7-8). The account discusses development philosophies, issues and approaches that could be regarded as distinctive to Trilogy. The original DSA model is presented in Figure 9.2 and is representative of Trilogy application development during 1996-99.

The application development commenced with the allocation of a “Product Idea” to a project team (box 2). A distinctive feature of Trilogy’s application development cycle is the concept of very small projects composed of 1-2 “Supercoders” (box 3). Trilogy’s management believed that one to two highly experienced and motivated programmers could do the work of 10-20 standard programmers.

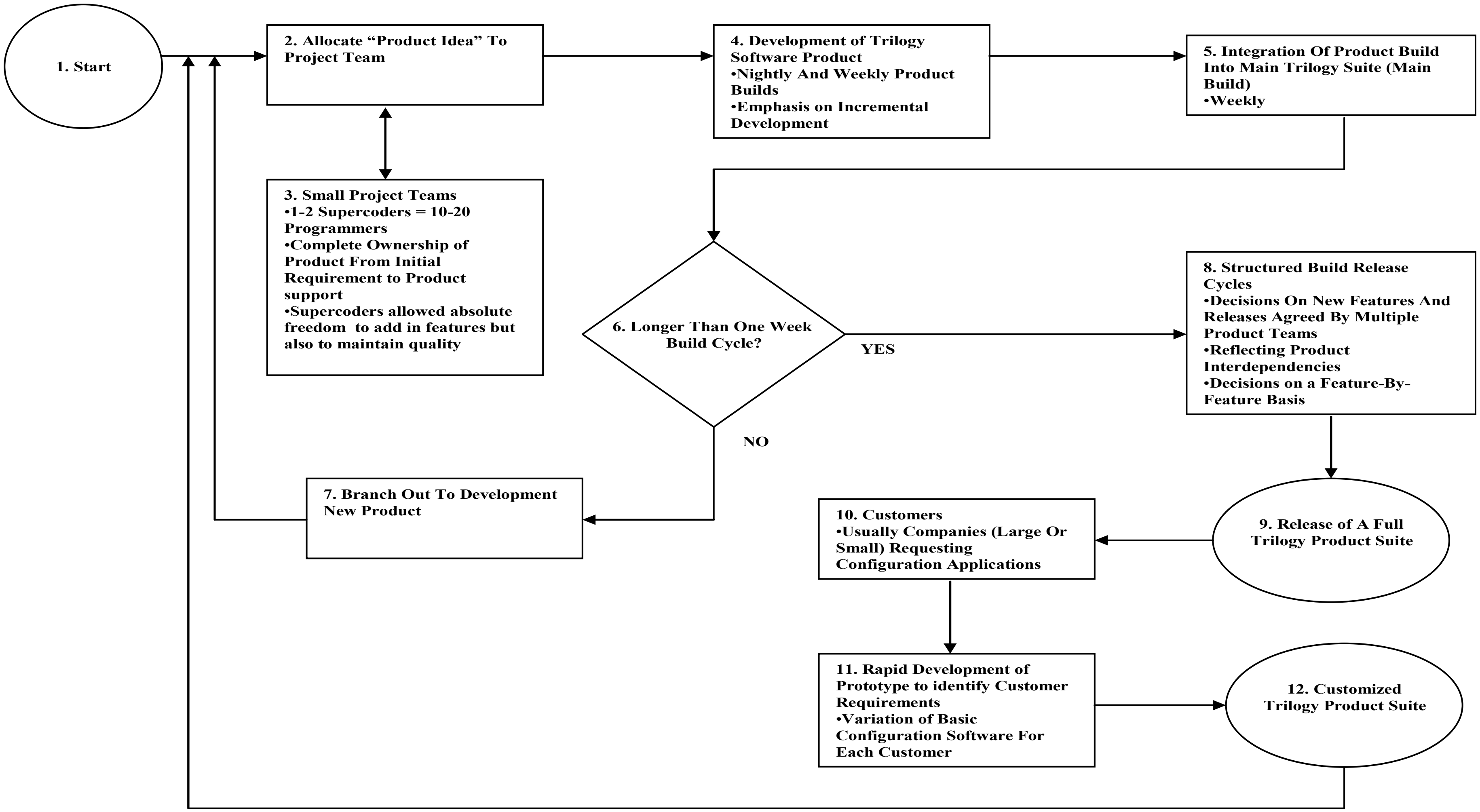


Figure 9.2 Summary DSA Model For Trilogy Application Development (1998)
Created from data in Austin (1998)

These Supercoders were given complete ownership of a project right through the whole application development cycle. Trilogy also believed in high levels of remuneration, bonuses and compensation, and ongoing training for the Supercoders, who were also accorded superstar status within the organisation.

Components of the application were built or released as new development versions either on a nightly or weekly frequency. Updates to the development version were usually incremental (box 4). Components were integrated into the application weekly (box 5).

A significant decision-point each week for the project team was whether a component or application would be likely to take longer than a weekly build cycle (box 6). If "NO" then the project team(s) can go onto new product development (box 7).

If "YES" then the project teams developed structured build release cycles incorporating decisions relating to new features and new releases (box 8). A set of collective decisions were made among a number of project teams working on application components or full applications, understanding that there were significant interdependencies between their projects. It is likely that this collective decision-making approach was actually incorporated into the application development cycle right from the commencement of the cycle.

A full Trilogy Product Suite was released (box 9). However, that was not the end of the development cycle. Customers often required major customisation of Trilogy's applications into their own configuration systems (box 10).

A significant number of these customer requests added up to an extension of the application development cycle, whereby Trilogy rapidly prototyped a variation of the basic configuration software tailored for a customer request (box 11).

The customised application was in effect a new version of the application and could be regarded as a completion point in the application development cycle (box 12).

Trilogy's application development cycle was supported by a rapid decision-making system with a mix of very small groups (one to two coders) working in concert, with high collective empowerment, with a vision of an integrated application suite. There was provision for rapid customisation beyond the basic application, which is worthwhile being viewed as part of the overall application development cycle.

Interviews with Franke (2002) and Hyams (2002) confirmed the DSA model for Trilogy without any modifications for Trilogy's application development cycles up to 1999.

9.7.1 New DSA Model (2001-02)

The interviewees for the Trilogy case requested that a new DSA model be created for Trilogy's application development cycles from 2001-02. The new DSA model reflects Trilogy's commitment to vertical (industry) focused business management and product development.

The interviewees provided substantial insights in the development of the new DSA model. Detailed product development cycle information was drawn from company documents on Trilogy Product Management Methodology, and is used but not directly referenced, with permission from the interviewees at Trilogy. These inputs were compiled into a vignette which is presented in Exhibit 9.6 with the new summary DSA model presented in Figure 9.3

Exhibit 9.6 Trilogy's Industry-focused Application Development

TRILOGY GOES VERTICAL – INDUSTRY-FOCUSED ENTERPRISE SOLUTIONS

In 2000, Trilogy restructured from a horizontally focused application vendor toward using a Software Development Methodology designed to quickly conceptualise, create and deliver highly verticalised applications. Previously, from 1993-1999, Trilogy's application development strategies were based on Trilogy Product Roadmaps.

The Executive Team established an Operations Group with five vertical divisions focused on key enterprise customers and products:

* Automotive
* Computer
* Telecommunications
* Financial Services
* New Business

Exhibit 9.6 Trilogy's Industry-Focused Application Development (Cont'd)

Trilogy has a two-dimensional matrix of management and delivery resources, located in Core company-wide Groups and within each Division with deep industry experience. These resources cover:

* General Managers plus Functional Vice-Presidents
* Development Staff (Division-Specific)
* Consulting
* Human Resources
* Finance
* Solutions Marketing and Business Development
* Product Management and Presales Resources

Trilogy's Fast Cycle Time (FCT) Software Development Methodology is now based on delivering vertical applications within Industry Maps set up for each division for a two-year timeframe. There are four phases in this methodology:

1. Product Ideation

During this phase, Trilogy's Division Managers with their Solutions Marketing and Product Marketing teams generate ideas for products, creates a vision for the product and the Industry, applies an investment justification process to the vision and product, and develops an Industry Roadmap supported by a Business Case. All product conceptualisation and development is driven with a vertical industry focus.

2. Product Planning

During this phase, the capabilities of the product will be identified and sequenced for delivery. Product features will then be expanded into actual development schedules including actual release features and initial estimates at release dates. During this phase, Product Management takes over with Development Teams in driving product planning.

3. Product Development

This phase consists of gathering the detailed requirements for the product, and commencing development using Trilogy's Fast Cycle Time methodology. Although Product Management and Development Teams are driving the product development process, chartered customers associated with specific Trilogy divisions may be involved in testing early versions of the product.

Exhibit 9.6 Trilogy's Industry-Focused Application Development (Cont'd)

4. Product Delivery

Product Delivery is defined as beyond beta Release. There is extensive interaction between Product Management, Development Teams, charted customers, and solutions marketing groups. The final product will be delivered during this phase. Once in production, the product will be maintained with patches, bug fixes and sub-releases. Consultants may work with chartered customers in deployment of new products.

An extended version of steps in Trilogy's Software Development Cycle is outlined below:

1. Identify and Define Product Concepts (Product Ideation)
* Industry Analysis
* Analyse Against Current Trilogy Industry/Product Maps
* Driven by Solutions Marketing/ Product Management; some input from Division General Management
2. Evaluate Product Concepts (Product Ideation)
* Product Vision and Product Concept
* Validate Product Concepts Against Industry Maps
* Develop Business Case
* Driven by Solutions Marketing/ Product Management; Division General Management
3. Sequence Capabilities and Define Product Features (Product Planning)
* Define and Sequence Product Capabilities
* Product Features against Industry Maps
* Driven by Product Management with input from Development Teams
4. Create Release Definition (Product Planning)
* Determine Release Features and Scope
* Determine Resource and Time Estimates
* Define Release Review Cycles
* Agreed and Finalised Between Product Management; Development Teams; Division General Manager; Some input from Nominated "Charter Customers"

Exhibit 9.6 Trilogy's Industry-Focused Application Development (Cont'd)

5. Define Release Requirements (Product Development)
* Identify Potential Product Requirements
* Prioritise Based On Business Value, & Decide on Release Contents
* Requirements Inputs (From Consultants
* Development Release Plan
* Undertaken by Product Management; Development Teams; some input from nominated "Chartered Customers"
6. Develop Product (Product Development)
* Iterative Development Via Feedback and Short Development Cycles
* Product Locked Once Released to beta
* Iterative Updates
* Develop Launch Development Plan at beta Stage
* Active Involvement from Product Management; Development Teams; Solutions Marketing (For Product Roll-out and Marketing); feedback from nominated "Chartered Customers" associated with testing
7. Release Product (Product Delivery)
* Product Release Certification Process
* Certification For Specific Platforms
* Marketing Launch
* Driven by Development Team; Solutions Marketing
8. Monitor Product (Product Delivery)
* Support including Bug fixes and Minor Enhancements
* Feedback From Installed Base
* Set Up For New Products And/Or Next Release
* Directed by Product Management with product fixes and sub-releases completed by Development Team

This vignette was prepared from: Hyams, Chris (2002a & 2002b).

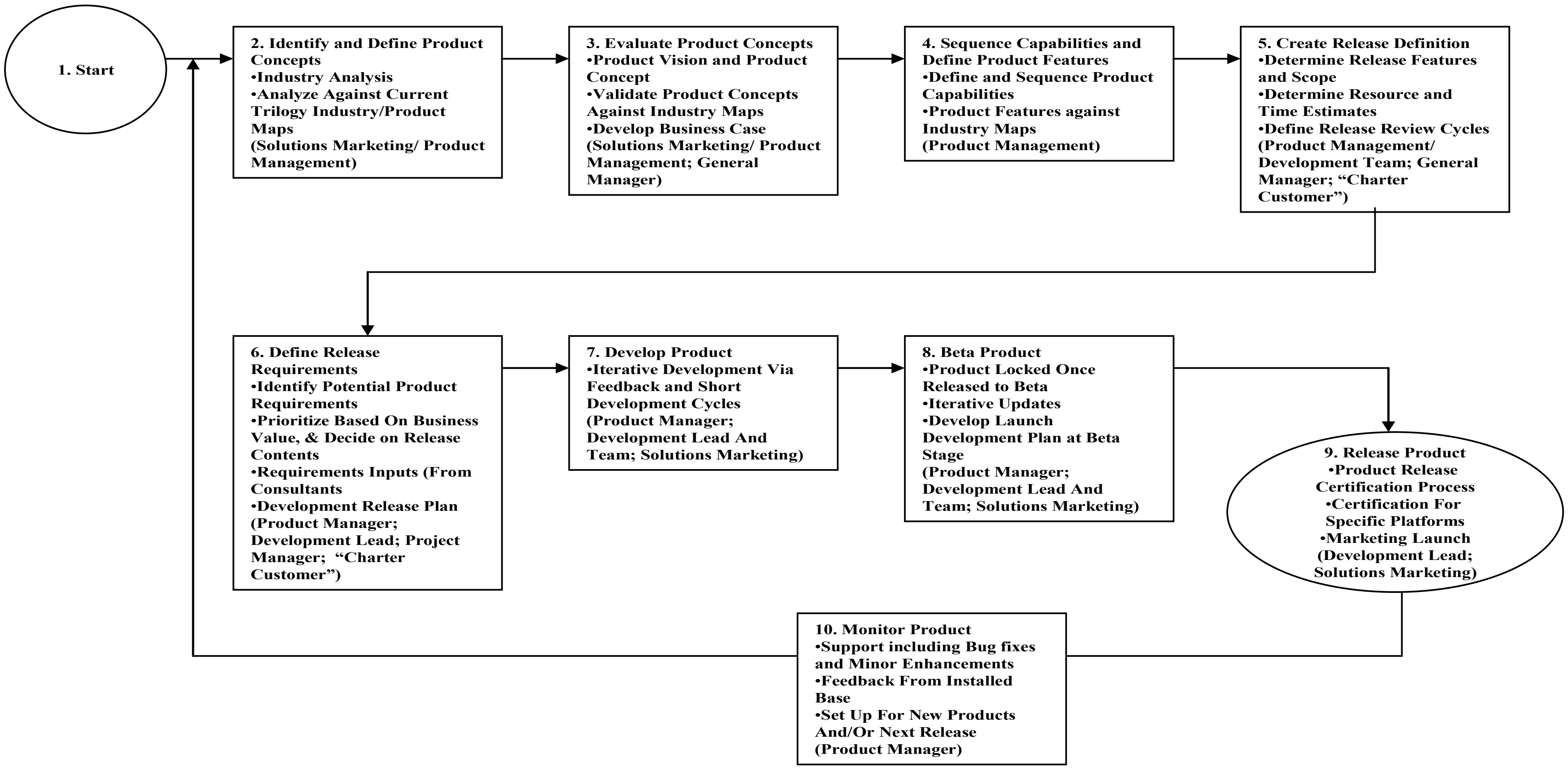


Figure 9.3 New Summary DSA Model For Trilogy Application Development (2001-02)
Created from data in Austin (1998); Franke (2002); Hyams (2002).

Trilogy's industry-focused approach to application development is evident at the commencement of the application development cycle as all product concepts that are identified and defined are subjected to rigorous industry analysis (see box 2). Solutions Marketing and Product Management are responsible for the decision-making associated with mapping concepts against existing Trilogy industry and product maps.

Industry Group General Managers with Solutions Marketing and Product Management conduct further validation of concepts against industry maps and develop business cases for the concepts (box 3).

Product concepts are transformed into actual product features and capabilities by Product Management (box 4). The defined product features are validated against industry maps, reinforcing that each of the Trilogy industry-based business divisions are directly involved in specific application development.

Several groups are involved in creating a Release Definition, which includes release features and scope, resource and time estimates, and release review cycles (box 5). Product Management and the Project Development Team will create the actual documentation and content, but will consult closely with the relevant industry group General Manager for specific release dates. Selected "Chartered Customers" will also be consulted, particularly with regard to product features and scope.

Trilogy's application development cycle has a specific step for the definition of release requirements which has more detail and further prioritisation of product features than the release definition (box 6).

There is an additional assessment of product features based on business value, and inputs from Trilogy and other consultants. Output from these assessments are then finalised into a Development Release Plan. Product Managers working with the Project Team and "Chartered Customers" drive the finalisation of the Development Release Plan.

The actual product development phase is iterative and through short development cycles similar to the weekly/monthly build approach in the original DSA model (box 7). Product Management work with the Project Team, but also include Solutions Marketing in this phase.

The direct involvement of industry-based solutions marketing persons in the actual product development phase is a distinctive feature of Trilogy application development cycle, when compared with other software houses in this project.

There is a distinct beta product milestone where product features are locked in and a Launch Development Plan is developed (box 8). There are still iterative updates based on testing the product at this stage. Product Management, with the Project Team and Solutions Marketing, are all active in beta testing and creating the Launch Development Plan.

Trilogy has formal processes for product release (box 9). Trilogy's products are formally certified for actual release and for specific platforms (specific operating systems and/or vendor's computer systems). There is also a formal marketing launch for the product release. Certification processes are driven by the Development Teams, while Solutions Marketing drives the Marketing Launch.

Trilogy has formal post-release processes for product monitoring, including bug fixes and minor enhancements (box 10). Customers provide feedback on product performance and recommended fixes. Product Managers oversee product monitoring and take inputs for setting up the next application development cycle.

The new DSA model represents a significant departure from the original DSA model in that it recognises that almost all application development is undertaken within the Business Divisions and is focused right from conception on the industry map. Business requirements are focused within that Division, rather than being a generic application that was customised at the end of the application development cycle.

There are still some application development inputs from Trilogy executives in charge of development, but that is rapidly contextualised into Industry/Division-Specific decision-making processes.

The new DSA model still reflects an internal focus for applications required for Trilogy's own application development. However, there is significant third-party interaction between Trilogy and selected "chartered customers" and consultants.

9.8 Events Chronology Map

Austin's (1998) "Trilogy (A)" case study provides a chronology of the genesis and rapid growth of Trilogy from 1990 through to 1998-99. An events chronology map was created incorporating key events during this period (see Figure 9.4). Additional symbols were employed for the map including the flowchart symbol for a delay or problem and a six-pronged box for a solution to the delay or problem.

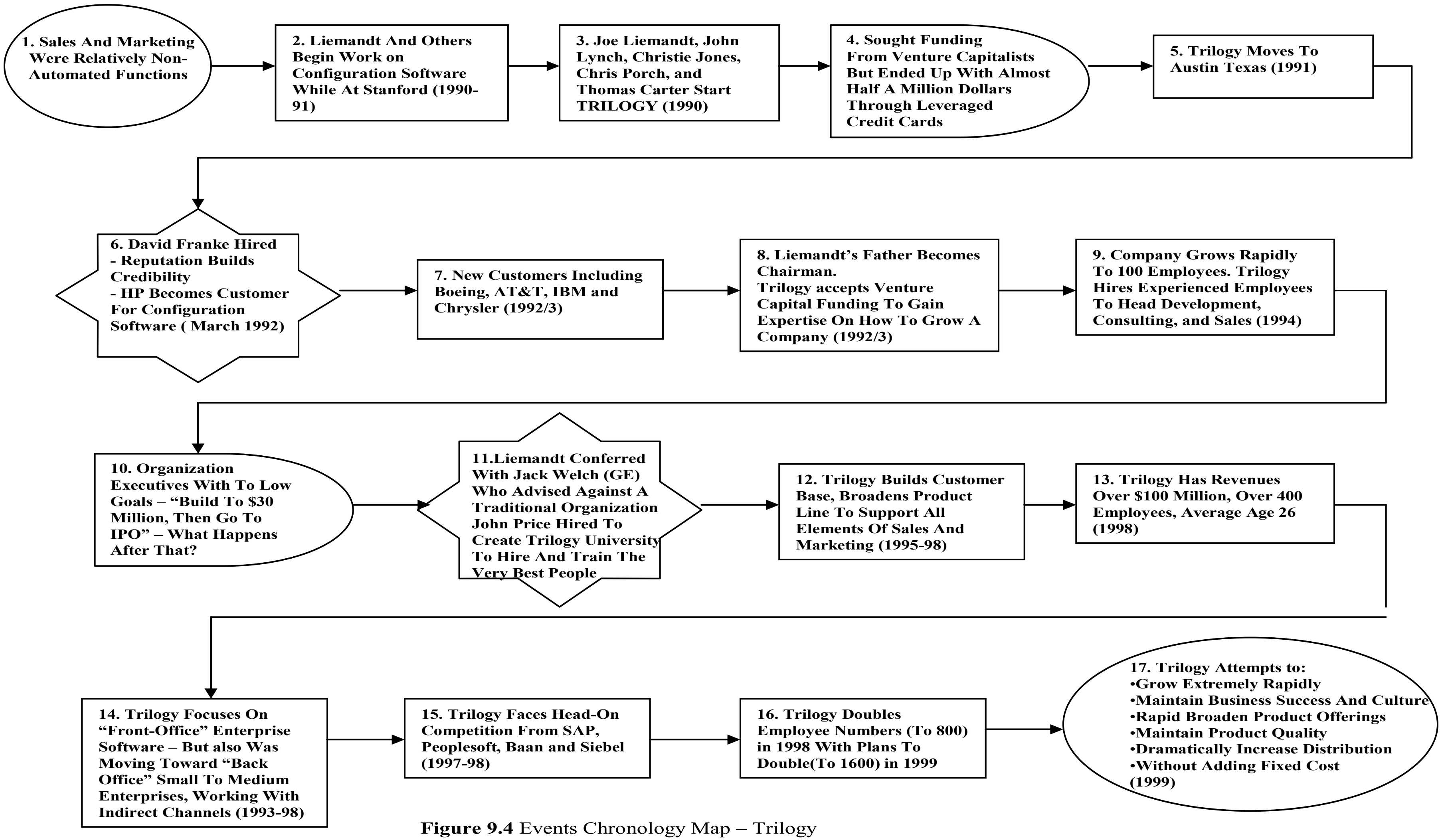


Figure 9.4 Events Chronology Map – Trilogy
Created from data in Austin (1998)

Joe Liemandt recognised that, compared to other functions with large computer hardware companies, sales and marketing were relatively non-automated functions (box 1). Liemandt believed that software to assist with the configuration of complex computer systems could greatly contribute to cutting costs in the sales and marketing functions.

While at Stanford University, Liemandt, with some other students, began work on configuration software in 1990-91 (box 2). Liemandt, Lynch, Jones, Porch and Carter set up Trilogy in 1990 (box 3). They sought funding from venture capitalists but were regarded as too young, and ended up leveraging their own credit cards up to almost half a million dollars. (box 4).

During 1991, Liemandt decided to move the new company to Austin, Texas, to be closer to his ill father (box 5). The move enabled Liemandt to more easily hire David Franke, who was an experienced and well-known software developer (box 6). Franke's strong reputation in software development was a contributor to Trilogy signing up Hewlett-Packard (HP) as a customer in March 1992.

Trilogy rapidly built on its success with providing a configurator solution for HP, through signing up new customers including Boeing, AT&T, IBM and Chrysler in 1992-93 (box 7).

Trilogy gained extra business development knowledge through the appointment of Joe Liemandt's father, Greg, as Trilogy's Chairman in 1992. Greg Liemandt had recently been a senior executive with GE and a Chairman of UCCELL. Trilogy turned down several venture capitalists' offers of funding, but chose other specific VCs to gain expertise on how to grow a company rapidly (box 8).

By 1994, Trilogy had grown to 100 employees, having hired experienced development, consulting and sales managers (box 9). However, Trilogy's executives were struggling with establishing reasonable growth targets for the company – and when to go public (or to IPO), and where the company might go after an IPO (box 10). Through his father's links with GE, Joe Liemandt was able to confer with Jack Welch the well-known CEO of General Electric (GE) about these issues (box 11).

Welch advised that Trilogy not establish a traditional organisation structure, but to focus on hiring the best people to develop the best products. John Price was hired to set up the Trilogy University that would hire and train high-quality developers.

Over the three years 1995-98, Trilogy built a substantial customer base, while broadening its product line beyond configuration to cover all elements of sales and marketing (box 12). By 1998, Trilogy was generating revenues of over $100 million, with over 400 employees whose average age was just 26 (box 13).

Also, through the period 1993-98, Trilogy moved toward end-to-end solutions, working with indirect channels partners such as system integrators (box 14). Expansion from “front-office” into “Back-Office” activities put Trilogy on a collision course with ERP vendors including SAP, PeopleSoft, Baan and Siebel (box 15).

During 1998, Trilogy’s employee numbers doubled and were expected to double again in 1999 (box 16). Austin (1998) highlights several issues facing Trilogy’s executive team for 1999, around continued rapid growth, broadening product offerings, maintaining product quality, and maintaining business success and a risk-taking development culture (box 17).

Trilogy also needed to dramatically increase distribution of its products without adding fixed costs to its overall business model. It seemed that Trilogy had to stay on an exponential growth curve forever.

9.8.1 Updated Events Chronology Map

The original events chronology map was validated during the interview stage without modification up to 1999, and includes reference to Mandel & Austin’s (2000) Trilogy (B) case study. The map timeframe was extended to 2002, and, an updated events chronology map is presented in Figure 9.5.

During 1998-99, Trilogy strove to extend its product range toward a full enterprise-wide commercial-off-the-shelf (COTS) e-commerce suite (box 18), through the establishment of an indirect business unit. Trilogy attempted to partner with developers to rapidly extend its application suite. Trilogy also attempted to tailor its packages for more easier use by developers. Vantage was noted as a key partner in 1998-2000.

Trilogy established various Internet spin-off businesses in car ordering, insurance ordering, appliance ordering and college hiring from 1999-2000 (box 19). These spin-offs leveraged off Trilogy’s configuration application knowledge, or developer training skills, or from enterprising Trilogy programmers and executives. who wished to branch out into new business ventures.

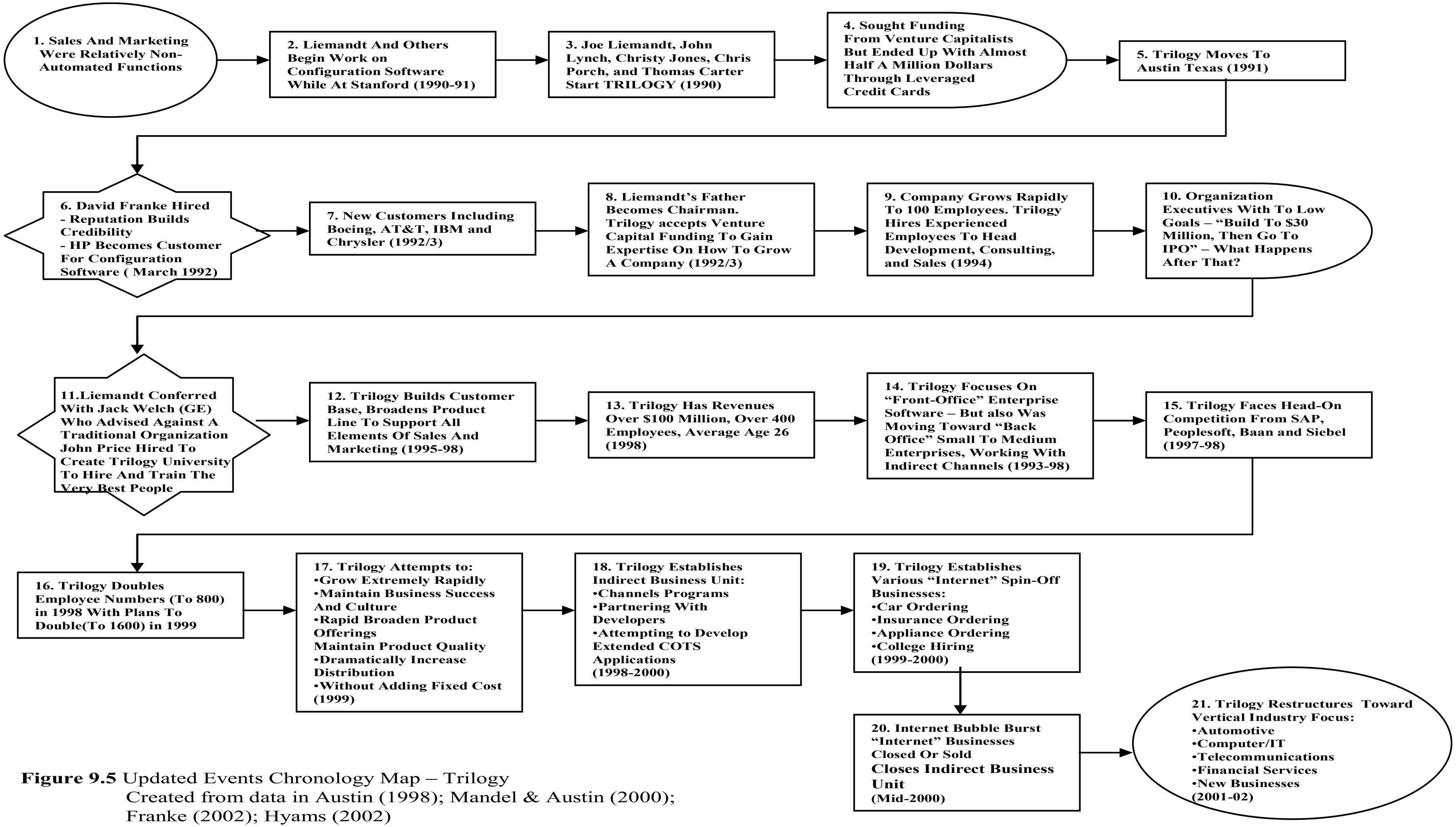


Figure 9.5 Updated Events Chronology Map – Trilogy
Created from data in Austin (1998); Mandel & Austin (2000); Franke (2002); Hyams (2002)

However, by mid-2000, with the Internet "bubble" bursting, Trilogy closed or sold most of its Internet businesses (box 20). The indirect business unit was also abandoned, mainly because of the complexity of attempting to establish an extended COTS application suite. This experience reinforced a view within the company that developing a general COTS application suite just like the ERP suppliers was definitely not a path that Trilogy should travel on.

Trilogy's own growth was halted and the company undertook major restructuring and consolidation into 2001. Trilogy's response to a falling information technology market was to restructure toward a vertical/industry focus in 2001-02, creating five Business Divisions (box 21).

Each Division manages its own application development, consulting, marketing and its own business functions. The five Business Divisions created include Automotive, Computer/IT, Telecommunications, Financial Services, and New Business.

Several issues emerging from the original and updated events chronology maps are explored further though cognitive maps.

9.9 Cognitive Maps

Cognitive maps created for the Trilogy case study were designed to complement DSA models and events chronology maps through additional exploration of selected decision-making issues and contexts.

All cognitive maps were created using the following procedures:

1. Selected extracts from Austin's (1998) case study were extracted to text files
2. One text file version of each extract was processed through the TACT package to produce a complete word-frequency list
3. Word frequencies were reduced through one or two iterations to ranked lists of words with frequencies ranging from over two to three words
4. A second version of the text file was input to NVivo
5. Ranked words were input to NVivo and coded as nodes in association with the text file
6. The overall extract was deconstructed into selected phases and sentences and coded directly as nodes through the InVivo tool within NVivo
7. The nodes associated with each extract were input to the Model Explorer and mapped using links incorporating codes of association plus addition node "items"

8. Several iterations of dynamic mapping were undertaken with the final cognitive map exported from NVivo to PowerPoint
9. Each PowerPoint version of the cognitive map was reformatted into a more readable and project-consistent format
10. The revised PowerPoint cognitive maps were presented to interviewees and validated or modified
11. The updated PowerPoint cognitive maps incorporated changes after the interview process.

Three sections were extracted from Austin's (1998) case study for further text analysis and creation of cognitive maps to complement the analysis presented in the original DSA model and original events chronology map.

The three initial cognitive maps created for the Trilogy case study addressed the following themes:

9.9.1 Early Development (1991-94)

9.9.2 Industry and Competition

9.9.3 Software Development

During the interview process, requests were put forward for an additional cognitive map reflecting updated insights from 2000-01:

9.9.4 Industry-Focused Application Development (2000-01)

The first cognitive map is an extended insight into the creation and early growth of Trilogy.

The second cognitive map provides insights into industry competition. Part of the second map parallels Trilogy's early growth, but about half of the map focuses on Trilogy's competition as it rapidly grew between 1993 and 1999.

The third cognitive map amplifies logic and philosophy behind Trilogy's distinctive application development methodology for 1993-99, but is also relevant through to 2002.

The additional cognitive map highlights insights into Trilogy's industry-focused application development in 2001-02.

9.9.1 Cognitive Map 1 – Early Development (1991-94)

Development details for the first cognitive map include:

- Extract section, size and exhibit:

- “Company Background” section, Austin (1998, pp. 2-3); 1¾ page; Exhibit 9.7

- TACT word frequency lists and comments on key words:
    - First pass: 29 words with frequency > 2
    - Second & final pass: 18 words with frequency > 3
    - Final ranked word list: Table 9.1
    - There was a diverse range of words in the final list with the top words being related to the *Trilogy Company* and *Liemandt*, but no real pattern in the remaining words in the list (italicised words are in the word list).
- NVivo ranked word, sentence and phrase node count = 123
- Association of concepts coding used: +, -. +/-
- NVivo PowerPoint cognitive map: Figure 9.6
- Reformatted PowerPoint cognitive map: Figure 9.7

The NVivo cognitive map covers insights into the identification of an opportunity in 1990 to automate activities within sales and marketing functions, onto addressing the opportunity through the development of a configurator application, and the creation of a start-up company. The map also provides insights into rapid growth of Trilogy through to 1994.

Exhibit 9.7 Text Extract – Trilogy - Early Development (1991-94)

Liemandt had come to Stanford knowing he wanted to start a software company. As an undergraduate, he had spent many hours researching the industry and thinking about where the best opportunities might be. In and around school work, he did consulting to pay bills and to stay on top of the latest in information technology practices. Eventually, experiences from consulting clicked with the research.

It seemed to Liemandt that hardware vendors had difficulty delivering their products with the right equipment. His consulting clients frequently received computers with missing or incompatible components. Selling and delivery processes for these complex products appeared to be largely manual and fraught with error. This observation prompted Liemandt to analyse the income statements of computer product companies, comparing spending patterns with the degree of automation in each company function. What he found surprised him (see Exhibit 1).

Companies typically spent only 8 to 10 per cent on General and Administrative costs, which had been extensively automated over the years. Research and Development, also highly automated in most high-tech companies, accounted for slightly more spending, about 10 to 15 per cent. Manufacturing was a similar story: mature cost saving technologies had been applied to reduce spending (less cost of goods sold) to 20 to 25 per cent of the firm's total expenses. What remained after deducting these major categories was the more than 40 per cent of expenditures that were mostly related to Sales and Marketing-an area which, surprisingly, was not very automated. If automation of the "selling chain" allowed companies to put an additional 2 per cent of revenues on the bottom line (a number which Trilogy now considers reasonable), that would be worth literally billions of dollars. It was a potentially huge market that had not yet been targeted by anyone.

While still in school, Liemandt and the others had begun working on configuration software, which incorporated complex if-then rules into a tool that would prevent mismatches between incompatible product parts. They continued this work after school, into 1991, always sure they were on the verge of solving the configuration problem that would finally give them a completed product. Companies like Hewlett-Packard (HP) and Digital were working on their own "configurators," which added urgency to Trilogy's efforts. Liemandt presented their ideas to venture capitalists, but none would invest in a company composed entirely of barely 20-years-olds. To stay afloat, the team leveraged more than 20 credit cards, managing to borrow almost half a million dollars in cash advances. Liemandt describes the mood in the days before the company had revenues or product:

Exhibit 9.7 Text Extract – Trilogy - Early Development (1991-94) (Cont'd)

3 Some of the factual material in this section was found in "Dream On," by Karen E. Starr, Selling Power, October, 1997, Vol. 17, No. 8; and "Holy Cow, No One's Done This!" by Josh McHugh, Forbes, June 3,1996.

Trilogy (A) 699-034

At the beginning, nothing worked. We lived failure every quarter. The product never worked. We'd sit around thinking "this is just ridiculous, why are we continuing?" We were living in a state of failure, but we had this hope, this shining star that wouldn't go out. What kept us together was the vision that this was a huge opportunity; we just needed to make it work. That, or the fact that we were yelling at each other the whole time.

In 1991 the company moved to Austin, Texas so that Liemandt could spend more time with his father, Gregory Liemandt, who had been diagnosed with a fatal illness. By this time Trilogy had an early product and had applied for patents covering its algorithms, but the company still had no customers. They were working hard to generate interest in their software, but nothing was working. HP, a key potential customer, sent a particularly discouraging letter saying, in essence, "we already have a configurator and don't need your product."

Meanwhile, however, being in Austin enabled Trilogy to hire David Franke, a software developer with an industry-wide reputation, from a research consortium in Austin. With Franke on board, the company suddenly had new-found credibility. Silicon Graphics became the first customer, signing a small deal. Within months, HP was back, this time offering $3.5 million for software and support services. The deal was consummated in March of 1992. At the time, Trilogy had eight employees.

When HP signed, everything changed for Trilogy. Software that was good enough for HP was good enough for a lot of other big companies, also. The floodgates opened and Boeing, AT&T, and, eventually, IBM and Chrysler became customers (the IBM deal alone was worth $25 million). Also significant: Liemandt's father, a former GE executive and chairman of UCCELL, who had called his son a moron for squandering his Stanford education, agreed to become Trilogy's chairman, a position he retained until he passed away in 1993. Trilogy accepted funding from two venture capitalists, not because the company needed the money, but because it needed the expertise on how to grow a company that those firms could offer. Venture investors who had refused to fund Trilogy in the early days came calling-and were turned away. Liemandt retained more than 50 per cent ownership.

Exhibit 9.7 Text Extract – Trilogy - Early Development (1991-94) (Cont'd)

As orders rolled in, Trilogy staffed up. The company grew rapidly to around 100 employees. They hired experienced executives to head Development, Consulting, and Sales. But by late 1994, Liemandt was not happy with the way things were going. Things were good at present, but he worried about the future:

Source: Austin 1998, pp. 2-3.

Table 9.1 Trilogy – Early Development (1991-94) - Ranked Word List

Anchor Concepts	**Frequency**
company	16
Liemandt	11
trilogy	11
work	9
product	8
hp (Hewlett-Packard)	6
percent	6
software	6
automate	5
more	5
time	5
would	5
configuration	4
consulting	4
customer	4
need	4
research	4
spend	4

Source: data analysis performed using TACT (1997)

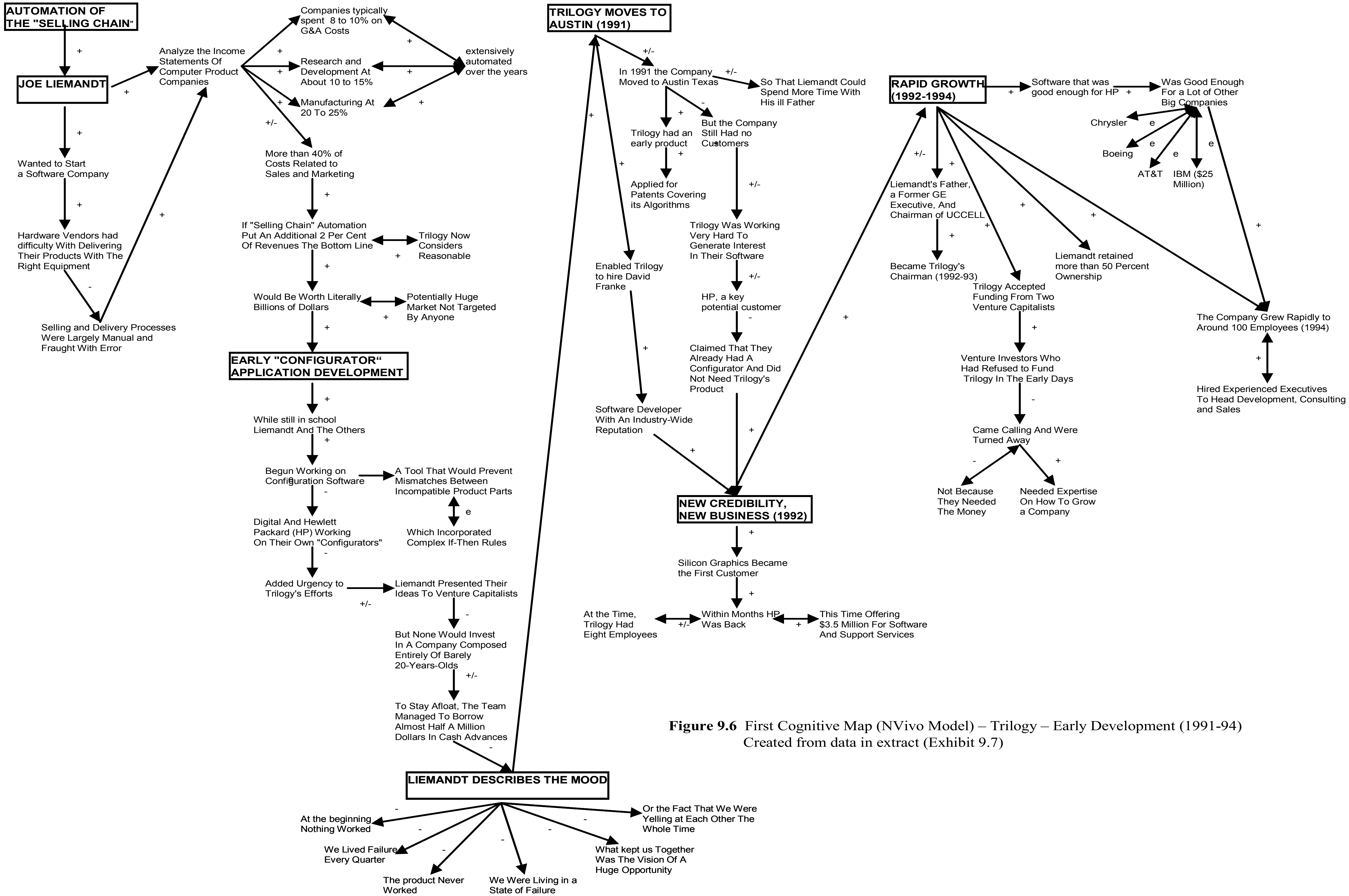


Figure 9.6 First Cognitive Map (NVivo Model) – Trilogy – Early Development (1991-94)
Created from data in extract (Exhibit 9.7)

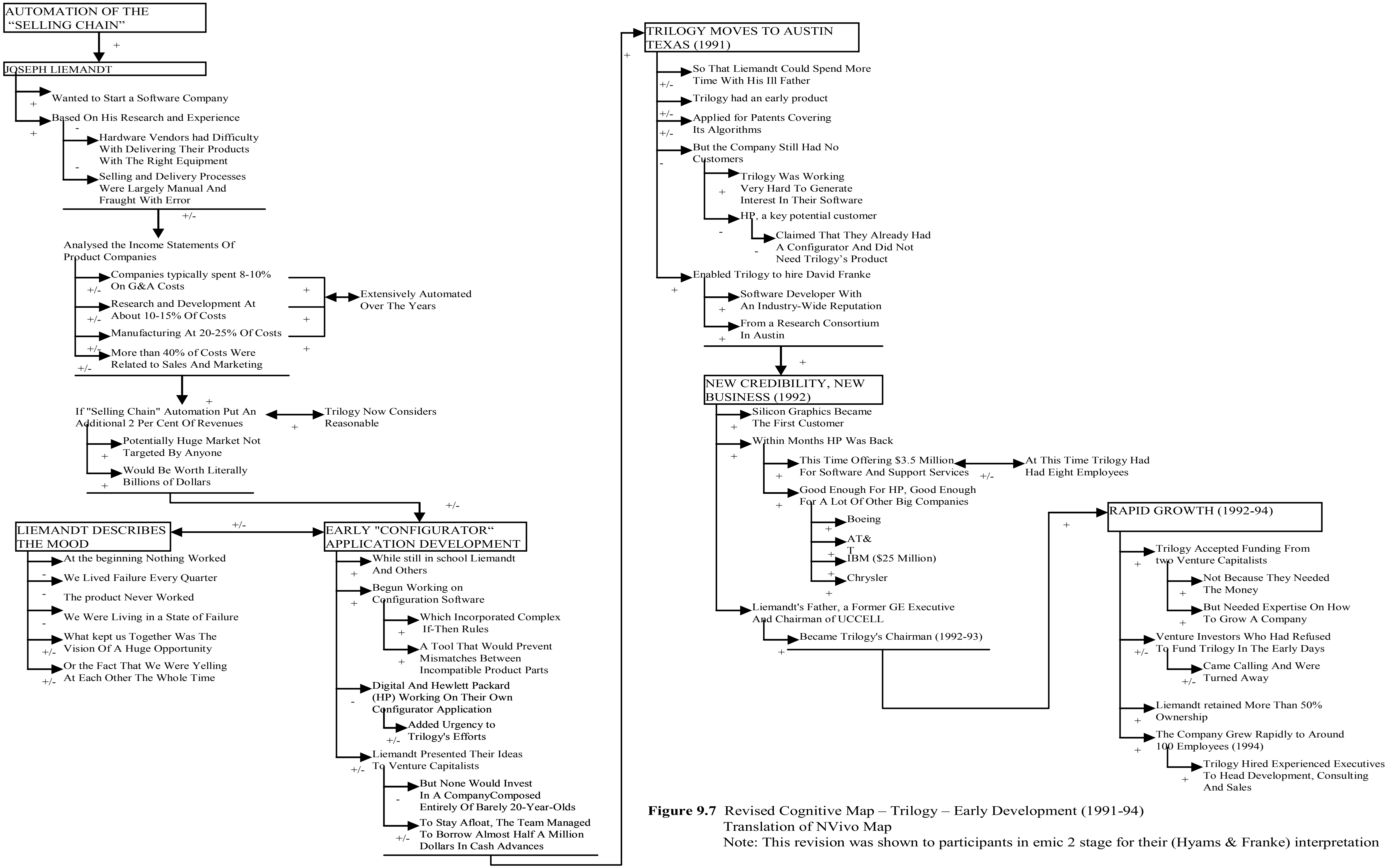


Figure 9.7 Revised Cognitive Map – Trilogy – Early Development (1991-94)
Translation of NVivo Map
Note: This revision was shown to participants in emic 2 stage for their (Hyams & Franke) interpretation

While studying at Stanford University, Joseph Liemandt wanted to start up a software company. He had unearthed an opportunity where computer hardware vendors had difficulty delivering products with the correct parts and equipment. Their selling and delivery processes were mainly manual, and were error-prone.

Liemandt conducted further analysis on the income statements of selected computer companies, finding a breakdown of costs into the following percentages:

- General and Administration (G&A) (8-10%)
- Research and Development (R&D) (10-15%)
- Manufacturing (20-25%)
- Sales and Marketing (40%)

G&A, R&D and Manufacturing had been extensively automated, but by 1990 there was very little automation within sales and marketing functions.

Liemandt calculated that attempting to automate aspects of the “Selling Chain” (the sales and marketing functions) to enable an additional 2% of revenues would be potentially worth billions of dollars – and was a new potential market not currently targeted by anyone. Austin (1998) indicated as an additional note that the estimate of adding 2% to revenues was a figure considered as reasonable by Trilogy.

Liemandt and some other students at Stanford started work on developing configuration software. Configuration software enables the user to incorporate all essential features and equipment for a product, plus nomination of optional additional products and accessories.

In the late 1980s, most quotes for computer hardware were line-by-line items manually entered without any form of validation or checking. For complex large system quotes, such as for multimillion dollar tenders, hardware consultants were employed to check system configurations and line-by-line. The author of this dissertation was a specialist consultant in the area of the configuration and pricing of complex computer systems in the late 1980s.

Early configuration systems were developed by DEC and HP in the mid-to-late 1980s, but they both required mainframe computer power to run through complex if-then rules, and even then only standardised systems with few additional options could be processed into quotes in a reasonable turn-around time.

Nevertheless, as minicomputer and small server performance was rapidly increasing, these configuration systems had the potential to be placed on smaller very powerful UNIX-based systems emerging at that time, and eventually on Windows-based PCs.

Liemandt realised that Trilogy needed to bring its configuration software to market quickly to counter DEC and Hewlett-Packard's early lead. Decision-making at this stage was all about evaluating options to commercialise and bring to an application rapidly to market.

Trilogy needed capital to accelerate application development. Liemandt sought venture capital support but no venture capitalist was prepared to invest in a company with such a young group of developers.

However, Trilogy's founders had such a strong faith in its capability and vision for its application that they leveraged its own credit cards with cash advances of almost half a million dollars. Such an attitude to risk-taking, that is, risk enough money to hurt, but not to break the bank is a fundamental organisational value at Trilogy and is explored in more detail in other sections of Austin's (1998) case study.

Liemandt described the atmosphere of Trilogy at the time (1990-91) as a state of failure where nothing worked, but a strong vision kept the team together. Or the team seemed to stay together by "yelling at each other the whole time".

A "circuit-breaker" for Trilogy seems to be Liemandt's decision to move Trilogy to Austin, Texas, in 1991. Liemandt wanted to spend more time with his ill father in Austin and moved there prior to his graduation from Stanford. Other members of the Trilogy gradually moved to Austin, some before graduation, some after graduation.

By the time the move to Austin was completed, Trilogy had developed an early application. In an unusual move for software developers, Trilogy applied early for patents – not so much for the actual application but for the algorithms used in the if-then rules within the application.

Although Trilogy was trying hard to generate interest in its application, they had no customers. In a twist suggesting Trilogy knew that its application was superior to Hewlett-Packard's existing configuration software, Trilogy was actively targeting Hewlett-Packard as a potential customer. At the time, Hewlett-Packard rejected Trilogy's overtures, claiming that it already had its own configuration software.

Moving Trilogy to Austin enabled Liemandt to hire David Franke from a research consortium in Austin in 1992. Franke was a well-known software developer with strong links to various hardware and software suppliers and he gave Trilogy a substantial boost in industry credibility.

Trilogy's increased industry credibility through Franke translated into new customers. Silicon Graphics (SGI) became Trilogy's first customer, and within a few months Hewlett-Packard signed up in a $3.5 million deal for software and support services. Trilogy was still a small start-up company with eight employees, facing exponential demand for its application.

Securing Hewlett-Packard as a key customer appeared to legitimise Trilogy in the eyes of several large companies who soon also signed up with Trilogy – and for large contracts. Boeing, AT&T and Chrysler rapidly signed up, while IBM took on Trilogy's software and support for over $25 million.

Liemandt's father Greg, although diagnosed with a fatal illness, was appointed Chairman of Trilogy during 1992-93, sharing valuable experience and knowledge from senior executive roles at GE and being Chairman at UCCELL. Trilogy now needed knowledge and resources to turn into a large enterprise servicing large corporate customers.

Trilogy was seen in this period as a great investment opportunity by several venture capitalists, but most were turned away as Liemandt really wanted expertise and resources to rapidly grow the company, and not just capital. Liemandt accepted funding from two venture capitalists that provided these additional benefits, but maintained 50% ownership of the company.

Trilogy grew rapidly through 1992-94 to around 100 employees and experienced executives were hired to establish organisational structure for development, consulting and sales. Trilogy was growing strongly with a "killer application" for sales & marketing – and that was all before the Internet revolution arrived.

9.9.1.1 Updated Cognitive Map 1 – Early Development (1991-94)

The cognitive map was revised through direct editing in PowerPoint with inputs from interviews with Franke (2002) and Hyams (2002). Several sections of the original map were validated with minimal changes, but there were some significant additional insights for other sections. The updated map is presented in Figure 9.9.

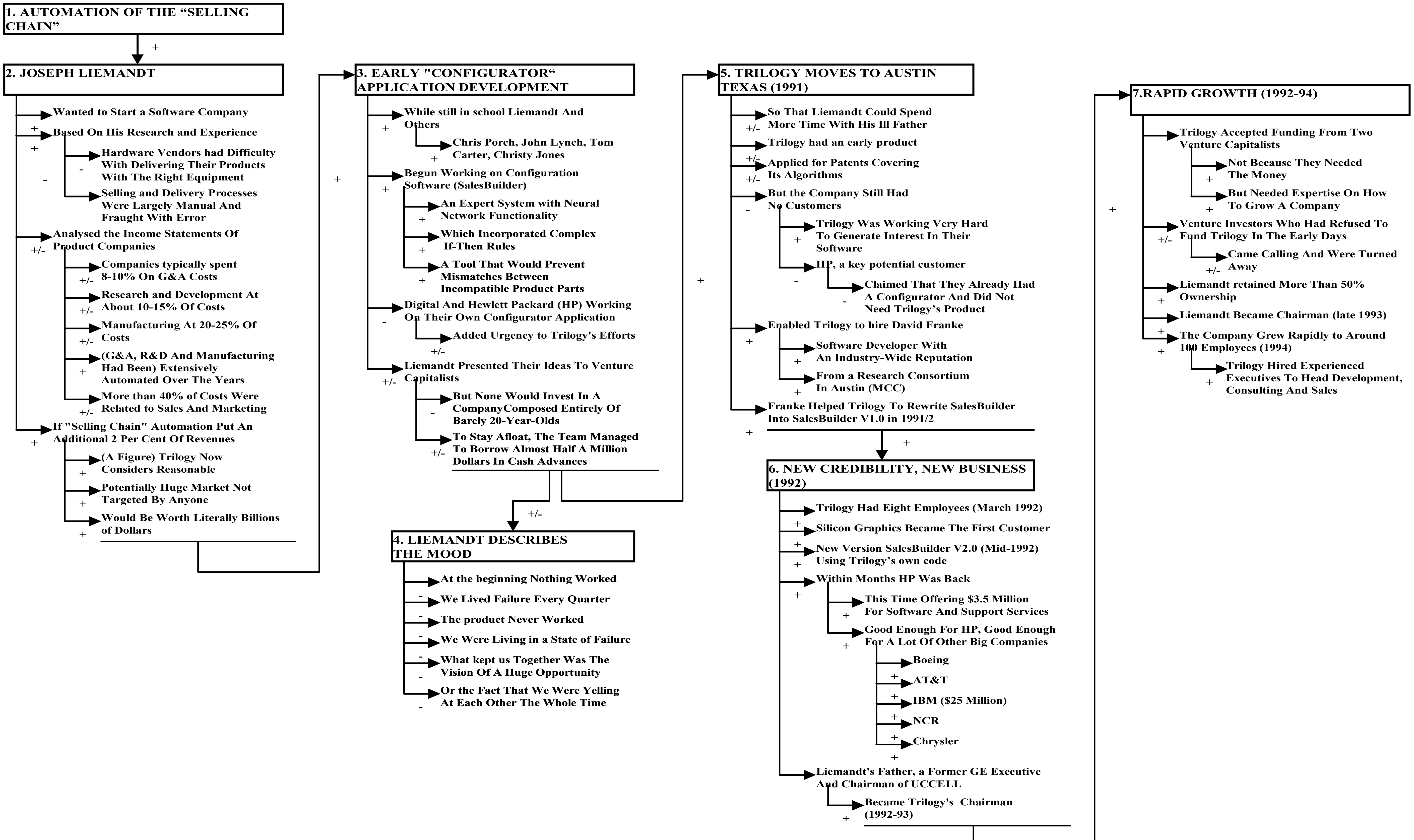


Figure 9.8 Second Revised Cognitive Map – Trilogy – Early Development (1991-94)
Note: This map represents etic 3 interpretation

Sections 1 and 2, apart from some formatting adjustments, were validated unchanged from the original cognitive map.

Section 3, on "early configurator application development", was updated to indicate that apart from Liemandt, the other early developers were Chris Porch, John Lynch, Tom Carter and Christy Jones.

The actual configuration application was known as SalesBuilder and was based on an expert system with neural network capability. These features enabled the setting up of complex rules for selection of features and options.

Section 4 was validated unchanged from the original map. Trilogy's move to Austin covered in Section 5, was amplified to provide more insight into Franke's move to Trilogy. Franke had worked with MCC in Austin, a company that was focussed on developing new computing technologies and software.

Franke had substantial knowledge in software development, expert systems and he was already quite familiar with algorithms and developments related to configuration applications.

Franke helped Trilogy formalise SalesBuilder into a formal release SalesBuilder V1.0.

Section 6, on New Credibility and New Business, was updated to incorporate additional insights. Trilogy had eight employees in March 1992. Trilogy developed SalesBuilder V2.0 in mid-1992, using its own code and developers. NCR was identified an additional large new customer in 1992.

Section 7, covering Rapid Growth, was validated unchanged except that Joseph Liemandt became chairman in late 1993 (after his father Greg).

9.9.2 Cognitive Map 2 – Industry and Competition

Development details for the second cognitive map include:

- Extract section, size and exhibit:
    - "Industry And Competition" section, Austin (1998, pp. 4-5); 1 page; Exhibit 9.8
- TACT word frequency lists and comments on key words:
    - First pass: 24 words with frequency > 2
    - Second & final pass: 13 words with frequency > 3
    - Final ranked word list: Table 9.2

  - The highly ranked words covered a diversity of themes, although *SAP* as an *ERP company* or competitor to *Trilogy* can be seen as a simple pattern from the list (words in italics were ranked highly in the ranking list).
- NVivo ranked word, sentence and phrase node count = 204
- Association of concepts coding used: +, -. +/-, e
- NVivo PowerPoint cognitive map: Figure 9.9
- Reformatted PowerPoint cognitive map: Figure 9.10

The cognitive map presented in Figure 9.10 provides a cognitive description of Trilogy's perception of its competition and industry space from the early 1990s through to 1999.

Exhibit 9.8 Text Extract – Trilogy – Industry and Competition

Industry and Competition

Liemandt's original analysis of spending patterns versus degree of automation in computer product firms had identified a wide-open market worth at least $10 billion. The few companies that were in that market at the time were bit players, selling things like contact management software for salespeople. Most of the functionality that constitutes the bulk of the "selling chain"-catalogue updating, configuration, pricing, bid preparation, commission calculation-was performed manually or by software written by product firms themselves. Trilogy had pushed rapidly and successfully into this mostly empty space.

But Trilogy's success did not go unnoticed. Beginning in about 1993, new companies like Aurum, Brightware, Calico, Clarify, Remedy, Scopus, Siebel Systems, and Vantive entered the general area of sales and marketing automation. Some of these companies targeted niches that were not in immediate competition with Trilogy. But all were operating in the same general space, going after that 40-plus per cent of P&L spending that Liemandt had first noticed as a student. More worrying than these small players, however, was the awakening interest of the giants of Enterprise Resource Planning (ERP)-rapidly growing companies that were many times larger than Trilogy, such as SAP, Oracle, Peoplesoft, and Baan (see Exhibit 3 for profiles of these companies – *note that this exhibit is not in this report but may be found in Austin (1998)*.

Exhibit 9.8 Text Extract - Trilogy – Industry and Competition (Cont'd)

Trilogy had intentionally positioned itself as an "enterprise software" company, meaning that its products were designed to work together to provide end-to-end functionality for a major segment of a customer's business (the selling chain). This was necessary because Trilogy's corporate customers were increasingly looking to build or buy integrated systems. Companies that did not sell enterprise products risked losing out to companies with more integrated and broader product offerings. But ERP vendors saw the enterprise software market as their turf. One company's supply chain, reasoned the ERP giants, was another company's selling chain. As the experts on integrating a customer's "back office"-the value stream from procurement through production to delivery-it seemed only natural to the ERP vendors that they should also integrate the "front office"-the selling chain. Liemandt summarised the threat to his company in stark terms:

They decided that Trilogy had done some fantastic research for them and that they'd just come in and take it over. The question was (and still is), "can we withstand the onslaught of giants ten times as big who want to move into our space?"

As early as 1993, Trilogy had realized that the number one threat to its long-term well-being was SAP, the largest of the ERP vendors, which by 1998 owned 70 per cent of the back office automation business for Fortune 500 customers. In 1997 and 1998, the threat from SAP and the other ERP vendors became more immediate. Baan purchased Aurum. Peoplesoft announced partnerships with Vantive and Siebel Systems (which itself bought Scopus). SAP pointedly failed to invite Trilogy to exhibit at Sapphire 1998, the SAP-sponsored tradeshow for its own customers, even though the company had

participated in earlier years.4 At that same tradeshow, Hasso Plattner, SAP's chairman and cofounder, announced to his customers that 80% of the company's R&D going forward would be aimed at building front office products.

Trilogy (A) 699-034

Trilogy had a considerable head start on ERP vendors in the development of key technologies, especially configuration software, some of which was by then protected by patents. But the protection provided by patents would be short-lived. Whether Trilogy would remain a factor would depend far less on past accomplishments than on what they could accomplish in the future.

Source: Austin 1998, pp. 4-5

Table 9.2 Trilogy – Industry And Competition – Ranked Word List

Anchor Concepts	Frequency
company	13
trilogy	12
erp	7
customer	6
sap	6
sell	6
chain	5
product	5
software	5
vendors	5
enterprise	4
integrate	4
office	4

Source: data analysis performed using TACT (1997)

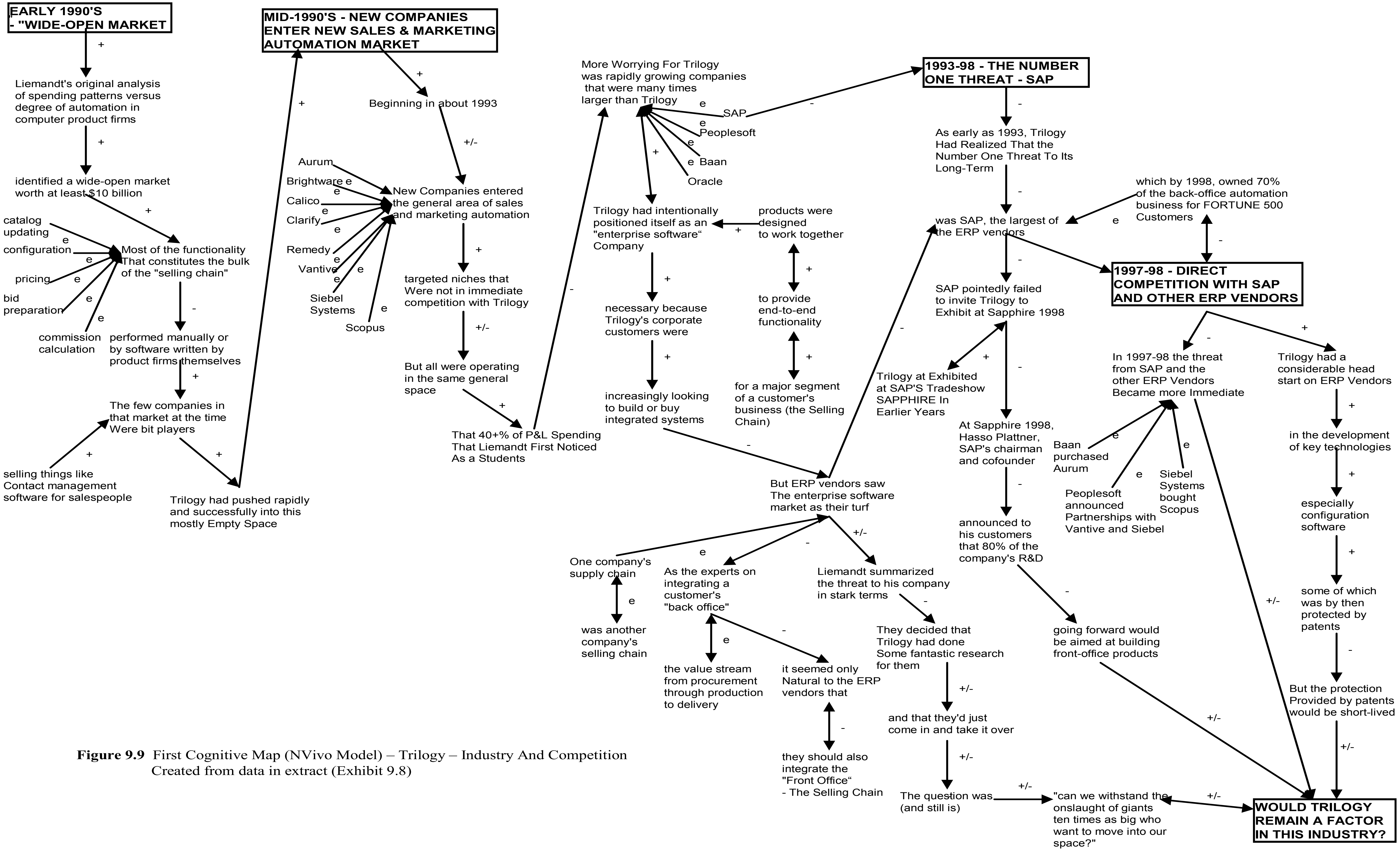


Figure 9.9 First Cognitive Map (NVivo Model) – Trilogy – Industry And Competition
Created from data in extract (Exhibit 9.8)

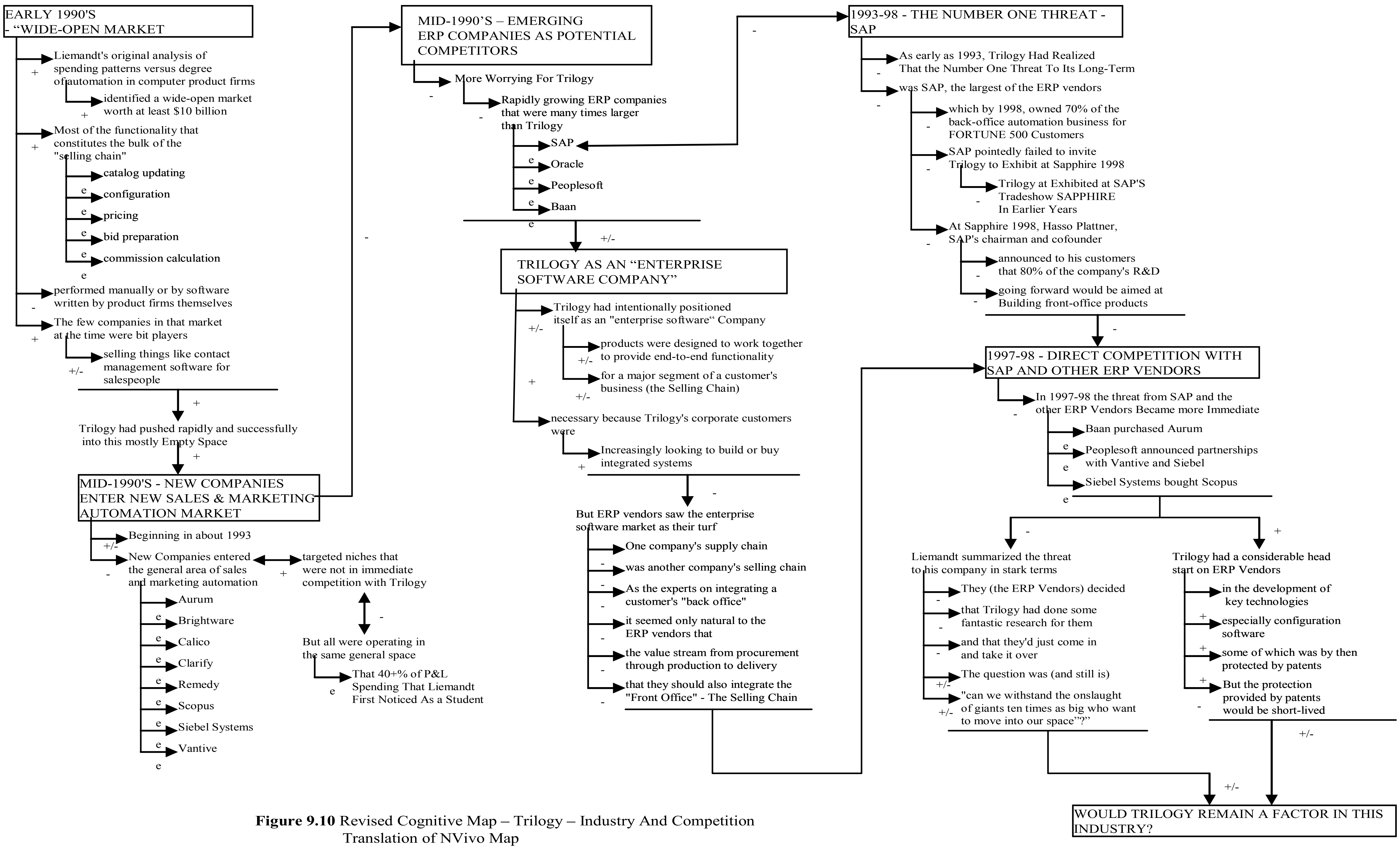


Figure 9.10 Revised Cognitive Map – Trilogy – Industry And Competition
Translation of NVivo Map
Note: This revision was shown to participants in emic 2 stage for their (Hyams & Franke) interpretation

Liemandt identified an unmet market opportunity (see also first cognitive map) that he estimated to be at least $10 billion in the area of the sales & marketing functions, or the "front-end" or "front-office" or the "selling chain".

Most activity in the selling chain activities for computer hardware companies such as catalogue updating, configuration, pricing, bid preparation and sales/consulting commission calculations were processed manually.

There were a few companies that sold software for specific activities in the selling chain such as contact management software. There were some forms of simple customer database applications available for direct marketing programs but CRM as a concept emerged in the late 1990s.

Trilogy was one of the first companies with a solution that went further than just one component of the "Selling Chain". However, it was not long before Trilogy faced competition from several fronts.

Several of the companies that we would now see as CRM vendors were created in the early 1990s. Eight companies are noted in the original cognitive map, of which probably Siebel Systems is best known for sales force management applications.

Most of the companies noted were not in immediate competition with Trilogy – but they all could be classified as competition, depending on how an industry space for a "selling chain" was defined.

Trilogy faced more aggressive and powerful competition from a different front. Trilogy's executive always saw the company as an "enterprise software" company that could address a full end-to-end "selling chain". Trilogy was in part responding to customers who wanted to build or buy systems that at least integrated the selling chain.

Such positioning put Trilogy directly in the path of large and rapidly growing enterprise resource planning (ERP) vendors. Companies such as SAP and Oracle had been established at least 10 years longer than Trilogy and had grown rapidly through the 1980s and 1990s offering various key enterprise applications.

SAP had grown out of offering enterprise financial software, expanding into manufacturing and logistics in the early 1990s. SAP could be put on several computer hardware platforms but was often used on IBM systems through the 1980s and early 1990s.

Oracle rode the Open Systems/UNIX boom of the late 1980s and early 1990s to be the major database application for various enterprise applications to run on UNIX hardware vendors including Sun, Hewlett-Packard, and DEC's UNIX systems.

Oracle was also early to position its applications as e-business solutions to work with the World-Wide Web (working with such applications as NetDynamics – see the NetDynamics case study).

Peoplesoft was created in the late 1980s with Human Resource Management (HRM) applications but spread quickly into other enterprise applications, working closely with UNIX computing vendors, but also with IBM and DEC's VMS systems of the early 1990s.

Baan was a Dutch company that emerged through enterprise financial applications in the early 1990s.

The ERP companies' main focus was integration of "back-office" activities such as manufacturing, logistics, and financial management. However, it was logical for the ERP companies to expand its application suites right from procurement to delivery, including integration of front-office activities.

Trilogy identified SAP as its number one long-term competitor as early as 1993. SAP dominated the "back-office" ERP business with about 70% share of that business by 1999.

However, for about five years Trilogy worked with SAP as a key ISV and regularly exhibited at Trilogy's main tradeshow, SAPPHIRE.

SAP changed its strategy in 1998 to focus heavily in building front-office applications. One side-effect of that strategy was not to invite Trilogy to SAPPHIRE 1999. At that event SAP's Chairman committed that 80% of SAP's R&D would go to development of front-office applications.

By 1997, competition between the main ERP vendors and Trilogy was quite direct, as a number of ERP companies either acquired ISVs with front-office applications, or they attempted to develop its own applications.

Trilogy's management believed that the company possessed a considerable lead over the ERP vendors in the selling-chain area. Trilogy was very strong in configuration technology, for which it held some significant patents.

However, Trilogy needed to continually develop new front-office-related applications and grow fast enough to gain sufficient critical mass to take on much larger companies pushing into the selling-chain space. Liemandt directly questioned whether

Trilogy could survive as a key player at least in the selling chain industry that he had defined back in the early 1990s.

Liemandt was reflecting on a not uncommon problem for small start-up companies with strong new technologies facing up to larger more established companies attempting to break into its area of expertise.

Other case-studies in this project and particularly those with enterprise software solutions have faced the same problem as Trilogy's.

NetDynamics gained a strong early lead with its application to link WebPages with databases, but was then acquired by Sun, as it needed size and resources to support a rapidly growing enterprise customer base.

Kana rapidly acquired ISVs to develop a full e-CRM solution. It moved from a focused application on e-mail communication management to a redefined and more integrated e-CRM and e-commerce applications space.

Trilogy could see that they needed to at least strengthen its product offering for the selling chain, although at the same time, it was not committing to redefine its offerings to match the total ERP solutions.

The overall cognitive map presents Trilogy's management perceptions of competitive evolution through the 1990s. The map also highlights Trilogy's executive team's strategic mapping of its competitive position, initially in a new industry, but then in a more broad and redefined industry with much larger and more aggressive competitors on the same perceived turf.

9.9.2.1 Updated Cognitive Map 2 – Industry and Competition

The cognitive map was revised subsequent to interviews with Franke (2002) and Hyams (2002), and is presented in Figure 9.11. Most of the original cognitive map was validated unchanged, but there were some additional insights and structural updates incorporated into the map.

The first section of the map covering "Early 1990s – Wide-open Market" (box 1) was validated with only two modifications. The first indicated that Trilogy released an application Selling Chain V1.0 in 1994-95 with the functionality described in the original map. The second modification was the incorporation of the comment in the original map relating to Trilogy's moving into a mostly empty space, into the first section of the updated map.

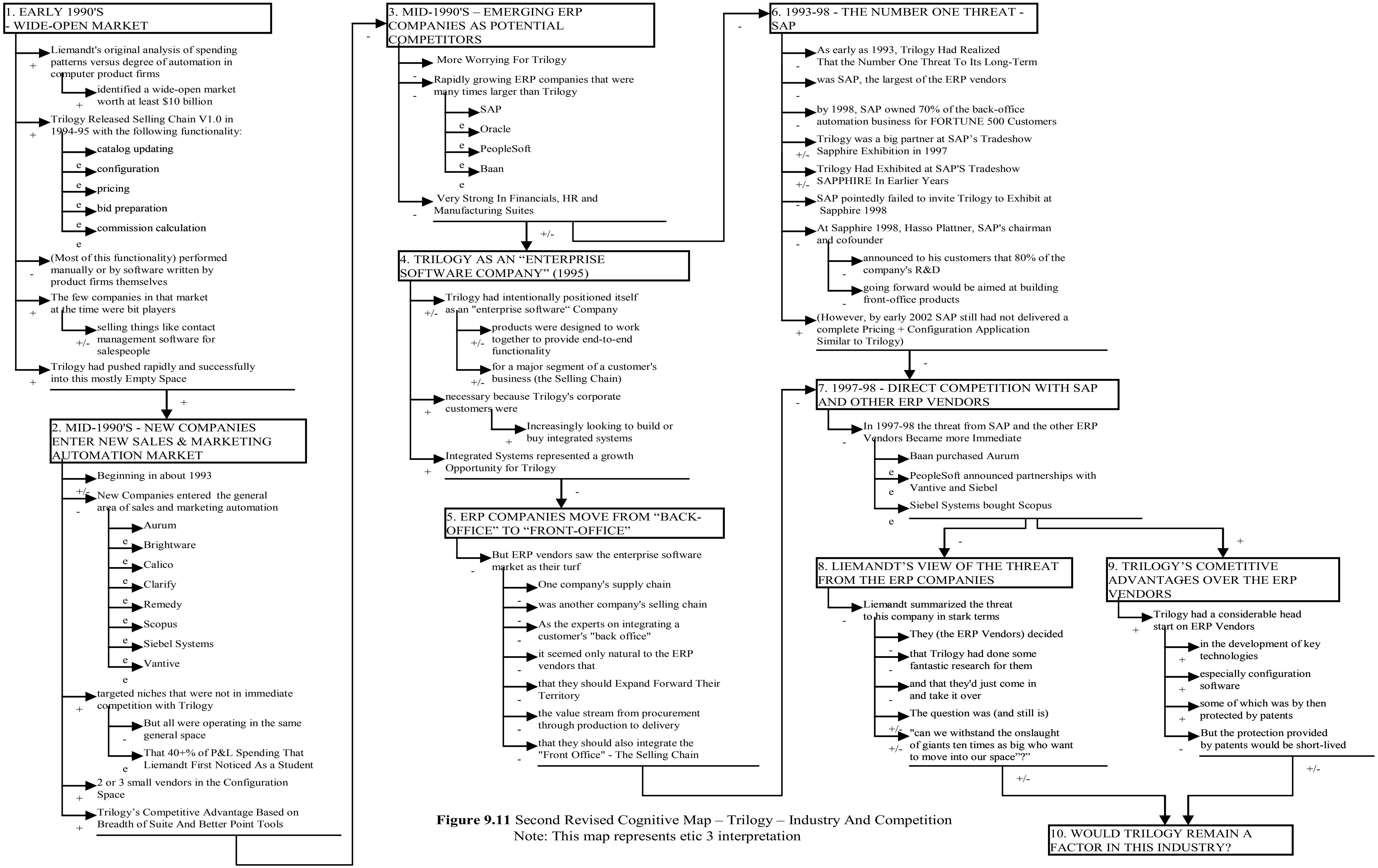


Figure 9.11 Second Revised Cognitive Map – Trilogy – Industry And Competition
Note: This map represents etic 3 interpretation

The second section, “Mid-1990s – New Companies Enter New Sales & Marketing Automation Market”, was validated with two modifications related to the same point. Franke (2002) noted that there were two or three vendors in the configuration space. Trilogy’s competitive advantage was based on the breadth of its new application suite (Selling Chain V1.0) and better point-by-point configuration capability.

The third section on emerging ERP companies as potential competitors was validated unmodified, but extended to highlight the ERP vendors’ strengths in Financials, Human Resources Management and Manufacturing suites.

Trilogy’s positioning as an “Enterprise Software Company” (box 4) was validated without modifications to content, except that the sub-section on ERP vendors seeing enterprise software as its turf was reformatted into a separate section (box 5 – ERP Companies Move From “Back-Office” to “front-office”).

The original section on SAP being the Number One threat to Trilogy was slightly reformatted (See box 6) but content was validated without modifications. An additional point was incorporated into the section indicating that, even by early 2002, SAP had still not delivered a complete Pricing and Configuration application similar to Trilogy.

The section on the consolidation of ERP Vendors with other “front-end” vendors through selected acquisitions was validated unchanged for content. However, this section, including the subsections on Liemandt’s view on the threat of from the ERP companies, and Trilogy’s competitive advantages over them, were split into three sections. (See box 7 for Direct Competition, box 8 for Liemandt’s view of the ERP companies, and box 9 for Trilogy’s Competitive Advantages over the ERP Vendors).

The question at the end of the original map regarding whether or not Trilogy would remain a factor in the industry was reformatted as box 10.

The revised cognitive map largely validates the original map, with some extra insights specific to Trilogy’s actual Selling Chain application suite.

9.9.3 Cognitive Map 3 – Software Development

Development details for the third cognitive map include:

- Extract section, size and exhibit:
    - “Developing The Product” section, Austin (1998, pp. 7-8); 1¼ page; Exhibit 9.9
- TACT word frequency lists and comments on key words:
    - First pass: 29 words with frequency > 2
    - Second & final pass: 19 words with frequency > 3

  - Final ranked word list: Table 9.3
  - The ranked word list highlights themes relating to *Product Development* at *Trilogy* and the role of *developer(s)* in creating *features* for Trilogy's products (italicised words were ranked highly in the ranked word list).
- NVivo ranked word, sentence and phrase node count = 135
- Association of concepts coding used: +, -. +/-
- NVivo PowerPoint cognitive map: Figure 9.12
- Reformatted PowerPoint cognitive map: Figure 9.13

The map provides further insights into Trilogy's product development capability and its fundamental philosophies of software development.

Exhibit 9.9 Text Extract – Trilogy – Software Development

Developing the Product

Trilogy's marketing goals depended vitally on the company's product development capability. Specifically, marketing objectives required that developers sustain the competitive advantage Trilogy enjoyed in configurator technology while dramatically broadening the product to fill the enterprise needs of large and small customers in a variety of industries.

Trilogy aspired to maintain a software development capability that was second to none. Their comparison set for evaluating themselves in this area was not their direct competitors, but other world class development organisations, especially Microsoft. In their aim to be the best, they believed that they had largely succeeded. Liemandt was convinced that no other enterprise software vendor was even a close second to Trilogy in development capability. Scott Snyder, Trilogy's Senior Development VP, estimated that 15 or 20 successful software companies could be built around the talent in Trilogy's development organisation.

Central to the company's development capability was the "rule of the super coders," which held that one superstar programmer could do the work of ten average programmers. "Getting the most out of great developers," remarked Snyder, "is one of the things Trilogy does amazingly well." The development process was geared toward giving Trilogy's superstar programmers the support and freedom they needed to produce great products. Snyder described some of the company's fundamental philosophies of software development:

Exhibit 9.9 Text Extract – Trilogy – Software Development (Cont’d)

Our development is based around four basic philosophies. Small teams, very small from a traditional development standpoint. We expect entire new products to be created by one or two superstar programmers you can count on to deliver great products quickly. Complete ownership of the product at the developer level from initial product requirements gathering through product support. We don't have a separate change team that insulates the developers from the impact of producing poor quality products. Intense focus on automation in order to free the developers (or anyone else) from having to spend their time manually performing frequently repeated tasks like regression tests. Finally, a focus on incremental development model that allows us to deliver new functionality quickly and provides us the flexibility to react to changes in the market quickly.

A key feature of the development process was that it evolved to maintain responsiveness to the market, becoming more structured as the product matured. As the product grew beyond a certain size, explained Snyder, maintaining responsiveness and high product quality depended on some key disciplines:

The goal is to maintain your code at ship level quality on a weekly basis. When a developer drops code for a new feature or bug fix into the build, it must be accompanied by the appropriate suite of automated tests to validate that the changes work as expected. These tests are added to the existing suite and the entire set is executed every time the product is built, whether that was a weekly or nightly build.

If you had to develop a feature that took longer than the weekly build cycle, you branch your development, develop the feature, develop tests, merge it all back into the main build, then rerun all the tests on the integrated code. In addition to the individual product tests, we also have automated system tests which test the interactions between products. The goal is to constantly improve the quality of the product as you increase its functionality. Again, a very incremental model. It's awesome and brilliantly suited for an environment which requires you to react quickly to any new requirement or change in market direction as long as you maintain the quality discipline.

As the product grew in complexity, programmers retained absolute freedom to add features in whatever way they saw fit, but they were obliged to maintain quality. As interdependencies developed between different developers' programs, the build and release cycle became more structured, with decisions being made about the timing of the release of new features on a feature-by-feature basis. New features were scheduled around a plan that used 60 to 70 per cent of Trilogy's development capacity with the remaining 30 to 40 per cent held in reserve for late breaking and urgent fixes.

Source: Austin 1998, pp. 7-8.

Table 9.3 Trilogy – Software Development – Ranked Word List

Anchor Concepts	Frequency
product	18
development	14
trilogy	8
feature	8
test	8
developer	7
new	6
maintain	6
quality	6
build	5
market	5
programmer	5
capability	4
quickly	4
Snyder	4
change	4
coder	4
develop	4
software	4

Source: data analysis performed using TACT (1997)

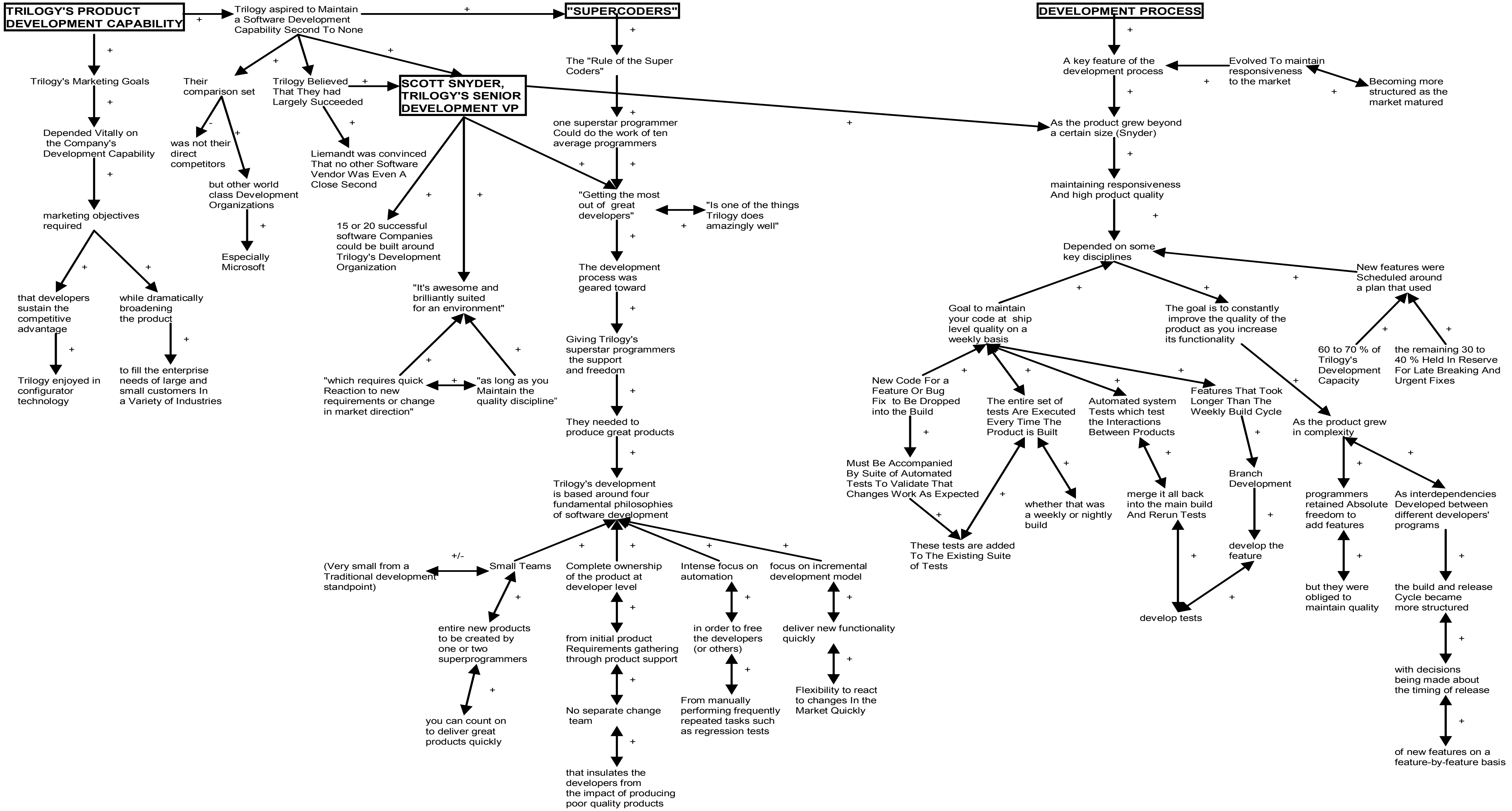


Figure 9.12 First Cognitive Map (NVivo Model) – Trilogy – Software Development
Created from data in extract (Extract 9.9)

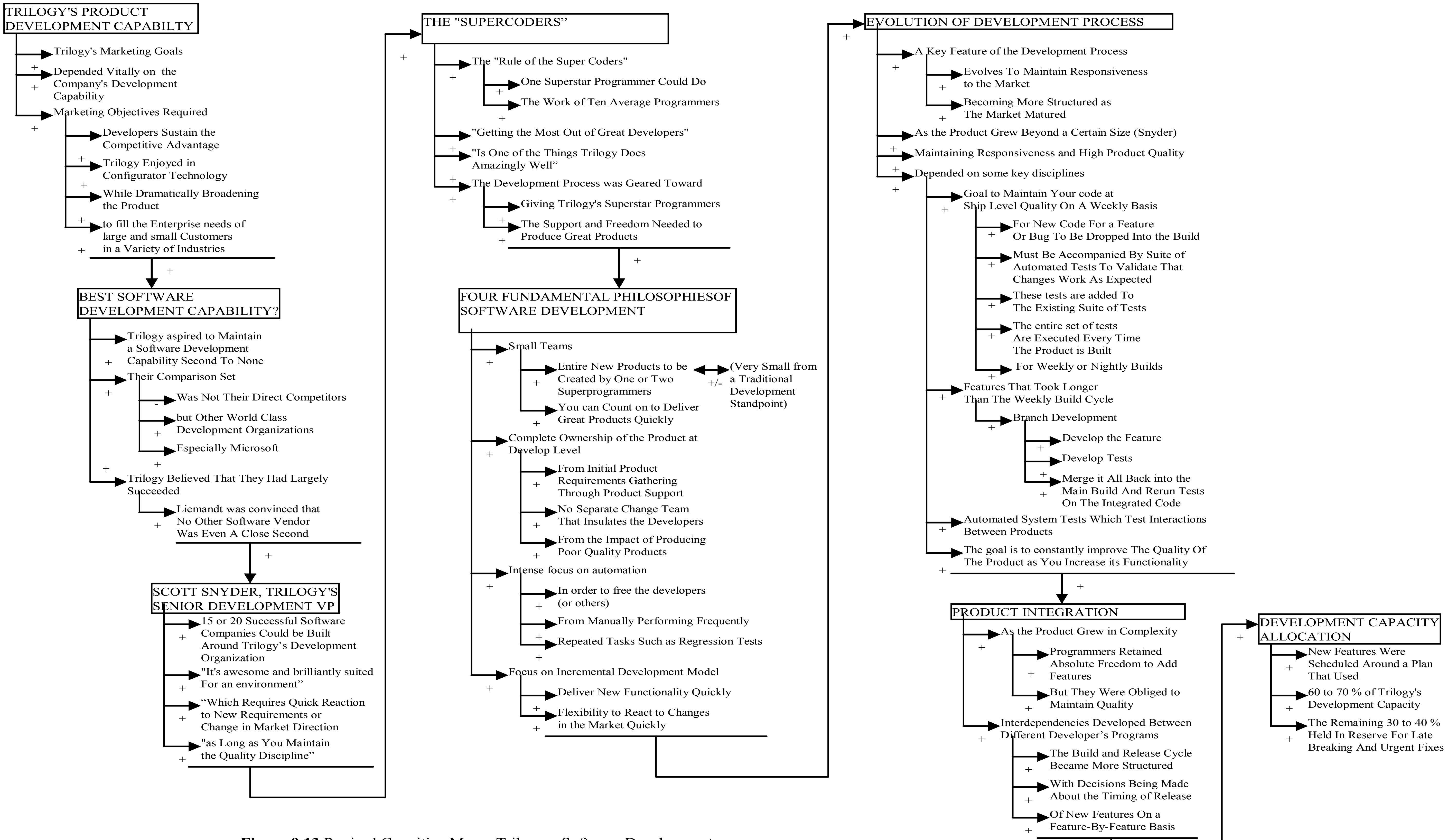


Figure 9.13 Revised Cognitive Map – Trilogy – Software Development
Translation of NVivo Map
Note: This revision was shown to participants in emic 2 stage for their (Hyams & Franke) interpretation

According to Austin (1998), Trilogy's marketing objectives were focused heavily on the company's development capability. Trilogy's developers had the goal of sustaining its competitive advantage in configuration software, but also to quickly broaden the product to meet the enterprise needs of a wide range of customers.

Balancing leadership in their core competency with broadening the appeal of the product to more customers was a logical approach to the rapidly growing enterprise software market during the mid-1990s. Such an approach also required the rapid recruitment of large numbers of very highly skilled application programmers.

Trilogy was created as a company with a deep belief in aspiring to be the best software development company in the world. Trilogy initially benchmarked its emerging development capability against other world-class development organisations, but particularly Microsoft. By 1998, Liemandt was convinced that Trilogy was the best by a long way in software development capability.

Trilogy's Senior Development Vice-President in 1998, Scott Snyder provided insights into Trilogy's view of itself as a best-in-class software development company. He regarded Trilogy's development organisation as "awesome and brilliantly suited for an environment which requires quick reaction to new requirements or change in market direction, as long as you maintain the Quality Discipline" (Austin 1998, p. 8). Snyder also believed that Trilogy's development organisation could be the basis for "15 or 20 successful software companies".

In order to grow and maintain a best-in-class software development capability, Trilogy encouraged a concept of "Super Coders", whereby one Superstar Programmer could do the work of ten average programmers. Trilogy focused on "getting the most out of great developers" through "giving them the support and freedom needed to produce great products".

A focus on maximising productivity from high-quality programmers, right through to linking its achievements through to marketing goals, is an unusual feature of Trilogy as an organisation.

Trilogy's software development was built around four fundamental philosophies which Austin has recounted from Scott Snyder.

Small teams, in fact teams of even one or two Superprogrammers developing entire applications (or at least modules of applications), follows logically from the very high value Trilogy placed on these individuals, both from a productivity and quality output perspective.

Complete ownership of the product at developer level is driven down to a unit of analysis of a team of one or two Superprogrammers. Trilogy gives its teams control of the full application development cycle – both for good and poor quality products. In reality the ownership is at a collective team level and this is evident where there are requirements to create new modules or new applications.

Intense focus on automation enabled Superprogrammers to be freed up from manually performing frequent repetitive tasks and to focus on direct application development.

Focus on incremental development model seems initially to be a contradiction in that Superprogrammers might be expected to quickly create radical new applications and products. However, Trilogy seems to have placed a higher priority on sufficient modularisation and break-down of development components to enable the Superprogrammers to quickly change components in response to fast changes in markets.

Applying the four fundamental philosophies of software development to actual development processes required balancing the need to maintain responsiveness to markets with becoming more structured as markets matured. According to Tom Snyder, maintaining this balance depended on some programming and development disciplines.

Development teams should have a goal to maintain its actual code at a level where it could be shipped (or be certified as completed) on a weekly basis. A new component or feature or bug that was to be incorporated into the weekly build of the overall application had to include a set of automated tests that would validate that the addition would work as expected – outside and within the weekly build.

The tests would be executed every time a build was undertaken. While a typical build may be undertaken weekly, some applications were built into new versions nightly – with the tests having to run at the frequency of the build.

Development Teams had to quickly judge if a feature or component or bug was likely to take longer than one weekly build cycle. Anything longer than the weekly build cycle prompted a branching of development, probably to a new or revised small team, which would undertake the development with the automated tests and then ensure that it merged successfully back into the overall application, usually in the next weekly build.

Additional automated systems tests were created to test interactions between applications, modules and other Trilogy products. Trilogy focused on constantly increasing the quality of its overall product-line as it also increased functionality within the line.

Over time Trilogy's product grew in complexity and, while the Superprogrammers retained the freedom to add features, they had to maintain overall product quality. Interdependencies emerged across and between the development teams with more decision-making required across the teams.

Cross-team decision-makers contributed to increasingly structured build and release cycles, and all additional features were subjected to substantial assessment before being accepted for both development and incorporation in the product.

By the late 1990s, Trilogy's development processes had evolved to allocate 60 to 70% of Trilogy's development capacity for new features, and 20 to 30% in readiness for late breaking and urgent fixes.

The cognitive map captures insights from a company that puts the very highest priority on best-in-class software development. There is literally a unique definition – and acclaim – of a "Supercoder" or "Superprogrammer" that is capable of much higher application development productivity than typical programmers. Trilogy pushes this elite group of programmers further with philosophies and supporting processes designed to maximise its productivity.

Trilogy appears to have achieved scalability throughout the late 1990s, with its "Superprogrammer" approach, maintaining frequent releases and updates, with both high-quality shipment code and effective bug and quality fix processes.

9.9.3.1 Updated Cognitive Map 3 – Software Development

The cognitive map was validated almost unchanged through interviews with Franke (2002) and Hyams (2002), and is presented in Figure 9.14. The original cognitive map was validated unchanged except for the numbering of sections and for a comment from Franke (2002) that there were 20-30 small development teams during the late 1990s (this comment was added into the *Small Teams* subsection of "Four Fundamental Philosophies of Software Development").

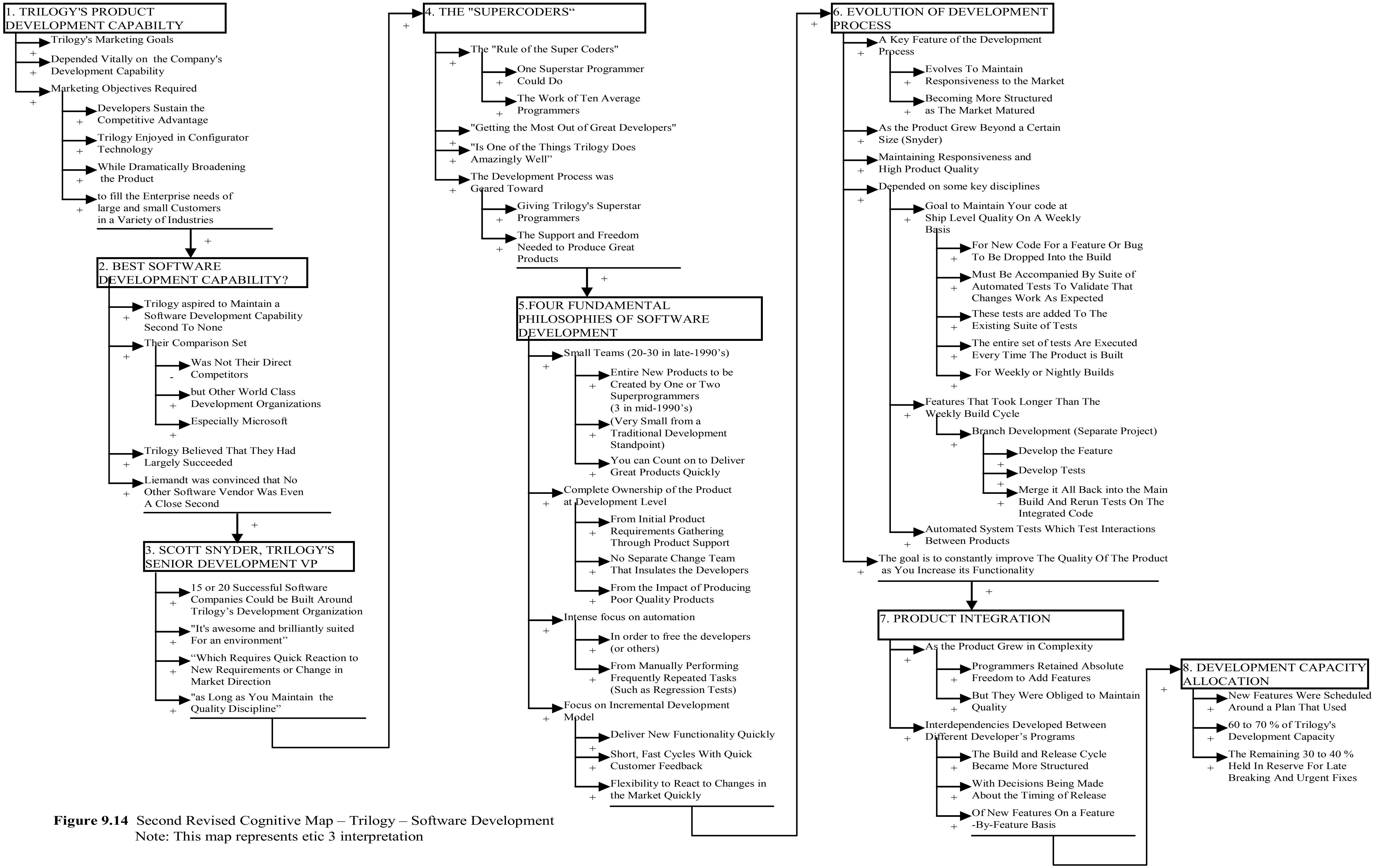


Figure 9.14 Second Revised Cognitive Map – Trilogy – Software Development
Note: This map represents etic 3 interpretation

Franke (2002) and Hyams (2002) agreed that Austin's (1998) account of Trilogy's software development was accurate and with substantial insights – and these had been effectively captured in the original cognitive map.

9.9.4 New Cognitive Map 4 – Industry-focused Application Development (2000-01)

During the interviews for the Trilogy project, suggestions were put forward that both a new DSA model and supporting cognitive map for Trilogy's application development in 2001 be developed for the case study.

Development details for the fourth cognitive map include:

- Extract section, size and exhibit:
    - Subset from "Trilogy Goes Vertical – Industry-focused Enterprise Solutions" Vignette; 1 page, Exhibit 9.10
- TACT word frequency lists and comments on key words:
    - First & final pass: 19 words with frequency > 2
    - Final ranked word list: Table 9.4
    - The ranked word list highlights themes relating to *product development* at *Trilogy* with an *industry* focus (italicised words mainly in singular form were ranked highly in the ranked word list).
- NVivo ranked word, sentence and phrase node count = 94
- Association of concepts coding used: +, -. +/-, e
- NVivo PowerPoint cognitive map: Figure 9.15
- Reformatted PowerPoint cognitive map: Figure 9.16

Exhibit 9.10 Extract From Vignette – Trilogy's Industry-Focused Application Development

TRILOGY GOES VERTICAL – INDUSTRY-FOCUSED ENTERPRISE SOLUTIONS

In 2000, Trilogy restructured from a horizontally focused application vendor toward using a Software Development Methodology designed to quickly conceptualise, create and deliver highly verticalised applications. Previously, from 1993-99, Trilogy's application development strategies were based on Trilogy Product Roadmaps.

The Executive Team established an Operations Group with five vertical divisions focused on key enterprise customers and products:

* Automotive
* Computer
* Telecommunications
* Financial Services
* New Business

Trilogy has a two-dimensional matrix of management and delivery resources, located in core company-wide Groups and within each Division with deep industry experience. These resources cover:

* General Managers plus Functional Vice-Presidents
* Development Staff (Division-specific)
* Consulting
* Human Resources
* Finance
* Solutions Marketing and Business Development
* Product Management and Presales Resources

Exhibit 9.10 Extract From Vignette – Trilogy's Industry-Focused Application Development (Cont'd)

Trilogy's Fast Cycle Time (FCT) Software Development Methodology is now based on delivering vertical applications within Industry Maps set up for each division for a two-year timeframe. There are four phases in this methodology:

1. Product Ideation

During this phase, Trilogy's Division Managers with their Solutions Marketing and Product Marketing teams generate ideas for products, create a vision for the product and the Industry, apply an investment justification process to the vision and product, and develops an Industry Roadmap supported by a Business Case. All product conceptualisation and development is driven with a vertical industry focus.

2. Product Planning

During this phase, the capabilities of the product will be identified and sequenced for delivery. Product features will then be expanded into actual development schedules including actual release features and initial estimates at release dates. During this phase, Product Management takes over with Development Teams in driving product planning.

3. Product Development

This phase consists of gathering the detailed requirements for the product, and commencing development using Trilogy's Fast Cycle Time methodology. Although Product Management and Development Teams are driving the product development process, chartered customers associated with specific Trilogy divisions may be involved in testing early versions of the product.

4. Product Delivery

Product Delivery is defined as beyond beta Release. There is extensive interaction between Product Management, Development Teams, chartered customers, and solutions marketing groups. The final product will be delivered during this phase. Once in production, the product will be maintained with patches, bug fixes and sub-releases. Consultants may work with chartered customers in the deployment of new products.

Source: Extracted from Exhibit 9.6.

Table 9.4 Trilogy – Industry-Focused Application Development (2000-01) – Ranked Word List

Anchor Concepts	Frequency
product	26
development	14
trilogy	10
industry	7
deliver	7
division	6
phase	6
management	5
team	5
application	5
vertical	5
release	4
marketing	4
solutions	4
customer	4
resource	4
during	4
methodology	4
business	3

Source: data analysis performed using TACT (1997)

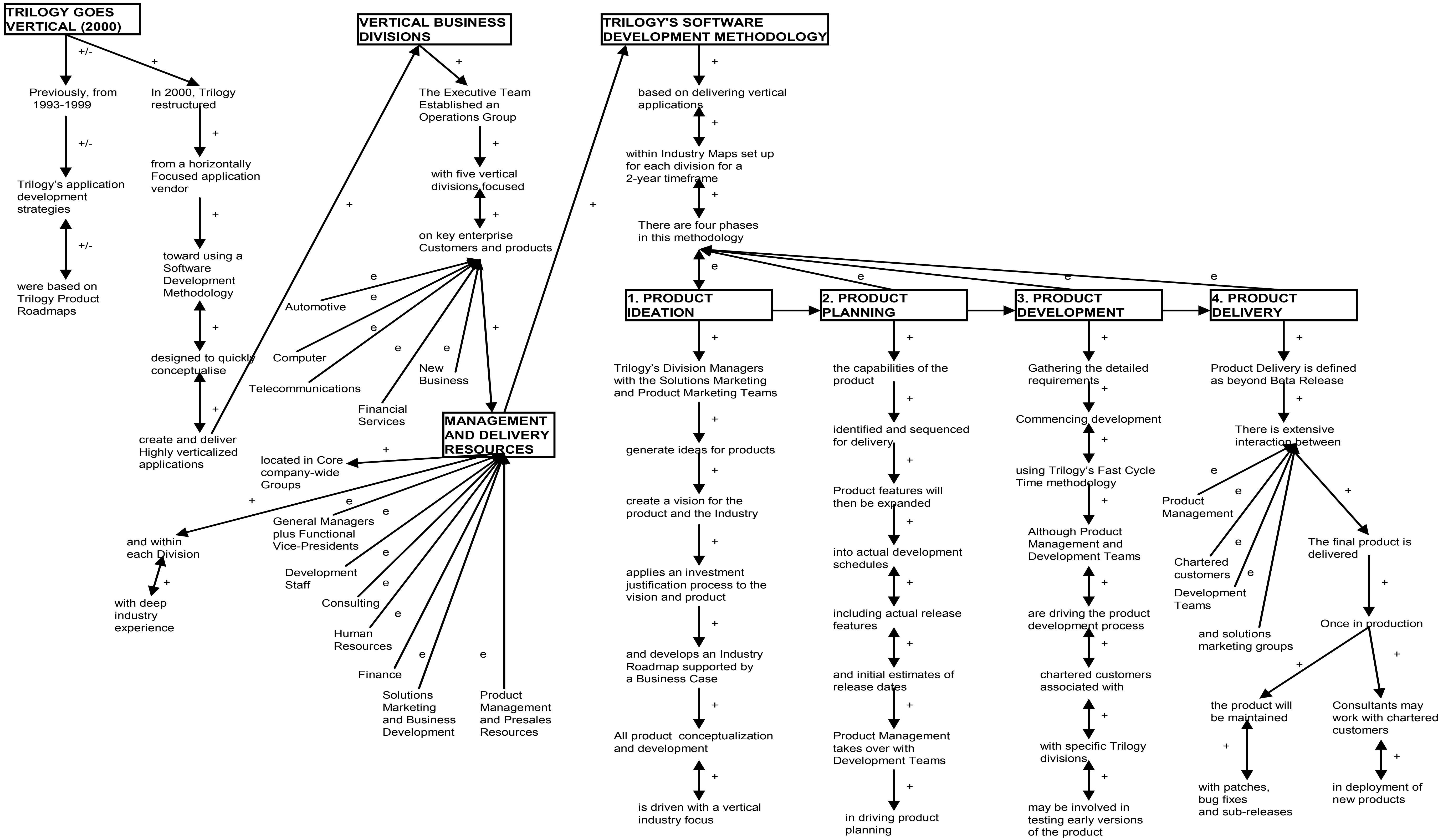


Figure 9.15 First Cognitive Map (NVivo Model) – Trilogy – Industry-Focused Application Development (2000-01)
Created from data in extract (Exhibit 9.10)

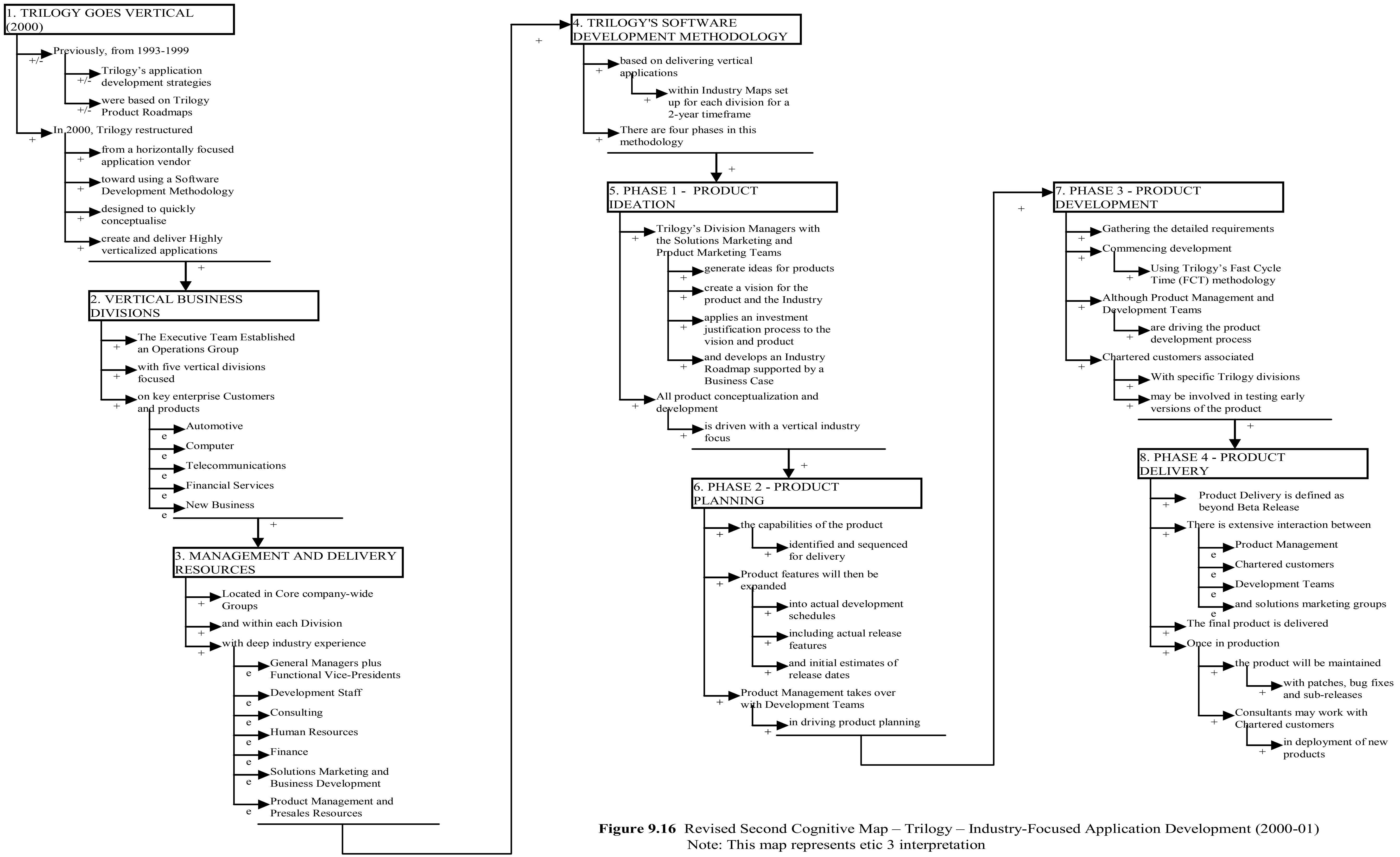


Figure 9.16 Revised Second Cognitive Map – Trilogy – Industry-Focused Application Development (2000-01)
Note: This map represents etic 3 interpretation

From 1993-99, Trilogy's focus was on excellence in software development, mainly in horizontal configuration applications. Trilogy had been servicing enterprise customers since its inception due to the nature of its configuration application and extensive potential for automation of key sales and marketing activities using the application and extended enterprise-oriented versions of it.

However, in 2000 during and after the "dot.com" crash and falls in technology/computing stocks, the company restructured toward a focus on selected vertical industry sectors (see box 1).

Trilogy revamped its software development methodology toward quickly conceptualising and road-mapping applications against targeted vertical markets, thereby developing and delivering highly tailored applications for specific industries and customers.

Trilogy's Executive Team established an Operations Group with five vertical divisions, based on its key enterprise customers and products (box 2). Trilogy's five divisions are Automotive, Computer, Telecommunications, Financial Services, and a New Business Division.

Each division was set up with a complete set of management and delivery resources (box 3). Each division possessed resources with deep vertical industry experience in all management areas, including General Management, Development, Consulting, Human Resources, Finance, Solutions Marketing and Business Development, plus Product Management.

In terms of decision-making, Trilogy's restructure was a radical departure from decision-making focused at the Superprogrammer or at the small development team level, toward managers with deep industry experience – but not necessarily focused on software development.

Trilogy's software methodology is now based on delivering vertical applications within each Division (box 4). Each Division has its own industry maps set up for a two-year timeframe which is designed around a four-phase development methodology.

The first phase of software development methodology is *Product Ideation* where Division managers, with solutions marketing and product marketing teams, work together on generating ideas and vision, which in turn are translated into products within the context of targeted industries.

Although there appears to be substantial freedom to visualise and conceptualise new applications in this early phase development, there is an immediate requirement to apply an investment justification process and to immediately map them to an Industry Roadmap with a Business Case.

All product conceptualisation and development is undertaken with direct alignment to specific divisional industry roadmaps, and thus is driven right from the outset with a vertical industry focus.

In the second phase of development, *Product Planning*, Product Management, with Development Teams (within each Division), drive the definition of product capabilities and scheduling of development and delivery (box 6). Product features are identified and defined into full development schedules with estimated and declared release dates.

In the third phase of development, *Product Development*, Trilogy employs a proprietary application development methodology (Fast Cycle Time, or FCT), to quickly develop new features and applications (box 7).

Although the full four phase methodology is known as FCT, the actual product development phase is an industry-focused evolution of Trilogy's earlier software development methodology, using small teams of Supercoders with automated tests, as described in the cognitive map on software development.

Key customers (or chartered customers) for specific Trilogy divisions may be involved in the testing of early versions of a product. These customers may also have had some say in product ideation via consultants working with them and through to the Solutions Marketing Group.

Trilogy commenced the development of Internet-enabled applications in 1996-97 mainly using Java, C++ and new object-oriented developer applications, plus they were early adopters of XML.

Trilogy's configuration systems saw am almost immediate deployment into online ordering systems in 1996-97 and have become one of the major enterprise-wide e-commerce applications. These applications are now growing strongly in all of Trilogy's targeted industry segments through further customisation and incorporation of new features and e-business technologies.

The final development phase, *Product Delivery,* commences when the product is beyond beta Release (box 8). Prior to final delivery, there is substantial collective interaction between product management, chartered customers, development teams and solutions marketing groups. A formal launch and roll-out of the product is finalised and the product is launched and delivered.

Ongoing product support and maintenance are classified as part of product delivery. Patches, bug fixes and sub-releases are planned and delivered within the product delivery phase. Further customisation and support may be required through consultants working with customers on the deployment and installation of the new product.

The cognitive map on industry-focused application development captures a radical transformation in decision-making associated with application development at Trilogy. There is a logical evolution of Trilogy’s small-team/Supercoder approach to actual application development, but it is now has strict boundaries and is focused toward targeted industries.

The new focus is further reinforced by the embedding of development resources, and most other management and operational resources, within specific industry divisions.

The cognitive map (and associated DSA model) highlights an evolution for some software houses from a product-focused horizontal application and organisational development approach, toward a vertical industry-focused approach.

Such an evolution may be relevant for software houses providing specialised enterprise-focused applications and is relevant for other case-studies in this project (Zaplet in particular also moved from a horizontal to a vertical orientation during 2001).

On the other hand, some software houses in this project have preferred to remain horizontally focused even with its enterprise applications. Kana and NetDynamics/iPlanet have continued mainly as horizontally focused operations, although even in these cases there is increasing evidence of at least some broad vertical categorization of aspects of its products and services.

Trilogy believes there is still great opportunity to further lower sales and marketing costs, and continues to develop new applications to address that challenge in the context of targeted industries.

9.10 Extension Questions

A set of additional questions was put to the interviewees for this case study, covering dominant logic, shared vision, leverage points, and strategic marketing issues. These questions were asked during the initial interviews and followed-up with further validation and commentary in the follow-up interviews.

9.10.1 Dominant Logic

The interviewees agreed that Joe Liemandt has represented dominant logic for driving Trilogy's vision since its inception in 1991. Franke (2002) believed that he contributed to dominant logic for application development methodology, particularly during the 1990s, but also right through to 2002.

Franke (2002) also identified John Price as the dominant logic for the creation of the "Trilogy University" to train and further develop Superprogrammers, during the mid-1990s. Hyams (2002) noted that around 1996, Scott Snyder was also noted as a dominant logic behind Trilogy's software development philosophy and in formalising the actual software development systems.

However, since Trilogy's restructuring in 2001, dominant logic has devolved from Joe Liemandt (who still drives overall company vision) out to managers in the Divisions.

There may be distinct "sub-logics" present within each Division as they work out its own visions and product ideas for its specific industry roadmaps. Hyams (2002) indicated that he provided dominant logic during 2000-02 for the core engineering related to Trilogy's applications, but not for specific products.

9.10.2 Shared Vision

From Trilogy's inception in 1991 through to 2000, the interviewees believed that a lot of people in Trilogy contributed quite strongly to creating and extending shared vision. No one was shy about letting others know what they thought in Trilogy. However, Joe Liemandt was always working on collecting these thoughts and articulating the shared vision.

According to Hyams (2002), in 1998 the Trilogy operations team articulated and presented the vision. The team included Joe Liemandt, Neroj Gupta, John Price, Tom Carter and Chris Porch.

Shared vision has changed since Trilogy's restructuring in 2001, in that while Liemandt still collects and articulates the vision, Divisional General Managers contribute distinctively to it. There is a case to be put that Divisional General Managers may actually articulate a shared vision for and within its own Divisions.

Hyams (2002) noted that during 2001-02 there were other contributors to shared vision within Trilogy, but not necessarily in the Divisions. These contributors included Pat Kelly, Jim Abeld, Debra Ingram, Tom Rowe and David Phillips.

From 1991-97, Trilogy had minimal relationships with third-parties and, apart from direct customisation of its applications for targeted customers; it had not developed a shared vision with third-parties. However, as Trilogy pursued a rapid growth strategy to attain critical mass against larger ERP competitors, it developed an indirect business unit.

Trilogy worked hard with selected partners to build an enterprise-wide COTS e-commerce suite and shared that vision with those partners. Trilogy persisted with the indirect business unit, but complexities in managing the development of a full COTS suite across third-parties combined with the dot.com bust saw Trilogy abandon the unit in 2000.

There may be opportunities for shared vision with key customers within business divisions, and possibly with strategic vertical industry system integrators and some developers, as the industry-focused approach evolves.

A significant evolution of shared vision under the industry-focused approach is that while Trilogy always customised its applications for key customers from 1991-2000, the communication and interaction of ideas and testing with key customers is now directly explicated in Trilogy's software development methodologies.

9.10.3 Leverage Points

Six leverage points were identified during the interviews. These are outlined in chronological order:

9.10.3.1 Moving to Austin (1991-92)

9.10.3.2 Officially Moving to "Selling-chain" Approach (Spring 1996)

9.10.3.3 Converting Development to Internet Applications (1997)

9.10.3.4 Internet Business Spin-off Strategy (Late 1999-2000)

9.10.3.5 Rebranding to e-Commerce Applications (2000)

9.10.3.6 Transition to Vertical Industry Focus (2001)

9.10.3.1 Moving to Austin (1991-92)

Liemandt's decision to move the fledgling company to Austin was attributed mainly to family reasons. A direct follow-up leverage point from moving to Austin was hiring David Franke, whose reputation for software development and networking with various industry organisations enabled Trilogy to successfully pitch its configuration software to at least two substantial computer systems companies that needed its type of application.

Franke immediately set about incorporating software development standards and completed a rewrite of the SalesBuilder application into a formal Version 1 release. Success with acquiring Silicon Graphics and Hewlett-Packard flowed on to signing up several large enterprises – and continuous rapid growth for Trilogy through 1992-99.

Trilogy also attracted high-quality software developers who built on Franke's early standards into a "best-in-class" software development organisation.

9.10.3.2 Officially Moving to "Selling-Chain" Approach (Spring 1996)

Liemandt's original vision articulated sales and marketing activities (and costs) as a "selling-chain". Liemandt believed that sales and marketing costs could be cut by around 2% through automation of this chain. From 1991-96, Trilogy's main contribution to automating this chain was the development and enhancement of its configuration and pricing application, SalesBuilder.

The official move to focusing on the "selling-chain" really meant that Trilogy was declaring its intentions to offer full "end-to-end"/"single product" application suites for the "front-office"/Selling Chain part of an enterprise's operations.

Declaring to the public and Trilogy's customers a full "selling-chain" offering was a clear recognition that sooner or later, the larger ERP application providers would extend out of its "back-office" applications toward full "end-to-end" overall enterprise application suites – meaning direct competition with Trilogy.

Liemandt knew that taking such a position meant that Trilogy had to grow very quickly to be at a critical mass when he expected the ERP vendors (SAP, Peoplesoft, Oracle, and Baan) to extend into the front-office area. Liemandt seemed to expect these incursions to occur around 1998-2000.

Trilogy "bulked-up" rapidly through 1996 to early 2000, growing its "front-office" competitive position. Nevertheless, the larger ERP companies, especially SAP and Oracle made strong inroads into the Front Office area from 2001 to 2003, and now represent very strong threats to almost all major front-office application providers.

Trilogy lost competitive ground during 2000, but restructuring toward targeted vertical industries was an effective competitive response for 2001 to 2003.

9.10.3.3 Converting Development to Internet Applications (1997)

Trilogy realised that its applications needed to become Web-based and switched to development using Java and XML in 1997-99. Trilogy had to hire large numbers of new programmers and staff with Web development skills. Hiring of new staff to develop Internet-enabled applications and systems doubled Trilogy's employees from 1997 to 1999.

Trilogy's acquisition of the new Internet skills and capabilities enabled the company to pursue new Internet-related applications and businesses (see in additional leverage points).

9.10.3.4 Internet Business Spin-off Strategy (Late 1999-2000)

Details on Trilogy's Internet spin-off businesses can be found in Mandel & Austin's (2000) "Trilogy (B)" Case study note. However, the case study note was not included for detailed analysis for this project (that is, for DSA modelling and cognitive mapping) as it contained no reference to application development.

One of Trilogy's founders, Christy Jones, had set up a "dot.com" venture, PcOrder.com, in 1995 using Trilogy's configuration software. As the "dot.com" boom reached its peak in 1999, Trilogy launched several ventures leveraging its "selling-chain" applications. Ventures included:

- CollegeHire.com
- CarOrder.com
- ApplianceOrder.com
- IveBeenGood.com
- InsuranceOrder.com.

Most of these ventures were launched toward the end of the "dot.com" boom and were closed during the bust in these businesses in 2000.

Until 2000, Trilogy was a company completely focused on rapid growth and taking risks to support rapid growth. The "dot.com" bust and general downturn in technology stocks forced Trilogy to consolidate, restructure and eventually to move to a targeted vertical industry-focused organisation.

9.10.3.5 Rebranding to e-Commerce Applications (2000)

Most ERP companies were slow to convert its applications to be Web-based, mainly because they were pre-occupied with implementing large and complex systems from around 1995 to 1999, in time to deal with Y2K contingencies.

Most "front-office" application developers moved to Web-based applications from 1997 onwards, gaining about a three-year advantage over their potential ERP competitors.

By 2000, several front-Office application developers had rebranded its applications as "e-commerce", meaning Internet-based applications.

Trilogy rebranded its applications as "e-commerce" solutions in 2000 joining this particular trend. Trilogy's commitment to the development and deployment of e-commerce solutions was deepened through joint ventures with large enterprises such as Ford (see Mandel & Austin 2000). Some of these ventures and other close partnerships with large customers would contribute to the identification and establishment of targeted vertical industry business divisions in 2001.

9.10.3.6 Transition to Vertical Industry Focus (2001)

The dot.com bust of 2000, plus severe downturns in technology stocks, severely impacted on Trilogy's revenues, customer base and growth philosophies. Trilogy's response was to transform the company into five distinct business divisions, each developing and leveraging core Trilogy applications directly for enterprise customers in targeted vertical industry markets.

Development cycles and philosophies changed. Recruitment focused on experience in targeted industries and not so much on computing or IT knowledge or experience.

At the time of interviewing for the case study (February-May 2002) it was too early to assess any outcomes from Trilogy's transition to a vertical industry-focused organisation.

However, by mid-2003 Trilogy had gained widespread recognition for its vertical solutions, from both its customers and several business and IT analysts. Trilogy added an additional dimension to its industry-based product offering, through linking its ongoing support revenues directly to its customers' own ROI measurements from using Trilogy's applications within its operations (see Kirsner 2002).

The addition of a specific customer ROI measurement to the development and deployment of an application suite can be traced logically back to Trilogy application development methodology and its Industry Roadmaps – but it is (at least in mid-2003) an apparently unique contribution to the software industry from Trilogy – and while it is beyond the scope of this project – it is a development that analysts and researchers are watching with great interest.

9.10.4 Strategic Marketing Issues

Three strategic marketing issues were identified from the interviews:

9.10.4.1 Vertical Industry-focused Marketing

9.10.4.2 Broadening Product Lines/Application Suites

9.10.4.3 Product Management and Product Marketing Roles

9.10.4.1 Vertical Industry-focussed Marketing

Trilogy's transition to a vertical industry-focused organisation has exposed the organisation to a different type of marketing approach, infrastructure and set of operations. For an organisation that was deeply grounded in product development for horizontal use, understanding all aspects of an industry-marketing approach has been a steep learning curve.

Trilogy has expressed its understanding of "industry marketing" by setting up industry-focused Business Divisions, each resourced with industry specialists in all management and operational roles. Perhaps Trilogy's philosophy of excellence in software development is now translated into establishing excellence in industry-based software development – including the supporting organisational structure to go with it.

Hyams (2002) noted that shifting toward an industry-focused approach was not just about the organisational structure, but also moving Trilogy's brand to be seem as associated with targeted industries.

9.10.4.2 Broadening Product Lines/Application Suites

Both interviewees noted that during 1995-96, broadening Trilogy's applications out to a full "front-office" or "selling chain" solution represented a significant strategic marketing issue. Definition of what actually constituted the "front-office", and then developing the expanded application suites were major marketing decisions at that time.

Positioning Trilogy's expanded "front-office" suite to effectively compete with and/or complement the large ERP application suites was and remains a major marketing challenge for Trilogy.

The strategic marketing issue of broadening product lines was not just pertinent for competition against large ERP providers. It was important for defining e-Commerce application suites as new enterprise e-Commerce applications emerged in the late 1990s and in 2000-02. Other software providers such as Kana branded its application suites as e-CRM, for example.

Rebranding of "front-office" applications to be "e" applications took up a lot of strategic marketing attention from 1998-2000. The larger ERP providers rebranded their offering as enterprise e-commerce application suites from 2000-02.

9.10.4.3 Product Management and Product Marketing Roles

The roles of Product Management and Product Marketing operating within a vertically focused organisation were raised as significant strategic marketing issues.

In almost all of the software houses researched for this project, Product Managers worked across both the development group and directly with customers and some key ISVs. As most of the software houses were start-up companies, product marketing was not usually a clearly defined group, apart from a marketing director and perhaps some marketing communications staff. Most software houses had defined sales groups that also worked with the product managers.

The main exception in the project is Intuit QuickBooks with an established marketing organisation and highly defined product management and product marketing roles. Intuit QuickBooks was set up as a B2C or consumer marketing organisation with these roles defined back in the 1980s – and it looks similar to software houses set up during that time (for example, Adobe and Microsoft).

Trilogy's shift to an industry focused structure is similar to other B2B or business-to-business computer hardware and software companies who have attempted to move in a similar direction. At various times during the 1980s and 1990s and in the current decade, IBM, DEC/Compaq, Hewlett-Packard, SAP, TIBCO, and to a lesser extent the other ERP software vendor have all experimented it or established specific industry-focused business units or divisions.

There are variations on how much actual product development or customisation is undertaken by industry business units or divisions. Some units may just undertake marketing and rebranding of the application for a specific industry with virtually no actual product modification. At the other extreme of a continuum, the core application may be quite basic and developed substantially within the industry business unit.

A common variation may be that the core application is significantly customised by third-party partners such as SIs or industry-focused ISVs. In this variation, product management in the industry business unit or division is likely to have a significant coordination role in bringing together the resources for further application development and customisation.

At the time of the interviews for this project, there were product management and product marketing for core applications development within each of the business divisions. Trilogy was trying to work out how both roles could work effectively with development groups to produce better industry-based solutions.

9.11 Summary

Decision-making issues associated with software development at Trilogy from its inception in 1991 to 2002 were captured and mapped in a package of DSA models, events chronology maps and cognitive maps and extension questions.

The Trilogy case study provides deep insight not just into software development and delivery, but also in the underlying philosophies behind developing and changing actual software development over time.

The package also highlights a transformation for Trilogy from a horizontal product-focused organisation through the 1990s to a highly focused vertical industry organisation in 2001-02. The interview process enabled extension of the case study timeframe to analyse Trilogy's transition in 2001-02, in substantial detail..

Key insights from the case study include:

- 9.11.1 Transformation from Product-focused to Vertical Industry-focused Decision-making.
- 9.11.2 Underlying Software Development Philosophies and Their Contribution to Decision-making
- 9.11.3 Decision-making Associated with Anticipatory Competitive Actions
- 9.11.4 Defining and Redefining Product Lines/Application Suites

9.11.1 Transformation from Product-focused to Vertical Industry-focused Decision-making.

The package of models and maps provides deep insights into a transformation in Trilogy's decision-making processes from the late 1990s through to 2002.

Between 1991 and 2000, Trilogy's business was based on very strong basic philosophies for software development, plus a clear vision to provide applications that would enable enterprises to reduce their sales and marketing costs. Trilogy's decision-making was occupied with rapidly developing applications to address an increasing range of “selling-chain” or “front-Office” opportunities.

The original vision presented huge potential revenues from large corporate customers – but only if Trilogy could grow quickly enough to have the resources and infrastructure to service these accounts.

Trilogy managed rapid growth through a very strong internal focus on extraordinary productivity from its development teams. By 1996 Trilogy had expanded its applications into a substantial “front-Office” application suite.

Trilogy used several ventures to leverage its applications into new businesses. However, a number of these ventures were “dot.com” businesses launched at the peak of the Internet boom in 1999. While they logically leveraged Trilogy’s front-office applications capabilities online, they were also early casualties in the dot.com bust in 2000.

Trilogy’s transformation to a vertically oriented organisation was a strong response to the serious downturn in its revenues in 2000. The main links between the old horizontal Trilogy and the new vertical organisation were key large customers who formed the basis for identification and establishment of Trilogy Business Divisions.

Almost everything about the Trilogy organisation was reworked into the new Divisional structure. Trilogy’s software development methodology was formally reworked to operate within each Business Division to create and deliver applications specifically tailored to Industry Roadmaps. Management and operations resources within Divisions were selected with emphasis on specific industry experience rather than IT industry experience.

The DSA package for the Trilogy greatly benefited from multiple interviews which enabled the creation of new models and maps to effectively capture the transformation to a focus that was not necessarily an obvious evolution or extension from Austin’s (1998) case study and Austin & Mandel’s (2000) case supplement addressing Trilogy’s Internet businesses.

9.11.2 Underlying Software Development Philosophies and Their Contribution to Decision-making

Trilogy set a vision to be the very best software development organisation in the world. The organisation’s decision-making processes reflected and strengthened this vision.

Underlying philosophies based on using small-teams of highly productive Superprogrammers, supported with automated testing routines and systems were factored into recruitment, training, and also to rapid application development and delivery.

These philosophies were heavily tested but maintained as the organisation brought in large numbers of Internet-skilled programmers and developers in 1997-98 in order to rapidly convert Trilogy's applications to be Internet-enabled and eventually to be branded as "e-commerce" applications.

Within 12 months of this influx of Internet-based resources, rapid software development enabled sufficient leverage of Trilogy's front-office applications into new online or dot.com ventures.

Since 2001, most underlying software development philosophies remain, but are now incorporated in a software development methodology driven directly by specific vertical industry opportunities and requirements.

There is less decision-making autonomy at the development team level, with substantial decision-making now residing at the Business Division Management Level with key inputs from various operations groups within that Division.

9.11.3 Decision-making Associated with Anticipatory Competitive Actions

Trilogy's decision-making processes highlight a strong concern for potential competitor actions. From 1993 onwards, Trilogy was tracking the main ERP vendors as potential competitors – and this was a major driving factor for pushing a rapid growth strategy through the mid-1990s.

Trilogy's management was certain that there was small timeframe to grow the company to a critical mass from where it could withstand aggressive competition from the likes of SAP and Oracle and other large ERP vendors. Fear (and some actions resulting from it) emanating from this perspective was identified and mapped in DSA analysis and cognitive mapping for this case study.

Decision-making associated with anticipatory competitive action was not highlighted in the transition to a vertically-focused organisation. Rather, decision-making seems to have shifted toward directly addressing specific customer requirement, ahead of competitive action. That would not mean that focus on competitive action has gone, but it is now viewed more in an industry-based context.

9.11.4 Defining and Redefining Product Lines/Application Suites

Trilogy's founder, Joe Liemandt, articulated a strong vision for Trilogy at its inception – to develop applications that could cut sales and marketing costs by around 2% of revenues for a typical large corporation.

Development of a full set of "front-office" applications caused Trilogy to realise the vision was not realistic for a start-up company. Trilogy commenced with an application for configuration and pricing of computer hardware and software.

Nevertheless, within three years of the launch of the official Version 1 of its SalesBuilder application, Trilogy was able to offer a comprehensive "front-office" application suite and reposition the company as an "end-to-end front-office" vendor.

In 1997-98 Trilogy converted its application suite to be Internet-based and, by 1999-2000, rebranded its application suite as an enterprise "e-commerce" suite.

Since 2001, Trilogy's applications have been defined, developed and mapped against Business Division specific industry and product roadmaps. The Business Divisions now work directly with key or "chartered" customers on large complex front-office solutions.

These solutions may connect to larger ERP offerings or to specific systems developed by customers for specialised business functions (examples include: Automotive Product Management application suite for Ford; several pricing and contract management systems for various computer companies, a new global contract management and pricing system for British Airways; and a Channel Management System for the Prudential Insurance Company of America – for more details on these examples see Trilogy 2003a and b).

9.12 Conclusions

The Trilogy case study highlights the decision-making associated with an organisational culture that exalts excellence in application development. DSA modelling augmented with events chronology and cognitive maps provide analysis and insights into rapidly building an enterprise application software house – and then transforming the whole organisation toward a vertical industry-based focus.

Conclusions are presented in the sections below:

9.12.1 Theoretical Issues (or Questions) for the Trilogy Case Study

The Trilogy case study explores the decision-making associated with combining a philosophy of "best-practice" application development with risk-taking in order to grow a company rapidly to prepare for an eventual and imposing competitive showdown with larger and more established corporations.

The case study also provides insights into decision-making from a big vision (automating enterprise sales & marketing activities) into a focused application offering (pricing configuration software) and onto a major provider of vertically focused (industry-focused) applications.

The Trilogy case study offers some special contributions to the overall research project through analysis of explicit philosophies on application development that appear to be as fundamental to the organisation as growing a strong and profitable software powerhouse.

Trilogy's strict and high-quality application development standards make it the most internally focused software house in the research project. These philosophies and standards provide a special contrast to all the other software houses in the research project (with the exception of the early stages of the Intuit QuickBooks case study).

The two extended storytelling methods used for analysis in the advanced hermeneutic analysis and extended DSA were effective at identifying key decision-making issues, events, and linkages associated with the dynamic evolution of Intuit QuickBooks' applications in terms of transitions to a more open and partner-oriented application development approach and experimentation with a number of online strategies for the application.

Both methods provided mapping and validating of initial accounts of decision-making, followed up with subsequent revisions of those accounts using a systematic emic/etic representation system – including development of new DSA models and maps for new application development approaches in 2002-03.

Interviews for the Trilogy case study captured substantial insights beyond Austin's (1998) and Mandel & Austin's (2000) case study, particularly on Trilogy's new industry-focused application development cycles. Additional material provided by Trilogy enabled the most comprehensive insights into actual application development cycles out of all the cases for this project.

9.12.2 Trilogy Case Study Limitations

Complete sets of multiple interviews with multiple respondents who were directly involved in the original case study on the company enabled effective development and validation of the original set of events chronology map s and cognitive maps, plus request and validation of the new DSA model and cognitive map.

More insights into Trilogy's transition to straight Internet-based applications during 1997-99 would have been desirable and more in line with the overall objectives of the research project. However, this issue was partly addressed through the collection of additional information on Internet development and Internet business initiatives during the interview process.

9.12.3 Contributions from the Trilogy Case Study to Strategic Sensemaking

The creation and validation of a set of DSA models plus cognitive mapping outputs represent an effective mapping of strategic sensemaking over time and in multiple contexts. This extended DSA approach offers insights into turning an idea into an application in conjunction with turning a small start-up business into a substantial and competitive enterprise software provider.

Analysis indicates that Trilogy's executive management team was driven to take substantial business and application risks in order to quickly reach a "critical mass" such that they could take on severe competition from much larger and more established ERP software houses. Both Trilogy and its "selling chain" competitors perceived that its niche would develop into in a larger market for enterprise-wide application suites.

DSA models and selected maps for the Trilogy case provide unusual insights into a very strong logic for setting innovative application development processes and cycles. In fact, Trilogy branded its Fast Cycle Time (FCT) application development system.

Trilogy has a special system of recruitment and training to instil shared vision and empowered decision-making at the individual developer level in projects.

The Trilogy case study offers mapping over time of decision-making associated with transformation from a "best-practice" product-driven business and application development approach into a vertical industry-focused business.

9.12.4 Contributions from the Trilogy Case Study to Practice

The Trilogy case study provides valuable insights into establishing "best-practice" application development groups. Trilogy's strong organisational culture around "super-programming skills, capabilities and testing excellence through recruitment and training are explored in the case study.

The interview process for the Trilogy case study provides valuable insights into transforming a software house from a strong product development focus across to industry-focused business groups and application development.

9.12.5 Trilogy Case Study Endnote

Trilogy is viewed as an innovative enterprise e-commerce (also seen in the media as an e-business) application provider and, although beyond the scope of the current project, new initiatives such as linking customer ROI to its ongoing use of Trilogy's applications could set the stage for further significant redefinition of what constitutes an "e-business" application suite (see Kirsner 2002).

Trilogy is also working on defining new measurements for customer satisfaction relating to enterprise e-commerce application suites incorporating business value metrics (see Sawhney 2003).

During 2003, Trilogy packaged its Fast Cycle Time (FCT) application development methodology. FCT Web-based tools and supporting services are presented in Exhibit 9.11 (see also Trilogy 2003c).

Packaging FCT for Trilogy represents the translation of fundamental application development philosophy and decision-making directly into a set of application management services – an unusual but very interesting offering over and above enterprise e-commerce application suites.

Exhibit 9.11 Trilogy – “Fast Cycle Time” (FCT) Methodology Tools

From methodology to reality

Other project methodologies rely almost entirely on abstract processes and organisational discipline. The Trilogy Fast Cycle Time methodology, however, includes a set of leading edge, Web-based "vision management" tools for automating critical aspects of the methodology, along with managed hosting services that provide end-to-end quality and availability for FCT projects:

Leadership.com is a multimedia Web-based application that allows executives to record project vision in a central location and makes the vision available to all project members.

eFeedback is a Web-based tool to elicit and consolidate user feedback, both on a Leadership.com site and on the target ebusiness Web site.

ePrioritise and eRankIt tools support project-level triage of features and requirements. The innovative combination of philosophy and technology that underlies the Trilogy Fast Cycle Time methodology keeps both project and vision synchronised, providing a sound foundation for e-business project management.

Managed hosting services fully integrate the FCT methodology with computing, storage, and networking resources, providing a solid foundation of quality, availability, and administrative responsiveness required for FCT project success.

Maximize business value with FCT

The Trilogy Fast Cycle Time methodology offers a superior approach to managing and developing e-business solutions, one that maximizes business value by constantly measuring progress and incorporating stakeholder feedback. To learn more about the FCT methodology, call the Trilogy Project Office at **1 .877.292.3266** or send e-mail to **PMO@trilogy.com.**

Source: Trilogy (2003c).

Chapter 10: Theory Development, Generalisation and Conclusions

10.1 Introduction

This research project addressed several questions:

- How can we craft accurate, rich and insightful stories on marketing decision-making?
- How can we unearth those nuggets of sensemaking gold that help us to learn more about how marketing decisions occur – or marketing decisions yet to be executed?
- How can we understand marketing decision-making applied to the development of new forms of codified knowledge such as application software, which are the building blocks of a new marketing information environment, and of new marketplaces (or "marketspaces")?
- Are there new issues and challenges that confront decision-makers as they develop applications to extend and enhance the Internet and its multimedia service, the World-Wide Web, or as products to be distributed through this new environment?
- Is this new environment different for B2B marketing decision-making from existing B2B manufacturing and high-technology contexts?

The project has focused on exploring, analysing and "making sense" of stories or accounts of application software houses developing new applications for, and with, the Internet. Detailed stories analysed by review and revisitation would contribute deeper insights into the decision-making processes.

The storytelling methodology developed during the project is a substantial contribution in the form of a dynamic and interpretive storytelling approach of researching marketing decision-making. The methodology incorporated a "sensemaking" perspective for understanding marketing decision-making.

By the end of the project there was a largess of insights into both developing software applications for a revolutionary new "world", that is, the Internet, and creating and rapidly growing new software houses. The three main contributions suggested at the outset of this report have been delivered:

1. Insights into the decision-making processes for the development and delivery of disruptive software applications (or applications contributing to other disruptive technologies (a new context for B2B marketing decision systems analysis (DSA)
2. The development of a new storytelling methodology based on a hermeneutic framework that is well suited to the exploration of B2B marketing accounts
3. The development of a new cognitive mapping system to enhance sensemaking of marketing decisions in a B2B context.

Overall, the new storytelling methodology validated most of the insights and analysis in the original accounts of decision-making used as base data input for the case studies used in the project – but exciting additional insights were gained for all six case studies. Furthermore, the package of analysis highlighted very strong and challenging issues and questions for further research.

Findings from the six case studies in the research project are highlighted, discussed and consolidated along four dimensions:

1. a direct case-by-case discussion of higher level hermeneutic analysis
2. a non-theory-driven discussion on the case-study findings (including development of a composite DSA model)
3. application of findings to aspects of a sensemaking cognitive theory of the firm, including revision and addition to propositions associated with software industry Structuration.

The first dimension is a direct case-by-case discussion of higher level hermeneutic analysis. While there is room for substantial cross-case analysis, discussion – and even development of a full cross-case hermeneutic "metasystem", discussion in this chapter was contained to the case-by-case hermeneutic analysis leaving the further "metasystem" development as a project following on from this research project.

The second dimension is a non-theory-driven discussion on the case-study findings. Findings from each of the cases are noted and discussed. The most rigorous cross-case analysis section was the development of a composite DSA model, whereby all case-study extracts for all case studies were consolidated into one datafile and passed through a range of analytical tools to produce a robust and comprehensive cross-case DSA model for software application development.

The third dimension attempted to apply cross-case-study findings to aspects of a sensemaking cognitive theory of the firm; including the revision, recasting and development of new propositions associated with software industry Structuration (see Huff, Huff, Barr & Stimpert 2000).

The report wrap-up offers conclusions relating to the research project, including discussion, research limitations and recommendations for future research. Figure 10.1 sets out the structure for Chapter 10.

10.2 Advanced Hermeneutic Analysis – Discussion and Case-study Findings

The six case studies in the research project represent an analysis of sensemaking over time, including emic and etic explanations of the decision-making associated with the development and delivery of Internet-based applications. All six cases revisit past decision-making as described in the Harvard Business School Case Studies that were used as base accounts for each case study in the project.

The exhibits drawn from each case study depicting the highest level of hermeneutic analysis are cumulative "meta-sense-maps" or representations of dynamic systems – and, in the case of emerging Internet-based applications – are actually effective representations of complex adaptive systems (see Exhibits 1-6).

10.2.1 Advanced Hermeneutic Analysis – Red Hat Case Study

The Red Hat case study has been advanced as an application of the proposals described in Chapter 1 for advancing hermeneutic research in B2B decision-making. At least two rounds of emic data and three rounds of etic data were included in five levels of hermeneutic analysis for the study. A summary of the five-level hermeneutic analysis for the case is presented in Exhibit 10.1.

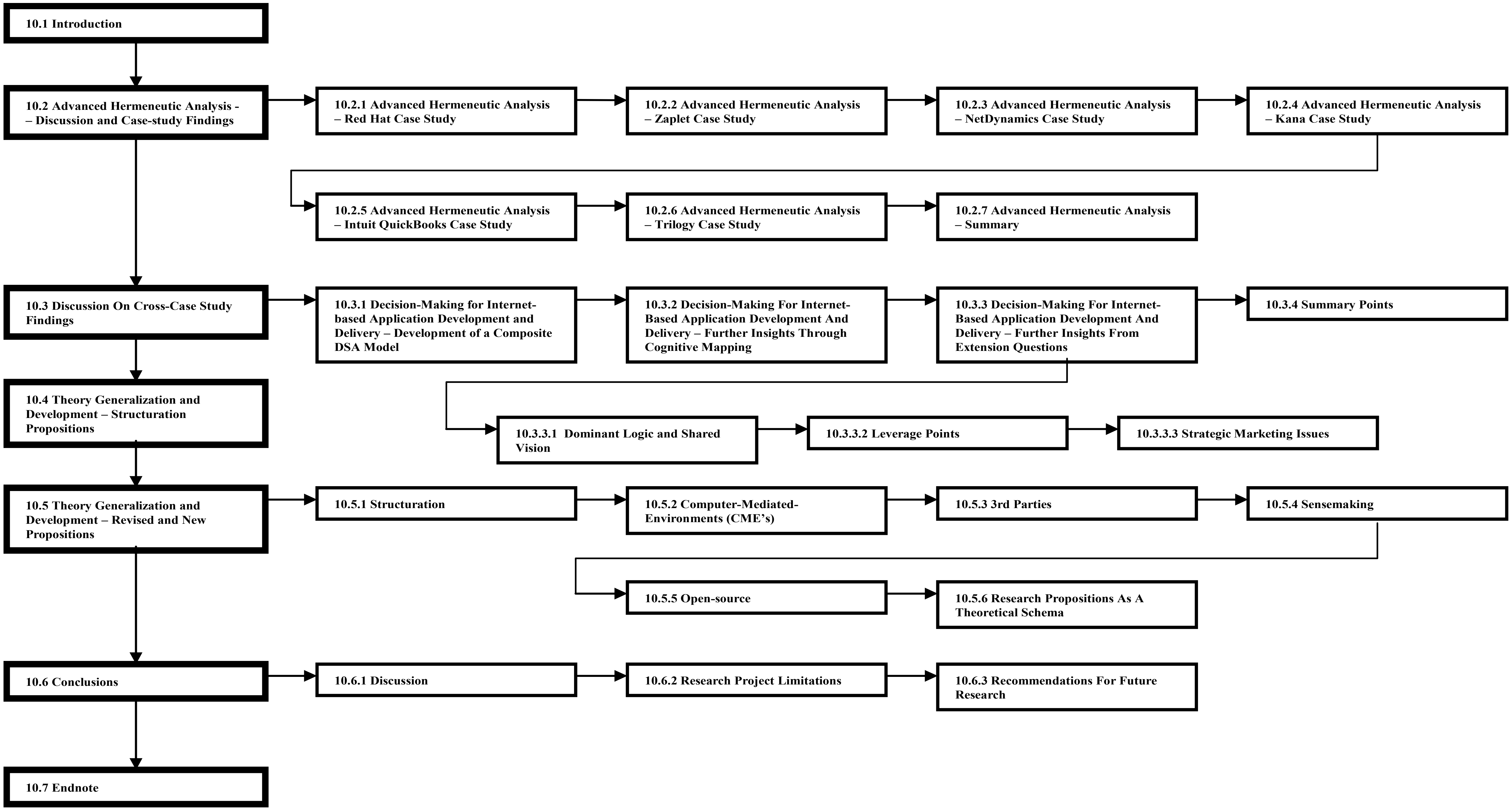


Figure 10.1 Theory Development, Generalization And Conclusions – Chapter Structure.

Exhibit 10.1 Level V Analysis for Red Hat

Emic 2 view:
•Reaffirmed all of Emic 1 view with extensions for open-source philosophy at Red Hat and Red's symbiotic relation with the open-source community
• 90 percent of Red Hat's Revenues for 2001 came from support activities
•Red Hat is like a fabricator such as Boeing – it assembles over 800 applications of which only about 35-40 are actually created by Red Hat
•Red Hat is building up their 3rd party and ISV Programs
•Perhaps Red Hat may develop a marketing structure similar to established software houses

Etic 1 view:
•Account of Evolution of Open-Source Software Philosophy – Creation of Free Software Foundation (late 1980's); GNU Project (late 1980's); General Public License; Emergence of Open-Source Initiative (OSI – late 1990's)
•Account of Linus Torvalds' RDC of Linux
•Description of Emergence of Red Hat
•Description of Red Hat's application development and distribution

Emic 1 view:
•"The one unique thing we could do that no else could do was, for the first time, we were giving control of the operating system to the user"
•Successful IPO of a company marketing software that supported free products (i.e. the Linux open-source operating system)
•"Not bad for a company that gives its products away for free"
•Would an open-source development approach work for developing applications that could run on Red Hat's Linux operating system?

Etic 2 view:
•New open-source development community and philosophy
•Red Hat created from Complimentary Needs Between Ewing &Young
•Importance of Internet for open-source community application development
•Red Hat's approach to symbiotic interaction with open-source development community

Etic 3 view:
Etic 2 views validated plus:
• Extension of open-source approach to developing advanced software applications
• Open-source approach applied to software-based information products and services
• "Open-source" marketing concept

1993-1999 Initial mental model:
•Bundle a set of applications to enable Linux to be easily installed and maintained (1st version released in 1993)
1993-2001. Extended mental model:
•Initial mental model with Release dates for Red Linux V3 (Set-Dec 1995); V4(Oct 1996); V6 (May 1995)
•Shift to focus on providing Red Hat Linux to major computer hardware vendors for their Enterprise/Large Account Customers (e.g. IBM, Dell); revenues mainly from providing technical support to large customers
Decisions-actions:
•Start-up of support tools for managing Linux open-source operating; market launch without beta test; RDC of 6 new releases; Absorbed into Red Hat" in 1995 on 3rd release (became Red Hat Linux Version 1); IPO August 1999; Red Hat V7 (Sep 2000); V8 (Sep 2002)

Overall, the five-level hermeneutic framework for the Red Hat case study has provided a strong and comprehensive framework for capturing and extending dynamic sensemaking for Red Hat's decision-makers.

Key etic 1 issues highlighted in the original case-study account (MacCormack & Herman 2000) included:

- Account of the evolution of open-source software philosophy – creation of Free Software Foundation (late 1980s); GNU Project (late 1980s); General Public Licence; emergence of open-source initiative (OSI – late 1990s)
- Account of Linus Torvalds' RDC of Linux
- Description of the emergence of Red Hat
- Description of Red Hat's application development and distribution.

Etic 2 and etic 3 issues highlighted in further levels of analysis for the research project included:

- New open-source development community and philosophy
- Red Hat created from complementary needs between Ewing &Young
- Importance of Internet for open-source community application development
- Red Hat's approach to symbiotic interaction with open-source development community
- Extension of the open-source approach to developing advanced software applications
- Open-source approach applied to software-based information products and services
- "Open-source" marketing concept.

Advanced hermeneutic analysis not only enables an exploration of deep changes in the application development philosophy at Red Hat, but also opens the door to articulating and developing further concepts for open-source application development toward information and knowledge development, and toward a redefined "open-source" marketing concept.

10.2.2 Advanced Hermeneutic Interpretation – Zaplet Case Study

The Zaplet case study is the second case study for the application of the advanced hermeneutic research approach developed for the research project. At least two rounds of emic data and three rounds of etic data were included in five levels of hermeneutic analysis for the study. A summary of the five-level hermeneutic analysis for the case is presented in Exhibit 10.2.

Overall, the five-level hermeneutic framework for the Zaplet case study has provided an effective systematic framework for capturing and extending the dynamic sensemaking for Zaplet's decision-makers.

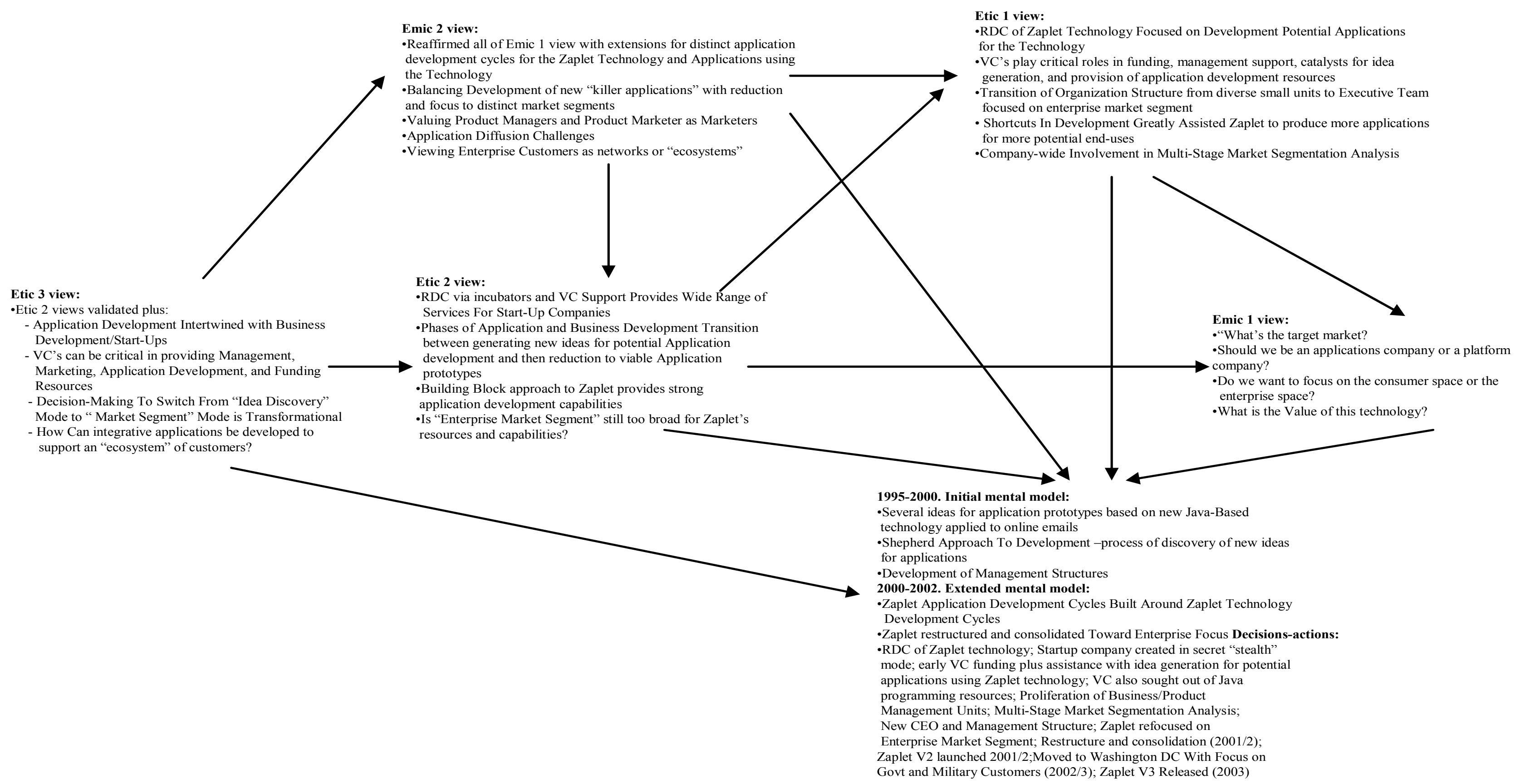
Key etic 1 issues highlighted in the original case-study account (DeLacey & Leonard 2001) included:

- RDC of Zaplet technology focused on development potential applications for the technology
- VCs played critical roles in funding, management support, catalysts for idea generation, and provision of application development resources
- Transition of organisational structure from diverse small units to executive team focused on enterprise market segment
- Shortcuts in development greatly assisted Zaplet to produce more applications for more potential end-uses
- Company-wide involvement in multi-stage market segmentation analysis.

Etic 2 and etic 3 issues highlighted in further levels of analysis for the research project included:

- RDC via incubators and VC support provides wide range of services for start-up companies
- Phases of application and business development transition between generating new ideas for potential
- Application development and then reduction to viable application prototypes
- Building-block approach to Zaplet provides strong application development capabilities
- Is "enterprise market segment" still too broad for Zaplet's resources and capabilities?
- Application development intertwined with business development/start-ups
- VCs can be critical in providing management, marketing, application development, and funding resources

Exhibit 10.2 Level V Analysis of Zaplet Case



- Decision-making to switch from "idea discovery" mode to "market segment" mode is transformational
- How can integrative applications be developed to support an "ecosystem" of customers?

Advanced hermeneutic analysis enabled an exploration of the decision-makers considering the potential uses for a new software technology platform, including moderation from motivated venture capitalists. The analysis also explored Zaplet's transition to working with large customers in defined markets. Overall, the five-level hermeneutic framework for the Zaplet case study has provided a comprehensive framework for capturing and extending the dynamic sensemaking for Zaplet's decision-makers.

10.2.3 Advanced Hermeneutic Analysis – NetDynamics Case Study

The NetDynamics case study is the third case study for the application of the advanced hermeneutic research approach developed for the research project. NetDynamics is also the case study in which a selection of DSA models and maps were submitted to an expert panel for an audit of the processes and depiction of qualitative data collection. At least two rounds of emic data and four rounds of etic data were included in six levels of hermeneutic analysis for the study. The audit system was based on Hirschman's (1986) Humanistic Inquiry validation criteria. A summary of the six-level hermeneutic analysis for the case is presented in Exhibit 10.3.

Overall, the six-level hermeneutic framework for the NetDynamics case study has provided an effective systematic framework for capturing and extending the dynamic sensemaking for NetDynamics' decision-makers.

Key etic 1 issues highlighted in the original case-study account (Iansiti & MacCormack 1996) included:

- Development of applications in "Internet-time"
- Daily builds of application with internal and third-party inputs
- Commenced the actual development before choosing the application language
- Rush to incorporate the key new application into the upcoming release of an industry-standard Web browser
- Importance of delivering an effective proof-of-concept to analysts and supporters
- New approach to application development (one case within a group to demonstrate "Internet-time application development).

Etic 4 view:
•Ethnographic Validation of Cognitive Maps Supporting Etic 2 analysis– by expert "judges"
•Generally validated the cognitive mapping process
•Judges looked forward to full Etic 3 conclusions and insights drawn from further analysis of Emic 2 views and inputs – beyond models, maps and early notes of findings

Exhibit 10.3 Level VI Analysis of NetDynamics Case

Etic 1 view:
•Development of applications in "Internet-time"
•Daily Builds of application with internal and 3rd parties inputs
•Commenced actual development before choosing application language
•Rush to incorporate key new application into upcoming release of industry standard Web-Browser
•Importance of delivering an effective proof-of-concept to analysts and supporters
•New approach to application development (one case within a group to demonstrate "Internet-Time application development)

Emic 2 view:
•Reaffirmed all of Emic 1 view with extensions for details on:
- Initial RDC of SPIDER
- Selection of Software Engineers
- Details of "Proof-Of Concept"

•High risks of using Java in Beta-Testing stages
•High uncertainties associated with determination and incorporation of features and applications in new Netscape Browser
•Go with incorporation of NetDynamics into new Browser in 60-90 days or next version in 12 months time?
•Huge challenges for small start-ups to manage rapidly growing enterprise installed base
•How to price an emerging operating system independent enterprise
•application

Emic 1 view:
•Development of a key Web-based application technology in "Internet-Time"
•"Proof-of-Concept" sufficient for analysts and investors to support NetDynamics
• Java Technology was not final choice for development language when project commenced
• NetDynamics noted that Java was likely to be incorporated into next Netscape Browser Release and took a risk on developing NetDynamics to work with both Java and Netscape technology and applications
•Project was successfully completed within 90 days

Etic 3 view:
•Etic 2 views validated plus:
- Concurrent ISV Application Development
- Diffusion of key Application Technologies for emerging computer mediated environments (CME's)
- Systems Integrators As Extended Virtual Application Marketers
- Paths and transitions for Start-ups with enterprise Applications

•A great account of genesis of Web Services

Etic 2 view:
•NetDynamics application is a "killer application" and a fundamental building block in development of e-commerce application suites
•NetDynamics applications was a foundation for the development of Web Services
•Were NetDynamics really as flexible about application language selection as claimed in the case?
•Had NetDynamics actually already decided about Java before starting the project?
•Triple ISV Beta-Testing and Application Development

1995-1996. Initial mental model:
•Identification of Potential Critical Internet-Based Business Application (Web Page access to Database records)
•Extension of Application from Unix into cross-platform environment
•Flexible about application development language
•Must deliver as part of and by the next release of Netscape Communications Web Browser
•Must have "proof-of concept" for investors in 30 days
•Must be prepared to completely redesign the application with different languages if necessary

1996-2001. Extended mental model:
•Expansion toward a key enterprise e-commerce application
•Funding and Enterprise installed based management issues prompt support from Sun Microsystems

Decisions-actions:
•RDC as SPIDER UNIX (1995) –Renamed to NetDynamics V1 mid-1995; Plan to redevelop for both UNIX and WINDOWS; Plan revised to make new application "cross-platform" and operating system independent; Software Engineers recruited mainly with Windows experience; Engineers "learn' Java sufficiently quickly to refocus Project toward that technology; "Proof-of-Concept" demonstrated successfully (Mar 1996); application revised daily in development phase; NetDynamics 2.0 released June/July 1996; NetDynmaics V3.0 released Oct 1996; NetDynamics V4.0 released Dec 1997; Sun Acquires NetDynmaics Jul 1998; NetDynamicsV5.0 released Mar 1999 – becomes I-Planet IAS Application becomes part if SUN ONE Web Services (2001)

Etic 2 and etic 3 issues highlighted in further levels of analysis for the research project included:

- NetDynamics' application is a "killer application" and a fundamental building block in the development of e-commerce application suites
- NetDynamics' application was a foundation for the development of Web Services
- Was NetDynamics really as flexible about application language selection as claimed in the case?
- Had NetDynamics actually already decided about Java before starting the project?
- Triple ISV beta-testing and application development
- Concurrent ISV application development
- Diffusion of key application technologies for emerging computer-mediated environments (CMEs)
- Systems integrators as extended virtual application marketers
- Paths and transitions for start-ups with enterprise applications
- A great account of the genesis of Web Services

The sixth level of analysis incorporated etic 4 data from the panel of experts auditing the DSA models and maps development process. Key etic 4 comments included:

- Ethnographic validation of cognitive maps supporting etic 2 analysis – by expert "judges"
- Generally validated the cognitive mapping process
- Judges looked forward to full etic 3 conclusions and insights drawn from further analysis of emic 2 views and inputs – beyond the models, maps and early notes of findings.

Advanced hermeneutic analysis has enabled an exploration into quite detailed insights into the rapid development of a new and revolutionary application to enable effective enterprise use of the World-Wide Web. The analysis also explored NetDynamics' transition to servicing large enterprise customers and its ultimate acquisition by Sun Microsystems.

10.2.4 Advanced Hermeneutic Analysis – Kana Case Study

The Kana case study is the fourth case study for the application of the advanced hermeneutic research approach. At least two rounds of emic data and three rounds of etic data were included in five levels of hermeneutic analysis for the study. A summary of the five-level hermeneutic analysis for the case is presented in Exhibit 10.4.

Exhibit 10.4 Level V Analysis of Kana Communications Case

Etic 1 view:
•Email management opportunity found while seeking different opportunity (online shopfronts for sports goods manufacturers)
•Expansion to inbound and outbound email management through acquisitions
•Transformation to supplier of e-commerce application through acquisitions and mergers with funding from an IPO
•Channels options focused on expanding Kana Online (KOL) and developing an Application Service Provider (ASP) group

Emic 2 view:
•Importance of working with “Lead-Users” on learning, development and implementation of new applications
•Focus on transformation and repositioning from a small software house to a major software infrastructure provider
•Emphasis on quality, specialization and solving specific online issues
•Balancing between building applications Versus acquiring applications and skills
•Focus on special application development and porting efforts for major hardware and technology partners
•Roles of System Integrators (SI’s) and Independent Software Vendors as extended sales & marketing and developer resources

Emic 1 view:
•Transition from accidental discovery of opportunity – managing emails – to new application
•Development of online communications management application mainly through acquisition of software houses with appropriate applications
•Founder (Gainey) always thought that Kana should add outbound e-mail management capability to its offering
•Kana wanted to be a leader offering a platform solution that could link online communications products to additional e-commerce applications
•Three pieces of infrastructure needed to build a billion-dollar e-commerce operation – content, to manage the transaction process, and manage the communication process
•Comprehensive channels options including Direct Salesforce, Key Business Partners and Kana Online (KOL)
•Investigation of Application Service Provider (ASP) channel option

Etic 3 view:
•Etic 2 views validated plus:
- Decision-Making For Rapid Development and Release of Internet-Based Applications
- Importance of System Integrators (SI’s) and ISV’s in marketing, referral and further application development and customization

•Broadbase effectively took over Kana but kept building up the Kana brand and applications
•Kana survived as a leading eCRM provider repackaging its applications into Kana iCare in mid-2001 and releasing Kana IQ V8 in Sep 2003

Etic 2 view:
•Founders worked directly with early customers to development an inbound email management application
•Focus on rapid acquisition of software houses to build up online communications application capability
•Management combined efforts on acquisition and application development during IPO process
•Adaptation of strategy toward offering a complete e-commerce application suite (still with emphasis on online communication management)
•Constant acquisitions and mergers to build up critical mass and application development and service capabilities
•Channels Partners for extended sales/marketing and developer resources (particularly SI’s)

1996-2000. Initial mental model:
•Initial business opportunity identified as setting up online storefronts for Sports Goods manufacturers
•Identified a different issue – complaints associated with managing emails
•Rapid Application Development directly with customers
•Rapid Expansion into Online Communications Management Through Acquisition of software houses with relevant applications
•Transformation to e-commerce provider through further acquisitions and mergers
•Establishment of Channels to service Enterprise Customers
•Exploration of Application Service Provider (ASP) model
2000-2001. Extended mental model:
•Full eCRM application provider focused on Enterprise Customers
Decisions-actions:
•Worked directly with customers to manage emails; Developed Kana V1 (Nov 1997), V2 (Mar 1998); V3 (Dec 1998 with SDK); Expanded online communications management capabilities through acquisitions of Connectifi, BEI, NetDialog; IPO Sep 1999; V4 released (Oct 1999); V5 released (Dec 1999); Move to full e-commerce application provider through merger with Silknet (2000); Kana V6 eCRM suite released (mid-2000); Kana merges with Broadbase and releases Kana V7 (mid-2001)

Overall, the five-level hermeneutic framework for the Kana case study has provided an effective systematic framework for capturing and extending the dynamic sensemaking for Kana's decision-makers.

Key etic 1 issues highlighted in the original case-study account (Sarvary 2000) included:

- E-mail management opportunity found while seeking different opportunity (online shopfronts for sports goods manufacturers)
- Expansion to inbound and outbound e-mail management through acquisitions
- Transformation to supplier of e-commerce application through acquisitions and mergers with funding from an IPO
- Channels options focused on expanding Kana Online (KOL) and developing an Application Service Provider (ASP) group.

Etic 2 and etic 3 issues highlighted in further levels of analysis for the research project included:

- Founders worked directly with early customers to develop an inbound e-mail management application
- Focus on the rapid acquisition of software houses to build up online communications application capability
- Management of combined efforts on acquisition and application development during the IPO process
- Adaptation of strategy toward offering a complete e-commerce application suite (still with emphasis on online communication management)
- Constant acquisitions and mergers to build up critical mass and application development and service capabilities
- Channels Partners for extended sales/marketing and developer resources (particularly SIs)
- Decision-making for the rapid development and release of Internet-based applications
- Importance of System Integrators (SIs) and ISVs in marketing, referral and further application development and customisation
- Broadbase effectively took over Kana but kept building up the Kana brand and applications

- Kana survived as a leading e-CRM provider, repackaging its applications into Kana iCare in mid-2001 and releasing Kana IQ V8 in Sep 2003.

Overall, the five-level hermeneutic framework for the Kana case study has provided a strong and comprehensive framework for capturing and extending the dynamic sensemaking for Kana's decision-makers over a number of phases of application development, evolution, and business growth – largely through the acquisition of other software houses.

10.2.5 Advanced Hermeneutic Analysis – Intuit QuickBooks Case study

The Intuit QuickBooks case study is the fifth case study in the research project in which the proposed advanced hermeneutic research approach was applied. At least two rounds of emic data and three rounds of etic data were included in five levels of hermeneutic analysis for the study. A summary of the five-level hermeneutic analysis for the case is presented in Exhibit 10.5.

Overall, the five-level hermeneutic framework for the Intuit QuickBooks case study has provided an effective systematic framework for capturing and extending the dynamic sensemaking for Intuit QuickBooks' decision-makers.

Key etic 1 issues highlighted in the original case-study account (Kochikar & Lal (2000) included:

- Intuit QuickBooks was a successful application with more than three million users and over one million upgrades and new versions sold each year
- Intuit QuickBooks was slow to move into online business
- Intuit needed to develop an online strategy focused on providing online services to complement QuickBooks' offline package
- The first online strategy to add a small business channel to Quicken.com was not successful
- The revised online strategy included a channels framework to evaluate potential service providers and developers and to tier them as different levels of relationship with QuickBooks.

Exhibit 10.5 Level V Analysis of Intuit QuickBooks Case

Etic 1 view:
- Intuit QuickBooks was a successful application with more than 3 million users and over 1 million upgrades and new versions sold each year
- Intuit QuickBooks were slow to move into online business
- Intuit needed to develop an online strategy focused on providing online services to complement QuickBooks offline package
- First online strategy to add a small business channel to Quicken.com was not successful
- Revised online strategy included a channels framework to evaluate potential service providers and developers and to tier them different levels of relationship with QuickBooks

Emic 2 view:
- Developing Value Beyond the Basic product or Software Application – In-House Or Through Partners
- Importance Cross-Selling Products Through All Of Intuit's Channels
- Using the Internet To Communicate Marketing And Alliance Programs
- Channels Selection Process for Revised online Strategy was more opportunistic than a clearly defined framework
- Development of a complete Online Developer and Services Provider Community associated with Intuit Inc – and QuickBooks
- QB Economy Umbrella Branding for QuickBooks Online Services

Emic 1 view:
- QuickBooks's future success depended on its ability to quickly and effectively generate incremental, recurring revenues by harnessing the Internet to offer value added services to its installed base of customers
- (Villanueva) championed the transformation of Quicken from a solely desktop product to a hybrid product with a strong online component
- To develop online strategy balancing QuickBooks to effectively compete without compromising the values and culture of the company (rigorous development standards and strict development and release cycles)
- Cook (Intuit founder) believed that ad-supported Internet would trigger explosive growth in the software industry, just as ad-supported television triggered explosive growth in the media industry.
- Developed a simple framework to help review the partnership proposals objectively (part of revised online strategy)
- Challenges with developing tiered channels strategy for prospective partners

Etic 2 view:
- Initially an experimentation view with online business – e.g. collect orders, then try out a portal approach
- Small Business Channel on Quicken.com really not reach their small business customers who were not regular users of the Internet
- Revised Online strategy included a rigorous partner selection framework – partly to accommodate issues with stringent quality controls within Intuit Inc

1983-1998 (Offline)/1998-1999 (Online). Initial mental model:
- Intuit Inc established in 1983 – Focus on Financial Software Industry
- Intuit releases QuickBooks – an accounting package for Small Business Owners (1992)
- Fast rise in sales through the Internet but no specific website or onsite strategy for QuickBooks (5% by 1999)
- Experimentation with Channels on existing Quicken.com site
- Shift to Advertising Model similar to Media Industry to drive a development of a new software industry
- Revision to focus on developing 3rd party relationships to create an extended set of online services for QuickBooks

1998-2002 Extended mental model:
- Branding combined application and online services (2001-2002)
- Multilevel Developer and Service Partner Relationships – At the Intuit Corporate Level
- Emphasis on Strong Interaction and Communications With Key Partners

Decisions-actions:
- Developed accounting package for small business owners; Accepted sales through Internet during late 1990's; Initial online strategy to offer services related to QuickBooks through a small business channel on Quicken.com (1998); Revised online strategy to focus on Value-Added eServices, and online extensions to QuickBooks through browser access; Development of framework for partner evaluation ranging from "light" to "deep" relationships with QuickBooks (1999); Creation of "QB Economy' umbrella branding (2001); Redevelopment of Intuit Developer Network for all Intuit Partners (2001-2002); Launch of "Right for My Business Strategy (2003); Full online version Of QuickBooks released (2003)

Etic 3 view:
- Etic 2 views validated or extended
- Transformation From Internal Product-Focused Decision-Making To Outward Services Focus
- Translation of Vision To Action Using Experiments
- Partnership Management And Involvement In Application Development Cycles
- Managing Partners As On Online Community
- Revised Online Strategy was revised with a more corporate emphasis
- Deeper Segmentation of QuickBooks Applications in 2003
- Finally a true online version of the QuickBooks Application (2003)!

Etic 2 and etic 3 issues highlighted in further levels of analysis for the research project included:

- Initially an experimentation view with online business – for example, collect orders, then try out a portal approach
- Small Business Channel on Quicken.com did not really reach its small business customers who were not regular users of the Internet
- Revised online strategy included a rigorous partner selection framework – partly to accommodate issues with stringent quality controls within Intuit Inc.
- Transformation from internal product-focused decision-making to outward services focus
- Translation of vision to action using experiments
- Partnership management and involvement in application development cycles
- Managing partners as on online community
- Revised online strategy was revised with a more corporate emphasis
- Deeper segmentation of QuickBooks applications in 2003
- Finally a true online version of the QuickBooks Application (2003)!

Overall, the five-level hermeneutic framework for the Intuit QuickBooks case study has provided a strong and comprehensive framework for capturing and extending the dynamic sensemaking for Intuit QuickBooks' decision-makers as they experimented with transitioning the QuickBooks application to an online environment with extended services. Further insights were gained into the transition from an internal single-firm development culture to a third-party oriented application development philosophy.

10.2.6 Advanced Hermeneutic Analysis – Trilogy Case Study

The Trilogy case study is the sixth and final case study in the research project in which the proposed advanced hermeneutic research approach was applied. At least two rounds of emic data and three rounds of etic data were included in five levels of hermeneutic analysis for the study. A summary of the five-level hermeneutic analysis for the case is presented in Exhibit 10.6.

Exhibit 10.6. Level V Analysis of Trilogy Case

Etic 1 view:
•High risk taking to rapidly develop a software house capable of taking very large ERP competitors
•Very strong focus on application development excellence
•Trilogy rated themselves superior to Microsoft for application development
•Their application development approach focused on rapid development in small teams of superprogrammers specially developed by Trilogy to maximise quality and speed of application development
•Trilogy programmers were rewarded highly and encouraged to take risks with defined boundaries

Emic 2 view:
•Validated Emic 1 views
•Transformation to Vertically Industry-Focused Divisions
•Trilogy application tailored by each Division for their key Customers
•Recruitment of Industry specialists and experts into Divisions
•Move to full front-end e-commerce positioning
•Consolidation away from Internet spin-off businesses back to industry focused enterprise customers

Emic 1 view:
•Liemandt identified problems with hardware companies managing their selling and delivery processes
• Sales and Marketing costs typically represented over 40% total computer hardware vendor expenses – and were open to substantial automation
•Development of pricing configurator application for hardware companies
•Expanded to be a complete "front-end" enterprise application
•To be the best application development company in the world
•15 or 20 successful software companies could be built around the talent in Trilogy's development organization
•New philosophies on application development based on small teams of "superprogrammers"
•Deep concern about reaching "critical mass" to compete against large "back-office" ERP providers

Etic 2 view:
•A company with an unusually strong emphasis on application development
•Focus on small team application development
•Not initially focused on Internet-based development but shifted quickly toward e-commerce in the late 1990's
•Focused on "anticipatory" competitive actions

1991-1998 Initial mental model:
•Focus on Automation of Sales & Marketing Activities
•Aim to cut hardware vendor Sales & Marketing Costs by 2 % per year
•Excellence in Application Development through using small teams of "super programmers" managing whole projects with frequent version builds and automated testing
•Trilogy University established to further develop superprogrammers
•Rapid expansion of application to be a complete front-end e-commerce application – before the back-end ERP vendors took that space from Trilogy
1998-2002 Extended mental model:
•Develop and Spin-off "Dotcom" businesses based on trilogy's applications
•Vertical Industry Focus with application development and delivery focused on Specific industry "roadmaps"
Decisions-actions:
•Developed configurator application (1991-1992); Sold to Selected Computer Hardware Vendor plus other manufacturing companies (1992-1994): expanded application to support all elements of Sales & Marketing (1995-1998); Faces Head-On Competition from main ERP Vendors (1997-1999); Rewrote all applications completely for Internet (1997-1998); spun-off several Internet businesses (1999-2000); restructured and consolidated into Vertical industry divisions (2001): Trilogy promotes focus on software productivity and attempts to link to ROI through customer programs (2002-2003); Trilogy releases FCT Methodology as a product/consulting service (2003)

Etic 3 view:
•Etic 2 views validated or extended
•Transformation To Vertical Industry-Focused Divisions
•Trilogy application development philosophies maintained in restructured Divisions
•Trilogy actually turned their application development philosophy into a Product/Methodology!
• Can Trilogy's' philosophy of application development excellence be successfully applied to developing and managing industry-specific applications?
•Trilogy continues to battle with the ERP Vendors
- sometimes in competition, sometimes in partnership
•New focus by Trilogy on ROI from software productivity encourages measurement of effectiveness of using front-end applications

Overall, the five-level hermeneutic framework for the Trilogy case study has provided an effective systematic framework for capturing and extending the dynamic sensemaking for Trilogy's decision-makers. Key etic 1 issues highlighted in the original case-study account (Austin 1998; Mandel & Austin 2000) included:

- High-risk taking to rapidly develop a software house capable of taking on very large ERP competitors
- Very strong focus on application development excellence
- Trilogy rated itself superior to Microsoft for application development
- Its application development approach focused on rapid development in small teams of superprogrammers specially developed by Trilogy to maximise the quality and speed of application development
- Trilogy programmers were rewarded highly and encouraged to take risks with defined boundaries.

Etic 2 and etic 3 issues highlighted in further levels of analysis for the research project included:

- A company with an unusually strong emphasis on application development
- Focus on small team application development
- Not initially focused on Internet-based development but shifted quickly toward e-commerce in the late 1990s
- Focused on "anticipatory" competitive actions
- Transformation to vertical industry-focused divisions
- Trilogy application development philosophies maintained in restructured divisions
- Trilogy actually turned its application development philosophy into a Product/Methodology!
- Can Trilogy's philosophy of application development excellence be successfully applied to developing and managing industry-specific applications?
- Trilogy continues to battle with the ERP Vendors – sometimes in competition, sometimes in partnership
- New focus by Trilogy on ROI from software productivity encourages measurement of the effectiveness of using front-end applications.

Overall, the five-level hermeneutic framework for the Trilogy case study has provided a strong and comprehensive framework for capturing and extending the dynamic sensemaking for Trilogy's decision-makers as they created and grew a company with a deep and strong commitment to application development excellence.

The analysis also highlighted a strong mindset focusing on growing to critical mass in order to compete directly against larger and more well-established software houses. Additional insights were gained into Trilogy's transition from an application development-focused organisation through to a provider of vertical industry-focused e-commerce applications and application development methodologies.

10.2.7 Advanced Hermeneutic Analysis – Summary

This research project has proposed and applied hermeneutical interpretations in B2B contexts to deepen the understanding and description of specific decision-making processes. In particular, the project has addressed the appeals by Langley et al. (1995) to open up decision-making through "re-analysis" of previously analysed decision processes – not just new ones. Multiple-level hermeneutic analysis was effective for systematic allocation and representation or "sense-mapping" of complex decision-making, for all case studies in the research project.

Advanced hermeneutic analysis for every case study constantly challenged the observer(s) to analyse and explore mental models, plus decisions and actions, more deeply over multiple iterations – to cast and recast etic interpretations. The analysis also challenged actual decision-makers to revisit, question, reshape and often to restate their original decision-making – to revisit their emic interpretations.

Advanced hermeneutic interpretation for each study has been greatly enhanced and complemented through the use of DSA, event, and cognitive mapping and collecting additional emic interpretations of original emic reports, as well as emic interpretations of early-on etic interpretations. High research value could be derived from consolidating the cross-case-study highest level hermeneutic analysis into a "meta-level" hermeneutic analysis. However, due to time and space constraints, it was decided that the full development of such a system would be a valuable follow-on project from this project.

10.3 Discussion on Cross-case Study Findings

There were several opportunities for substantial cross-case study analysis for the research project – indeed there was room for several "sub-projects" addressing thorough cross-case study analysis in: commonalities across case study findings; developing a comprehensive cross-case-study extended DSA package containing composite DSA models, events chronology maps, and cognitive maps; and a composite analysis of all extension issues.

However, these sub-projects would have greatly added to the time to complete this project and would have added at least an additional chapter to an already large project report. Nevertheless, it is recognised that all of this additional analysis would represent a very strong set of follow-on activities to strongly round out the whole research project. In order to demonstrate at least one area of cross-case analysis for the project, the author chose to focus on the development of a composite cross-case DSA model.

10.3.1 Decision-making for Internet-based Application Development and Delivery – Development of a Composite DSA Model

Extended DSA was undertaken for all six case studies in the research project. Additional material (which could also be regarded as triangulation data) collected beyond the original case studies and interviews included updated website material for all cases, and in some instances additional source materials, such as books (in particular for Red Hat) and company documents used with permission.

Content analysis was undertaken on two datasets to gain further insights for cross-case discussion:

1. A file containing an aggregation of all case-study chapters in this report (the Red Hat case study was a combination of the specific Red Hat sections in Chapter 1 and the case study in Chapter 4 (processed through TACT and Copernic analytical applications)
2. A file containing an aggregation of extracts from the Harvard Business School Case Studies (including additional written vignettes) used in the development of DSA models, events chronology maps, and cognitive maps (processed through the Copernic Summariser analytical application).

For the aggregated case-study chapter file, TACT word lists were generated and reduced to the top twenty words. A new application for 2004 claiming superior word, phrase, string and linguistics concepts recognition to the TACT application, was selected for cross-case analysis. The Copernic Summariser application generated lists of words rated within the package as "significant" concepts. The list of top comparative words and concepts from both the aggregated case-study and extract files as processed using TACT and the Copernic Summariser is presented in Table 10.1.

Table 10.1 Aggregated Case Study Chapters and HBS Case Study Extracts.

CASE STUDY CHAPTERS COPERNIC SUMMARISER	CASE STUDY CHAPTERS TACT WORD LISTS	HBS CASE STUDY EXTRACTS COPERNIC SUMMARISER
applications	development	business
customers	applications	customers
interviews	new	marketing
marketing	kana	exhibit
managers	study	features
cognitive maps	trilogy	managers
business	software	extract
insights	into	technology
Internet	quickbooks	applications
decision-making	through	interns
maps	Zaplet	engineers
enterprise	product	communications
DSA models	netdynamics	integration
technology	business	cycle
online	intuit	Trilogy
	Linux	
	ongoing	
	company	
	management	
	their	
	red	
	hat	

A new generation of applications for analysing unstructured information, focusing on semantic linguistic linkages and new forms of textual engineering are emerging, that will offer much richer and more reasoned analysis of word and language patterns. Copernic Summariser is one of the new these new applications.

The W3C Organisation is developing the World-Wide Web into a Semantic Web, whereby data throughout the Web will be able to be associated through advanced linguistic matching concepts and processes – way beyond current hypertext and markup language tagging and links. For further discussion on developing the World-Wide Web into a Semantic Web, see W3C Worldwide Consortium (2004).

Leaders in the field of developing new tools and methodologies in this area include the Fraunhofer Institute in Berlin (see Fraunhofer IITB 2004); the University of Sheffield and the General Architecture for Text Engineering – GATE (see Natural Language Processing Research Group at the University of Sheffield 2005); and Copernic with a specific search and summarisation application – Copernic Summariser – based on Semantic Web design (see Copernic 2005).

The lists indicate that the main themes of the case-study chapters are in line with the objectives of the research project, that is, to research the decision-making associated with the development and delivery of Internet-based applications, including online applications and services, using DSA models and cognitive maps – validated through interviews with managers and marketers, to gain their insights into these processes (italicised words were in the concepts list for the aggregated case-study chapter).

A composite DSA model was created (see Figure 10.2). Findings associated with DSA and with extension questions and case summary points were consolidated into a composite DSA model and cross-case tables. Discussion on DSA models, extension questions and cross-case summary is outlined in the following sections.

DSA models and discussion were reviewed from all case studies and a composite DSA model was created (see Figure 10.2). A basic application development cycle incorporating setting application development objectives (box 2); developing application design plans (box 10); application development (box 11); beta testing and QA (box 12); training, pre-launch packaging (box 13); full release (box 14); and ongoing support (box 18) is common to all of the software houses in the research project.

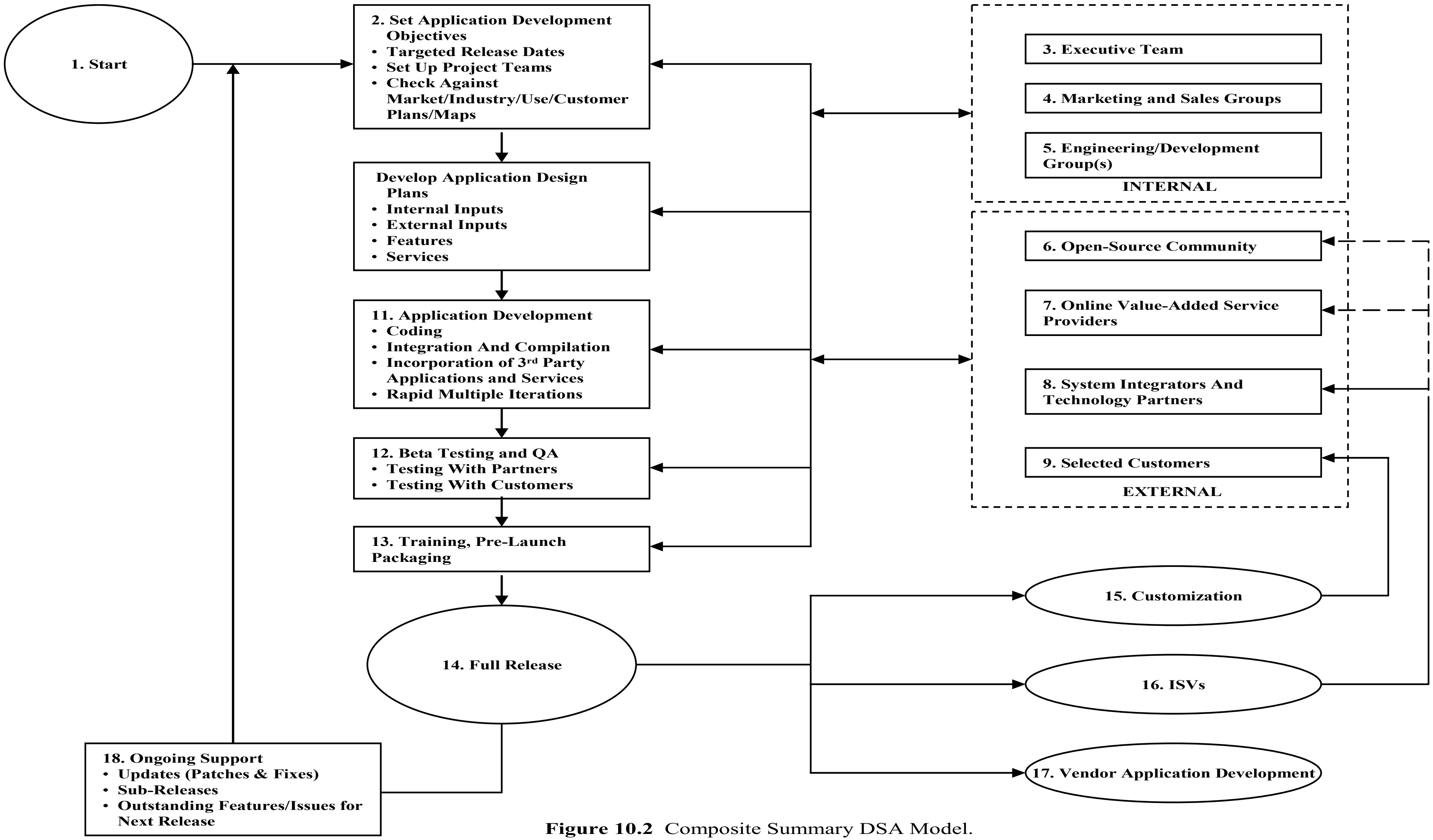


Figure 10.2 Composite Summary DSA Model.

Internal (boxes 3, 4 and 5) and external (boxes 6, 7, 8 and 9) inputs to decisions at all application development phases are also apparent for all software houses in the project.

However, significant variations occur for each software house in the project. Actual development cycles varied from four months (early versions of Red Hat's Linux) through to rigid 12-month cycles (Intuit Quickbooks) and around two years for larger enterprise e-commerce applications (Zaplet, Trilogy and later versions of Red Hat's Linux).

There were variations in the degrees of internal and external inputs to application development. Trilogy and Intuit QuickBooks were the most internally focused software houses, with very strong software development philosophies and processes. Zaplet was also strongly focused on internal software development; although it had to quickly develop internal application development capability through bringing in groups of programmers from Sun Microsystems and other places with Java programming capability.

Kana started with a focus on building up internal developer resources, but aggressively pursued mergers and acquisitions to extend its application features and developer capabilities.

NetDynamics built developer capability but was involved in a multi-way development of Web-based infrastructure with Sun Microsystems and Netscape Communication. As such NetDynamics took in – and gave back – a lot of inputs from these third-party developers.

Red Hat's symbiotic relationship with the open-source community meant that it was both a part of a larger third-party developer community and was developing applications for that community. Red Hat's application development cycle appears to be the most open to third-party inputs.

However, all software houses shifted their decision-making input sources over time. All software houses incorporated increased inputs from more groups within their own companies, from targeted companies, and from third-party application and service providers.

Intuit QuickBooks transformed from being an internally focused business-to-consumer (B2C) software house to incorporating third-party application and service provider inputs throughout its whole application development cycle, to offer a comprehensive online environment for their applications. The other software houses increased third-party application provider and large customer inputs.

The composite DSA model deliberately delineates between a full release of an application and further internal and external development of it. Several applications in the research projects were “application technologies” or application “platforms” set up for development into higher level applications and uses. A portion of this development would be to customise for specific customers or industries or activities. Such customisation might be done by the vendor directly with the customer or system integrators/technology partners working with customers.

Another portion of development may be for the application to be incorporated into an ISV’s application for use in specific markets and activities. The other portion of application development is for the vendor to create its own applications through further development. Trilogy’s industry divisions took Trilogy’s basic applications and customised them into specific industry-based applications. Zaplet attempted to take the basic AppMail application and develop further applications for particular activities and uses.

Moving toward both Internet-based application development and delivery appears to have prompted all software houses in the project to open up their decision-making processes much more to third-party inputs right through their application development cycles.

10.3.2 Decision-making for Internet-based Application Development and Delivery – Further Insights through Cognitive Mapping

The DSA modelling on its own for each case study has provided decision-making highlights for actual application development cycles. However, the DSA has been augmented with an events chronology map to capture the application and software house development and evolution over time.

Both DSA models and event history maps have been reviewed, and extracts with selected issues were drawn from the Harvard Business School case studies, further analysed through the TACT and NVivo content analysis applications, with cognitive maps created using the NVivo Model Explorer application.

Compared to other applications used for case-study data analysis and mapping, NVivo is strictly driven by direct researcher judgement. It does not use algorithms or calculations to attempt to automatically connect coded concepts or words into a map.

Moreover, NVivo allows the researcher to quickly create maps dynamically and iteratively, giving the researcher control to determine link directions and types of associations for links between words or concepts.

Cognitive maps produced from NVivo were clear and holistic representations of quite detailed insights associated with decision-making on a range of selected issues, events and timeframes. These maps were consistent with the cognitive map formats of Nath & Newell (1998). However, for most extracts, there was a lot of detail to map and a translation of formatting with more structure and nesting was completed on transferring maps from NVivo to PowerPoint.

The maps were effective "reconstructions" of sensemaking used with interviewees to retrospectively review, validate or review their sensemaking over time. Almost all interviewees complimented the detail, flow and accuracy of the original cognitive maps. Some interviewees insisted that sensemaking timeframes be extended to incorporate additional and new insights to provide a more complete (and dynamic) overall sensemaking view of their company, their applications and themselves.

The final maps for each case all contained significant updated insights, nuances, and flows of concepts, highlighting that the multiple interview process and collection of triangulation data was not just a validation effort, but a means to collect much more valuable insights and to produce far richer cognitive maps than the original representations. The main points from the cognitive maps were captured in case-study summaries, and some will be discussed further in the section on summary points later in this section.

Development of a full cross-case analysis including composite events chronology maps and cognitive maps, while representing a very strong contribution to the report, was regarded as beyond the cope of this report and an extension project.

10.3.3 Decision-making for Internet-based Application Development and Delivery – Further Insights from Extension Questions

Although extended DSA (DSA models plus an events chronology map and sets of cognitive maps) have provided deep insights into decision-making issues, the extension questions rounded out very important issues associated with decision-making – and especially from third-party involvement and marketing perspectives. Main cross-case points relating to extension questions are discussed question-by-question, with the exception of dominant logic and shared vision, which are addressed under one discussion section.

10.3.3.1 Dominant Logic and Shared Vision

Key points from extension questions on dominant logic (see Prahalad & Bettis 1986) and shared vision (see Dougherty 1992) for each case study were summarised in a cross-case table, presented in Table 10.2.

In all case studies the founders of the software houses exercised dominant logic in creating the vision and direct development of applications. Five out of the six software houses in the research project were created after 1991, with founders simultaneously developing visions for commercialising ideas, starting up companies and creating application development and delivery capabilities.

Intuit Inc. was created in 1982, but the QuickBooks product line, which is the focus of the case study, was created in 1992. Nevertheless, Intuit's founder provided strong dominant logic for the ongoing development of QuickBooks.

Dominant logic varied over time for most of the case studies in the project. Executive teams and senior application development executives either influenced or contributed to dominant logic in most case studies (Red Hat, NetDynamics, Intuit QuickBooks (particularly later in the case timeframe) and Trilogy). Changes in CEOs and Executive Teams prompted changes in dominant logic. For Zaplet and Kana, major changes in vision and direction emerged from new CEOs. For Trilogy, an organisational restructure saw dominant logic partially devolve from the foundation to industry division general managers.

There were other significant contributors to dominant logic identified in most case studies – most were senior technical executives. There were also two cases with significant contributions from third parties. Zaplet's venture capitalist provided strong contributions to assist Zaplet's visualisation of new application ideas and application development resources.

The Red Hat case study provided the most unusual contribution of dominant logic – collectively from the open-source community. Further analysis of the open-source community suggests that those members of the community who coordinate and review ongoing integration of open-source applications are "dominant logics" within the community.

For all case studies, founders claimed that vision (emerging from their dominant logic) was strongly shared throughout their companies. There were significant variations in sharing vision with third parties.

Table 10. 2 Cross-Case Dominant Logic and Shared Vision.

	Red Hat	Zaplet	NetDynamics	Kana	Intuit QuickBooks	Trilogy	Cross-case Commonalities
DOMINANT LOGIC:	**Vice presidents for development (throughout case timeframe). Open-source community – coordinators and reviewers who consolidate and integrate applications**	**Pre-1999 – founders. 1999-2000 – founders plus venture capital partner. 2000-01 – new CEO.**	**CEO and chief engineer.**	**1996-199 – founders 1999-2000 – new CEO**	**Intuit Corporation founder; Developers of QuickBooks online strategy. 2001-02 Intuit product development group.**	**1991-present – founder. 1992-present – chief software developer. Mid-1990s – creator of Trilogy University; developer of Trilogy's software developer philosophies. 2001-02 – founder plus industry division managers.**	**Founders; New CEOs and Executives Vice-Presidents of development**
SHARED VISION:	**Shared throughout open-source community; symbiotic linkages between red hat and open-source community.**	**Shared across company. Moving toward increased sharing across ISVs and partners in 2001.**	**Engineering and marketing vision shared across the company. Shared vision with selected mutual ISV partners – stronger with Sun Microsystems for java plus some vision with Netscape. Strong shared vision with customers for application development, testing and release.**	**Founders and executive team. Opportunistic but strong with targeted, ISVs, system integrators, and business partners. Shared vision with targeted enterprise customers.**	**1992-98 - product management, sales and marketing, development groups. 1998-2002 – Intuit product development group and executive team with Intuit developer network (key ISVs and service providers).**	**1991-97 – strong collective shared vision throughout. 1998-2000 – shared vision within company plus within Dot.Com spin-offs. 2001-02 – collective shared vision translated into industry divisions; shared vision with selected industry enterprise customers and partners .**	**Shared across company; Shared with ISVs, selected business partners and customers**

There was minimal sharing of vision with third parties (except perhaps customers) in the early timeframes of three cases (Trilogy, Intuit QuickBooks, and Zaplet).

For the other three cases, shared vision with third parties ranged from strong sharing of vision with customers and key ISV/Suppliers (NetDynamics), to active sharing and development with technology vendors, system integrators and customers (Kana), through to a strongly symbiotic creation and sharing of vision between the software house and a broad community of developers and users (Red Hat).

From 2000 onwards, Trilogy and Zaplet shifted toward an increased sharing of vision with third parties. Trilogy had developed a broader shared vision through its dot.com spin-offs, but then focused on sharing vision with targeted enterprise customers and business partners that fit with its vertical industry mapping and divisional structure. Zaplet moved toward sharing vision with its enterprise customers, ISVs and business partners from 2001 onwards.

Intuit QuickBooks transformed from a strict internal sharing of vision prior to 1998 toward strong sharing of vision with ISVs and services providers – particularly for the collective development of the online environment for QuickBooks, enriched by further applications and services. The development and use of Internet-based computer-mediated environments (CMEs) appears to have been an important factor for shifts in the shared visions in all case studies in the project.

10.3.3.2 Leverage Points

Leverage points from all case studies were consolidated in the cross-case table presented in Table 10.3 (for more discussion on leverage points see Klein 1999). Each case study contains distinct sets of leverage points, but there were some distinct common cross-case points in the table.

For most of the founders of the software houses in the case studies, there was a point where a business or particular problem was identified and articulated into an idea. In some cases there was a social issue (for example, Zaplet), in other cases there was a particular business issue (for example, managing business email (Kana); reducing sales and marketing costs (Trilogy); accessing business information from databases via the Web (NetDynamics); and how to move a successful software application to an online environment (QuickBooks)).

Table 10.3 Cross-Case Leverage Points.

Red Hat	**Zaplet**	**NetDynamics**	**Kana**	**Intuit QuickBooks**	**Trilogy**	**Cross-case Commonalities**
GNU Project initiated (1983). Linus Torvalds creates and distributes Linux Kernel Program (August 1991). Torvalds signs GNU Public License for Linux (March 1992). Marc Ewing packages Linux with installation and version management applications (October 1994). Red Hat Inc formed (January 1995). Red Hat IPO (August 1999).	**Development of Zaplet technology prototypes (1998-mid-1999). Roberts' and Axe's presentation to KPCB (23 July 1999). Recruitment of key managers and developers (2nd Half 1999). Zaplet building blocks could be used to create several applications (Late 1999), Public launch of Zaplet.com (13 March 2000). Appointment of Alan Baratz to Zaplet Board (Early 2000). Enterprise market segmentation (September 2000). First major Zaplet restructuring (March 2001).**	**Decision to extend development beyond Web pages to server applications (late 1995). Selection of Java as the scripting language (November-December 1995). First Integration (10 March 1996). Testing and feedback from the beta customers (April-May 1996). Acquisition of NetDynamics by Sun Microsystems (July 1998).**	**Emergence of idea to handle Incoming e-Mails (1996). Early rounds of funding (1997) Kana Communication's official launch as a company (March 1998). Emergence of need for new e-commerce applications – such as online customer communication management (1998). Kana's IPO and release of Kana 4 (September-October 1999). Kana's Merger with Silknet (February 2000). Kana's merger with Broadbase (Mid- 2001)**	**Creation of QuickBooks application for small-business owners (1992). Development of Portal Strategy (Spring 1998). Reorganization into business groups (1999). Revised online strategy based on partnerships (Late 1999-Summer 2000). Creation of "QB Economy" concept (2001).**	**Moving to Austin (1991-92). Officially moving to "selling-chain" approach (Spring 1996). Converting development to internet applications (1997). Internet business spin-off strategy (Late 1999-2000). Rebranding to e-commerce applications (2000). Transition to vertical industry focus (2001).**	**Articulation of idea. Selection of technology or language for development. Revision of original strategies**

A second common leverage point was the selection of particular technologies or languages. Sun Microsystems' development of Java appears to be an important leverage point for Zaplet, NetDynamics and Trilogy (for when they converted their applications to the Internet).

There appears to be a collective leverage point (and possibly a "tipping point") where intense interest in gaining and using Java programming skills among programmers and developers in the mid-to-late 1990s greatly influenced language selection for several application projects across a very large number of universities, incubators and new software houses.

The NetDynamics application (which became part of Sun's iPlanet application suite) is a tipping point for the creation of e-commerce applications – and may have relevance for the emergence of e-commerce application suites from Kana, Trilogy and possibly Zaplet. Creation of the World-Wide Web CME could be regarded as a leverage point for Red Hat and Intuit QuickBooks, for the collective development and delivery of online application and services.

A final common leverage point is that all software houses in the project significantly changed their strategies – and particularly in the face of the technology stock corrections of 2000-01. Some companies refocused on large enterprise customers (for example, Zaplet, NetDynamics, Kana); specific vertical enterprise market segments (for example, Trilogy) and new partners and communities (for example, Red Hat, QuickBooks).

10.3.3.3 Strategic Marketing Issues

Strategic marketing issues from all case studies were consolidated into a cross-case table presented in Table 10.4.

Several marketing issues are common across the case studies. Application diffusion, adoption and mass acceptance was a major marketing issue identified in three case studies (Red Hat, Zaplet and NetDynamics).

Shifting from a horizontal to a vertical industry focus was an important strategic marketing issue in three cases (Zaplet, Kana, and Trilogy).

Table 10.4 Cross-Case Strategic Marketing Issues.

Red Hat	Zaplet	NetDynamics	Kana	Intuit QuickBooks	Trilogy	Cross-Case Commonalities
Gaining mass-acceptance for software applications. A new concept of "Open-source" marketing. Using the Internet for concurrent application development.	**Understanding user requirements specifications. Consolidation to a specific market focus while creating new "killer" applications. Product managers and product marketers seen as marketers. Development of customer networks or "ecosystems". Application adoption issues. "Viral" diffusion of applications.**	**Rapid application diffusion and "first-mover" advantage. Pricing evolution from client to server/enterprise applications. New channels for enterprise applications – especially system integrators. Changes in structure (and ownership) to support rapidly growing enterprise customer base.**	**Working with "lead-users" on learning, development and implementation of new applications. Defining potential development and use of new and emerging applications. Transformation and repositioning from small software house to a major software provider. Emphasis on quality, specialization and solving specific online issues. Building applications versus acquiring applications and skills. Vertical versus horizontal marketing issues Roles of system integrators (SIs) and Independent Software Vendors (ISVs).**	**Developing value beyond the basic product or software application – in-house or through partners. Cross-selling products through all of Intuit's channels. Using the Internet to communicate marketing and alliance programs.**	**Vertical industry-focussed marketing. Broadening product lines/application suites. Product management and product marketing roles.**	**Shifting from a horizontal to a vertical market focus. Managing growing enterprise customer bases. Roles of business and channel partners, for examples ISVs, and technology/software partners. Internal versus external participation in application development.**

Managing enterprise application development and customers was a significant strategic marketing issue for three cases. Senior executives from NetDynamics, Kana and Trilogy saw substantial marketing challenges associated with expanding applications quickly from a small set of programs into an enterprise strength application suite – plus providing ongoing support to enterprise customers.

The assistance of Systems Integrators (SIs) in customising applications and providing marketing resources to service enterprise customers was an important strategic marketing issue for NetDynamics and Kana.

Issues associated with using both product managers and product marketers as marketers were considered as significant for Zaplet and Trilogy.

Using the Internet for application development, delivery and channel management were also marketing issues for Red Hat and Intuit QuickBooks.

Significant strategic marketing issues for single software houses in the project included a concept of "open-source" marketing (Red Hat); developing customer networks or "ecosystems" (Zaplet); working with "lead-users" (Kana); building applications through acquisition and in-house development (Kana); augmenting applications with services for online environments (Intuit QuickBooks); and cross-selling products through a range of channels (Intuit QuickBooks).

10.3.4 Summary Points

Case summaries from all cases were incorporated into one cross-case table and are presented in Table 10.5. While there were several distinct case-study insights, a number of cross-case commonalities were discerned from the case summaries. The distinct case-study points are noted first, followed by cross-case commonalities.

The Red Hat case study contributed new and extended insights into one of the most revolutionary movements in application development, that is, "open-source" application development. The concept is extended to marketing and will advance further into product development and societal information creation and sharing.

The Red Hat case study also highlights the substantial capabilities of using the Internet (the World-Wide Web) as an environment for creating and delivering applications. As the Red Hat case study was the first undertaken in the project, there were various vital revisions to data collection and presentation that were applied to all the other case studies.

Table 10.5 Cross-Case Summary Points.

Red Hat	Zaplet	NetDynamics	Kana	Intuit QuickBooks	Trilogy	Cross-Case Commonalities
Collective decision-making associated with "open-source" application development. Identification and articulation of an "Open-source" marketing concept. Decision-making and application development using the internet as a concurrent application development and delivery environment. Revisions for subsequent project case studies.	Commercialization of an application in conjunction with creating a new business. Role of venture capitalists in providing leadership in development of ideas for potential applications and providing application development resources. Decision-making associated with transition from broad idea generation to highly focused enterprise development in providing application development resources. Searching for an integrative application to support an enterprise "ecosystem".	Diffusion of key application technologies for emerging computer-mediated environment. Multi-dimensional ISV concurrent application development. Systems integrators as extended or virtual application marketers. Startup business management of large customer installed bases over time.	Decision-making associated with development and rapid releases of new Internet-based applications. Decision-making associated with extending broadening application capabilities and suites. Business growth and application development through rapid and strategic acquisitions (and mergers). Channels of distribution and third-party partnerships for Internet-based software houses.	Transformation from internal product-focused decision-making to outward services focus. Translation of vision to action using experiments. Partnership management and involvement in application development cycles. Managing partners as an online community.	Transformation from product-focused to vertical industry-focused decision-making. Underlying software development philosophies and their contribution to decision-making. Decision-making associated with anticipatory competitive actions. Defining and redefining product lines/application suites.	Rapid creation and delivery of new software applications – at the same time as growing a new business. Diffusion of Internet-enabling technologies. Diffusion of the Internet as a development and distribution platform for applications, products and services. Innovation through direct partner involvement in software application development

The Zaplet case study contributes insights into combining the development of new applications with starting up a new business. It highlights the venture capitalist role in facilitating idea generation and application development resources. There were insights into transitioning over time into targeted enterprise application development. The case study also provided insights into searching for applications that would support enterprise "ecosystems" – powerful networks of customers and partners.

The NetDynamics case study highlighted issues associated with the collective development of applications for an emerging computer-mediated environment (CME). There were very high risks associated with developing a revolutionary application component for an emerging Web browser technology – but large payoffs if the development could be achieved within a six-month timeframe.

All three main software developers, NetDynamics, Sun Microsystems, and Netscape Communications, had developed very early pre-release versions of their piece of a larger new Web browser technology puzzle. They were effectively ISVs and beta testers for each other. The case highlights issues for multi-dimensional and concurrent ISV application development.

The NetDynamics case study also highlights the use of System Integrators (SIs) as extended marketing resources for small software houses to large enterprise customers. The case also highlights the challenges for small software houses, associated with the ongoing management of rapidly growing enterprise customer-installed bases, over time.

The Kana case study highlights the decision-making associated with the rapid growth and expansion of both a start-up software house and a simple e-mail communications management application into a full enterprise e-CRM suite within a three-year timeframe.

The Kana case study highlights strategic acquisitions (and toward the end of the case timeframe, large mergers), for the purpose of broadening application capabilities and development resources. There is also a focus on using third-party partnerships to extend marketing and developer resources – especially for System Integrators and Technology partners.

The Intuit QuickBooks case study contributes insights for a well-established consumer-oriented software house's efforts in transitioning toward a comprehensive online application plus services development philosophy. There is an emphasis on experimentation, with at least three iterations of online strategy from 1998 to 2003.

The QuickBooks case represents the greatest transition of all in the research project, with Intuit Inc. transforming from very strict and defined internally focused application and delivery processes, to the involvement of third-party partners at all stages of development, complemented by a strong online Intuit Developer Network.

The Trilogy case study contributes valuable insights into an organisation with very strong application development philosophies and culture. Trilogy puts very high value on "Superprogrammers" and possessing "best-in-world" application development capability as a foundation company philosophy.

The Trilogy case study provides insights into the minds of executives focused on rapidly growing a company to be at a critical mass to anticipate competitive actions against very large and powerful ERP software houses.

Further insights are provided into Trilogy's expansion from a simple computer hardware pricing configuration program to a comprehensive enterprise e-commerce pricing and front-end application suite.

The Trilogy case study also highlights a transformation from a horizontally focused application development focus to a vertical selected industry-focus for application development and organisational structure. Trilogy managed to shift to targeting selected industries while preserving its strong application development values. In a quite recent development, Trilogy now sells its application development system as a service!

There are three broad cross-case summary points derived from the case-study summaries.

The first cross-case summary point is that all software houses were tackling the concurrent challenges of rapidly developing new software applications (although, in the case of QuickBooks, a new business division), while at the same time growing a new business.

Most of the software houses were racing against real or perceived competitors to gain a critical mass such that they could service large enterprise customer bases, and compete against larger more established software vendors. Indeed, most of the larger vendors noted in the research project had not transitioned to Internet-based applications, or online-based environments, during the original timeframes of the six case studies in the project.

The second cross-case summary point is that diffusion of Internet-enabling technologies, particularly Java, but also moving applications or channels to the Internet, was a strong cross-case issue – which was not so surprising as the software houses chosen for the project were specifically focused on the development and delivery of Internet-based applications, just at the time when the Internet was taking off as a environment for business use.

The third cross-case summary point, and it is a strong point, focuses on the involvement of partners in software application development – which by, Krishnan and Ulrich's (2001) definition of production definition, places the rapid and flexible development of software applications for and using the Internet, as more linked to innovation and entrepreneurship rather than conventional product development.

Everyone is involved in application development in the project case studies, ranging from full collective community development approach (for example, Red Hat) through to ISVs and customers (in all the other case studies). There was provision in most of the case studies for business partners to act as extended sales, marketing and business support resources.

It cannot be doubted that developing new applications, services and "marketspaces" in "Internet time" required tremendous managerial ability and agility to work literally in real-time with third parties, including SIs, ISVs, ISDs, customers and other channels partners.

Each case study in the research project has contributed distinct and valuable insights, plus a number of cross-case insights have been derived, so far without much directed theory. However, there are also contributions that can be assessed and incorporated in theory generalisation and development.

10.4 Theory Generalisation and Development – Structuration Propositions

Theory development for the research project is based on analysing cross-case findings against a revised and reduced set of Huff, Huff, Barr & Stimpert's (2000) Structuration propositions, with commentary relating to a software industry context.

There may be room to further subdivide the software industry into a special sub-industry for the development and delivery of Internet-based software applications – and indeed, at the outset of this research project, when the Dot.com era was at its peak, such a categorisation could well have been reasonable.

However, through 2002-04 most of the software industry transitioned to either having Internet-based application development and/or using the Internet (and in particular, the World-Wide Web) as an online development and distribution environment.

Five of the six case studies in the research have remained as successful businesses within the software industry. From this perspective, the case studies in the research project could be regarded as pioneers and precursors of disruptive changes in software – and ultimately information service – industry Structuration.

Five of the six software houses have been successful and remain in business in 2005. Three software houses (Trilogy, Red Hat and Intuit QuickBooks) have grown to become relatively large and successful software vendors. Two software houses have been successful but remain as relatively smaller but strongly vertically focused vendors (Zaplet and Kana Communications).

One software house, NetDynamics, was acquired by Sun Microsystems – however, that business division within Sun Microsystems, has become a central component of Sun's Web Services offerings – which may yet be the greatest contributor to the next generation of both the World-Wide Web and new forms of information management seen so far in the short life of the Internet as a business environment.

NetDynamics' descendants may well become the greatest contributor to changes in industry Structuration out of all the cases – although it is the author's own view that NetDynamics could be trumped in this contribution by Red Hat Inc. and the "Open-source" community.

Reverting to the analysis of Structuration propositions, the revised and reduced set of propositions is presented in Table 10.6. New propositions will be discussed as part of theory development.

10.4.1 Proposition P_1: As knowledge actors within the industry draw on past success to respond to new conditions, they draw primarily on familiar rules and resources. Because firms within the industry have knowledge gained from past experience in the industry, even when they are entrepreneurial, knowledgeable incumbents tend to reconstitute industry structures that are quite similar to previously existing industry structures.

Five out of the six case studies in the report project are software houses that have been created since 1991. However, the executives interviewed in the project would be regarded as knowledge actors – although this perspective is more relevant in an individual or group or firm context than for an industry context.

Table 10.6 An Anchored Cognitive Theory for Strategic Change – Structuration Propositions Selected for the Research Project.

Proposition P_1:	As knowledge actors within the industry draw on past success to respond to new conditions, they draw primarily on familiar rules and resources. Because firms within the industry have knowledge gained from past experience in the industry, even when they are entrepreneurial, knowledgeable incumbents tend to reconstitute industry structures that are quite similar to previously existing industry structures.
Proposition P_2:	The motivation for entry increases as outsiders perceive opportunities in actual or contemplated activities that are ignored or perceived as trivial by industry insiders.
Proposition P_3:	New entrants respond to system opportunity in novel ways by drawing on rules and resources that are unfamiliar to most industry participants.
Proposition P_4:	New activities by entrants, amplified by alliances with mavericks within the industry and/or actors not previously involved in the industry, modify the underlying structure (rules and available resources) of the industry.
Proposition P_5:	Active monitoring and interpretation by entrants tends to quickly increase their relative knowledgeability and power. Strong niche performance motivates successful new entrants to expand their activities and invites further entry by imitators.
Proposition P_6:	Efforts to institutionalise their success begin to structure entrant activities, reduce their capacity to respond to subsequent changes in the industry, and thereby increase the opportunities for new entrepreneurial activity by others.

The propositions reproduced above were extracted from Huff, Huff, Barr & Simpert (2000, pp. 212-9).

Founders and executives drew upon knowledge actors such as software developer experts (for example, Trilogy), venture capitalists with resource, application, funding and business development expertise (for example, Zaplet, NetDynamics, and Kana). Kana and Zaplet brought in knowledge actors in the form of new CEOs to restructure and change the strategic direction of their companies.

Trilogy and Intuit QuickBooks set out to reconstitute application development processes that were similar to existing industry structures, but actually had to modify them in response to changing to Internet-based applications and application development. All software houses attempted to develop third-party programs that were initially similar to existing programs from established software houses.

All software houses in the project indicated preferences to set up marketing and sales structures and capabilities similar to established software houses – although most had not done so by the end of 2002.

In the Trilogy case study, existing large ERP software houses extended their existing capabilities through from back-office applications through to end-to-end enterprise application suites. If the enterprise segment of the software industry could be defined as an "industry" in itself, then the ERP expansion noted in the Trilogy case study would lend some support to the proposition.

10.4.2 Proposition P_2: The motivation for entry increases as outsiders perceive opportunities in actual or contemplated activities that are ignored or perceived as trivial by industry insiders.

All case studies in the research project represent the development of applications that most established industry insiders totally ignored at the time of their creation and further development.

Substantial chunks of the established software industry almost by definition ignored opportunities to develop new applications and technologies for the Internet until very late into the 1990s and even into the early years of the current decade.

Trilogy tapped into a massive opportunity to reduce enterprise sales and marketing that no other industry insider software could in the early-mid-1990s.

Red Hat grew on the back of a new operating system that could provide Unix-like functionality on PCs – Linux was literally ignored in the early-mid-1990s by industry insiders, and Red Hat took an obvious – but largely ignored – step to package Linux on CDs with installation and version management software.

Zaplet provided greatly increased functionality to e-mails – an opportunity largely ignored by other software houses.

Kana also provided e-mail management capability and then full e-CRM capability – gaining an early advantage as other software houses ignored that opportunity.

Intuit Inc. was already an established industry player, taking advantage of an ignored opportunity to provide personal and small business financial services. For the online phase of QuickBooks' development, to some extent Intuit QuickBooks noticed rather than ignored the opportunity, and capitalised on it.

Overall, there was high motivation for taking up new opportunities ignored by existing industry players. However, it should also be noted that all the founders and developers taking up new opportunities in the case studies were not new to the industry.

10.4.3 Proposition P_3: New entrants respond to system opportunity in novel ways by drawing on rules and resources that are unfamiliar to most industry participants.

Every case study in the research project responded to opportunities with some special response in terms of drawing on rules, convention, and resources in novel ways.

Red Hat was symbiotically involved in the open-source community that used rules (Copyleft and different licensing) and collective development process that were opposite to conventional industry rules and resource deployment philosophies.

NetDynamics challenged conventional waterfall system lifecycle application development though daily builds of the application during development. They also challenged conventional development rules by using beta versions of other software houses' application to create their own new application, in an unusual multidimensional form of collective ISV application development.

Trilogy took conventional rules and resource deployment philosophies to a deeper and different level though creating "Superprogrammers", very small development teams, and a Trilogy University to train "Superprogrammers". In 2003 Trilogy turned its unusual development philosophies into a consulting service.

Zaplet used some novel approaches to develop ideas for potential uses of its AppMail e-mail application with assistance from its venture capitalist. Its approach to creating a development team and to application prototype development could also be regarded as novel in terms of industry development philosophies.

Intuit QuickBooks moved from quite conventional industry rules and resource philosophies to application development philosophy based on conceptualising QuickBooks as an online application with substantial third-party application and services components.

There are indications from all case studies of software houses challenging conventional industry rules and resource philosophies.

10.4.4 Proposition P_4: New activities by entrants, amplified by alliances with mavericks within the industry and/or actors not previously involved in the industry, modify the underlying structure (rules and available resources) of the industry.

The proposition is best assessed in terms of the Internet, challenging the underlying industry structure, rather than one particular firm. However, if Christensen's (2003) criterion for disruptive technology is assumed to change the underlying industry structure, then all case studies support the proposition. See Table 10.7 for disruptive technology assessments.

Four case studies used alliances to create and build on their initial opportunities. Red Hat's rapid application development and association with the open-source community represents probably the most maverick and revolutionary approach – with the ongoing potential to turn upside down, not just the software-related industry, but all knowledge-based industries and several aspects of society.

NetDynamics formed a maverick set of alliances to collectively create new applications to revolutionise Web browser technology.

Kana aggressively sought acquisitions (and mergers) to grow, plus strong partnerships with System Integrators and Technology Partners to extend its sales and marketing capabilities. While such an approach to growth would not be necessarily regarded as maverick for the software industry, the speed and intensity of execution was quite unusual.

Intuit QuickBooks stepped away from the conventional software industry structure to create comprehensive third-party relationships to produce an online environment with enhanced applications and services.

Both Zaplet and Trilogy worked with third parties, particularly toward the end of each case study – but neither company's third-party programs would be regarded as maverick in relation to the software industry's structure.

10.4.5 Proposition P_5: Active monitoring and interpretation by entrants tends to quickly increase their relative knowledgeability and power. Strong niche performance motivates successful new entrants to expand their activities and invites further entry by imitators.

The proposition was not really addressed in the case studies in the project, except for Trilogy. In the Trilogy case study there was considerable focus on Trilogy's building up a critical mass in anticipation of aggressive competition from existing ERP software houses and other relatively new front-office software vendors.

Table 10.7 Disruptive Technologies.

Software House	Disruptive Technology?
Red Hat Inc.	Yes, Linux, and now open-source software development
Zaplet Inc.	Yes, direct enhancements to email
NetDynamics Inc.	No, linked to Java, Web Browsers and ultimately Web Services (both now likely to be considered as disruptive technologies
Kana Communications Inc.	Yes, related to email and online communication management
Intuit QuickBooks Inc.	Yes, Quickbooks
Trilogy Inc.	No, application cut sales & marketing costs and revolutionised quoting systems which would now be regarded as a disruptive technology

Assessments based on Christensen (2003, 56-65).

10.4.6 Proposition P_6: Efforts to institutionalise their success begin to structure entrant activities, reduce their capacity to respond to subsequent changes in the industry, and thereby increase the opportunities for new entrepreneurial activity by others.

Four out of the six case studies (Trilogy, Zaplet, Kana, and NetDynamics – by then part of Sun's iPlanet Group) were forced to change their operations and strategic focus due to the technology stock corrections in 2000-01. It is too early to tell if these changes have blunted the ability of these companies to respond to (or lead) subsequent industry changes.

The Red Hat case study throws up an interesting question to the proposition – how would "institutionalisation" be defined for open-source application development?

10.5 Theory Generalisation and Development – Revised and New Propositions

A new list of propositions includes twenty propositions in five categories: Structuration; Computer-mediated environments (CMEs); third parties; Sensemaking; Open-source. Table 10.8 presents the full list of propositions.

10.5.1 Structuration

The first four Structuration propositions (**Propositions P_1-P_4**) are unchanged from the list in Section 4.4, although **Proposition P_1** should be viewed from an individual or firm perspective rather than an industry perspective.

Propositions P_5 and P_6 are restructured from the original list. **Proposition P_5** reflects an entrepreneurial sensemaking perspective to fit more in line with research start-up and emerging software houses.

Proposition P_6 recognises that software houses have been forced to change mainly because of the economic and financial consequences of the technology stock corrections of 2000-01.

10.5.2 Computer-mediated Environments (CMEs)

The research project was about analysing the decision-making associated with the development and delivery of Internet-based software. Almost by definition, an exploration of computer-mediated environments (CMEs) is required.

The World-Wide Web is an exemplar CME, so propositions regarding application development and delivery in and with CMEs are a logical progression.

Proposition P_7 states that CMEs enable new forms of concurrent business-to-business application development. The proposition also accommodates a new classification of "business-to-developer" relationships. The World-Wide Web has enabled concurrent application development – within organisations but also across organisations.

Proposition P_8 states that CMEs provide a platform for the rendering of an extended concept of applications and services. Developing an online presence is not just offering an application for downloading through a website.

A range of applications, features, services and community options to be offered in conjunction with third parties is required for effective online (or CME) business success.

Table 10.8 Research Project Propositions.

Structuration:

Proposition P_1: As knowledge actors within the industry draw on past success to respond to new conditions, they draw primarily on familiar rules and resources. Because firms with the industry are knowledge gained from past experience in the industry, even when they are entrepreneurial, knowledgeable incumbents tend to reconstitute industry structures that are quite similar to previously existing industry structures.

Proposition P_2: The motivation for entry increases as outsiders perceive opportunities in actual or contemplated activities that are ignored or perceived as trivial by industry insiders.

Proposition P_3: New entrants respond to system opportunity in novel ways by drawing on rules and resources that are unfamiliar to most industry participants.

Proposition P_4: New activities by entrants, amplified by alliances with mavericks within the industry and/or actors not previously involved in the industry, modify the underlying structure (rules and available resources) of the industry.

Proposition P_5: Shared schema-based interpretations in start-up software houses reflect entrepreneurial sensemaking attributes where there is a constant perception of competition emerging from both incumbent and new industry players.

Proposition P_6: New and emerging firms will withdraw or restructure to apparently "safer" or "stronger" stronger" strategies in the face of financial change that reconstitutes industry structure such that there is direct financial impact on their business.

Computer-mediated Environments (CMEs):

Proposition P_7: Computer-mediated environments (CMEs) enable new forms of concurrent business-to-business (and business-to-developer) application development

Proposition P_8: Computer-mediated environments (CMEs) provide a platform for rendering of extended online applications and services

Proposition P_9: Computer-mediated environments (CMEs) prompt software developers to revise their sensemaking toward re-conceptualisation of "online environments" as sets of applications and services offered by the developer and selected third-parties

Third parties:

Proposition P_{10}: Third parties such as System Integrators and Technology Partners provide critical extended sales, marketing and development resources for new software houses.

Proposition P_{11}: Independent Software Developers (ISDs) are an extension of the conventional Independent Software Vender (ISV) terminology to include all individuals, groups, and firms involved in application development external to the software vendor – including customers.

Proposition P_{12}: Software houses are now required to offer highly packaged system developer kits (SDKs) and strong online developer communities to support their ISDs.

Sensemaking:

Proposition P_{13}: New sensemaking capabilities are required for coordinators involved in the collection, integration and compilation of multiple concurrent application development inputs.

Proposition P_{14}: Cognitive sensemaking among software and technology developers influences the assessment and diffusion of application development skills, languages and technologies.

Proposition P_{15}: Retrospective emic representations provide stronger interpretations of events and decisions in terms of contextualisation over time.

Proposition P_{16}: Sequencing of emic representations and schemas before asking extension questions significantly influences the responses to those questions.

Open-Source:

Proposition P_{17}: Open-source philosophy enables rapid collective online application development, and use.

Proposition P_{18}: Open-source philosophy is now applicable to higher level applications, and not just operating systems and lower level system software.

Proposition P_{19}: Open-source philosophy will open up new forms of knowledge management and use.

Proposition P_{20}: Open-source philosophy challenges the conventional marketing conceptualisation of products, services, revenues, and collective knowledge creation and use.

Proposition P_9 highlights a need for software developers to revise their sensemaking toward re-conceptualising "online environments" as a set of applications and services offered by them and selected third-parties. The proposition is similar to Proposition P_5 except that it emphasises the cognitive perspectives of developers.

10.5.3 Third parties

A recurring theme throughout all case studies in the research project was increasing reliance on and involvement with third parties. A set of third-party activity-related propositions has been created.

Proposition P_{10} recognises the very important roles that System Integrators and Technology Partners play in providing extended sales, marketing and development resources way beyond the capabilities of new and emerging software houses.

Proposition P_{11} proposes a new category or entity related to software development. While the concept of the Independent Software Vendor (ISV) has been established for nearly two decades, a broader definition of software developers to include all individuals, groups and firms (including customers) using the software house's application(s), is required. The term Independent Software Developer (ISD) is proposed.

Proposition P_{12} recognises that software houses must now provide quite sophisticated services for their key third-party relationships. Highly packaged System Developer Kits (SDKs) and rich online developer community programs are now essential offerings for Internet-based software houses.

10.5.4 Sensemaking

A set of propositions has been created, addressing the sensemaking relating to application development and for the research approach.

Proposition P_{13} highlights a requirement for new sensemaking capabilities for individuals in the roles of coordinators of applications developed from concurrent multiple development inputs.

Coordinators of open-source related applications are expected to chart a broad direction for application features, and then accept inputs from several developers concurrently, and integrate and compile the inputs into one updated version or "build" of the application.

This rapid approach to application development was common across all the software houses in the research project, and not just for the open-source developer community.

Proposition P_{14} recognises that individual software and technology developers influence the assessment and diffusion of new application development skills, languages and technologies. Acceptance of Java for most applications referred to in the case studies for the project was an example of sensemaking at work among a set of individual software and technology developers – initially emanating from Sun Microsystems in the heart of Silicon Valley.

Proposition P_{15} states that the research approach of retrospective emic representations provides stronger interpretations of events and decisions, with a stronger understanding of contexts and timeframes.

Proposition P_{16} states that the sequencing of emic representations and schemas (such as DSA models, events chronology maps and cognitive maps) before asking extension questions influences responses to those questions. Interviewees are likely to respond with a mindset cast by the information provided in the package of models and maps.

10.5.5 Open-source

The Red Hat case study opened the door to the exploration of a new approach to application development that has now bloomed into a new philosophy for knowledge management and use. A set of relatively speculative propositions was created to explore this emerging and exciting area.

Proposition P_{17} recognises that the open-source philosophy encourages rapid collective online application development through licensing and collective developer community principles. The philosophy encourages developers to use, improvise and upgrade applications.

Proposition P_{18} recognises that at the end of 2003, the open-source philosophy is applicable for high-level application development. In 1999-2000, the conventional thinking was that open-source development could not be used beyond operating systems and basic system software.

Proposition P_{19} stakes a big claim for open-source philosophy opening up new forms of knowledge management and use. Application of the open-source philosophy has major implications for software, technology, media, business, education, and society implications.

Proposition P20 challenges conceptualisations of marketing strategies and activities. There is room for the articulation of an "open-source" marketing concept.

10.5.6 Research Propositions as a Theoretical Schema

The twenty propositions represent a theoretical schema that encapsulates the most significant issues to emerge from the research project – and that would be worthy of further exploration. The proposition framework represents generalisation from the case studies in the research project. However, real theory generalisation would have to be ascertained through subjecting the propositions to other related cognitive research projects.

10.6 Conclusions

The research project has explored cognitive sensemaking for the decision-making associated with the development and delivery of Internet-based software applications. Two new storytelling methods were used to explore the decision-making.

An advanced hermeneutic analysis framework was proposed and successfully applied to the case studies in the project. The multiple-level hermeneutic framework represents a significant advance on the conventional, single phase etic analysis typically used in B2B marketing research.

The storytelling method proposed also builds on established DSA used in B2B marketing research through critical additions of upgraded DSA models, events chronology maps, and cognitive maps – which, when viewed as a collective set of extended analysis – provide rich and dynamic insights into the decision-making about Internet-based software application development.

An extended case method approach was applied through collecting first-round etic data in the form of six written case studies – which contained emic introspections of past application developments for six software houses, with key individuals who were the subjects of the account or application.

The actual materials used with the key individuals were sets of strategic schemas including DSA models, event history maps, and new forms of cognitive maps – that constituted powerful and effective reconstitutions of the original written accounts. This material was in fact effective etic data, which were revised to produce second and third rounds of etic data.

New software applications were utilised to prepare and create cognitive maps, and to identify key cross-case-study concepts. Substantial findings were discussed for every case study, for application development cycles, selected application development issues, dominant logic, shared vision, key leverage points and strategic marketing issues. A set of summary issues was identified and discussed for each case study.

All substantial findings from all cases were consolidated and subjected to cross-study analysis. A composite DSA model was developed. Cross-case findings were analysed and discussed for dominant logic, shared vision, key leverage points and strategic marketing issues.

Finally, theory development was undertaken through assessing a set of Structuration propositions against case-study findings and summaries. Twenty propositions were finalised over five categories and presented as theory outputs from the project.

10.6.1 Discussion

The research project delivers contributions from each case study and at the cross-case project level. Table 10.9 presents contributions to strategic sensemaking from all case studies. Table 10.10 shows contributions to practice from all of the six cases.

All of these contributions are strong outputs from the project in their own right without further aggregation.

Propositions were developed around the themes of Structuration, computer-mediated environments (CMEs), third parties, sensemaking, and Open-sourcing. The propositions effectively capture cross-case summary points and point toward areas for future research.

10.6.2 Research Project Limitations

Table 10.11 presents the limitations for each case study in the research project. Most of the limitations were associated with limited numbers of interviews and respondents. Ideally a research project investigating cognitive perspectives on strategic issues should aim for several interviews with several respondents over an extended timeframe.

There were also limitations associated with not having access to relevant third parties in various cases. Insights from key third parties would have provided a more balanced perspective in most case studies.

The base accounts of decision-making in the form of Harvard Business School case studies were effective inputs for the research project, and there is applicability for research methodology for other types of accounts of application development. However, there is value in creating further accounts and case studies while the research project is "in-train".

Table 10.9 Contributions to Strategic Sensemaking.

Red Hat	Zaplet	NetDynamics	Kana	Intuit QuickBooks	Trilogy
First ever research project addressing mapping decision-making associated with "open-source" application development. Decision-making associated with a software house acting as an "open-source" ISV, taking an "open-source" application and packaging it with its own applications. "Metatheory" of an "Open-source" marketing concept.	**Mapping decision-making associated with commercialising a technology – at least to an early prototype stage. Mapping for creating as many ideas for potential applications built on an application technology as possible. Mapping decision-making processes for an organization al shift toward focus on enterprise applications.**	**Mapping associated with very rapid product development ("product development in Internet Time"). Mapping DSA for transition from embryonic business to acquisition by a large high-technology company. Mapping DSA issues associated with managing rapidly growing base of enterprise business accounts.**	**Contributes to strategic sense making through application of DSA augmented by cognitive mapping to rapid application development. Extended DSA approach maps insights on "scalability" of dynamic decision-making associated with rapid expansion of software applications. Applies extended DSA to channel selection and development. A distinct concept that emerges from extended DSA in the case is acknowledgement of both channels of distribution and channels of development.**	**Applies an extended DSA approach using DSA models, event history maps plus cognitive mapping to decision-making for application development and product strategy over time. Three strategic iterations are mapped and analysed from 1998 to 2002. Provides insight into both Internet application development and establishment of a computer mediated environment for business-to-business services and support. Provides mapping of decision-making associated with increasing inclusion of third-party applications and services in application development cycles. Highlights a fundamental transformation to strategic thinking to reconceptualise "application" (or product) development for computer mediated environments.**	**Extended DSA approach offers insights on turning an idea into an application in conjunction with turning a small start-up business into a substantial and competitive enterprise software provider. Provide unusual insights into a very strong logic for setting innovative application development processes and cycles. Offers mapping over time of decision-making associated with transformation from a "best-practice" product-driven business and application development approach into a vertical industry-focused business.**

Table 10.10 Contributions to Practice.

Red Hat	Zaplet	NetDynamics	Kana	Intuit QuickBooks	Trilogy
"Open-source" application development and use is one of the most important current issues for both software houses and Chief Information Officers. Highlights in challenges associated with concurrently working with tens of thousands of developers on applications. Highlight major requirement for very strong coordination and integration skills in order to dynamically integrate and compile very high numbers of application code inputs. Management of intellectual property associated with the "open-source" approach.	Valuable insights into challenges for entrepreneurial application developers seeking to create commercialise and deliver software applications with viable potential market demand. Highlights important contributions that venture capitalists can offer to emergent technology companies beyond funding. Highlights how contributions from venture capitalist may change over time and as a company restructures its business and market focus.	Provides insights for application developers seeking to create new businesses out of development of new products for revolutionary or discontinuous infrastructure. Provides insights for application developers and marketers managing product development within a multiple ISV strategic alliance arrangements.	Provides insights on both rapid application and new organisation development. Provides distinct insights into management of application development using mergers and acquisitions. Practical insights on channel and selection. The importance of System Integrators as extended developer and sales and marketing resources for small but growing software houses.	Provides valuable insights into transitioning a "traditional" software house with established strict internal application development processes and standards into a more "open" application development approach. Highlights decision-making associated with negotiating and incorporating third-party applications in application development processes. Substantial insights are offered on transitioning a popular application online. Additional decision-making is required to develop a virtual community with appropriate third-party services within the new community. Challenges traditional software house executives' views of what actually constitutes an "application".	Provides valuable insights into establishing "best-practice" application development groups. Provides valuable insights into transforming a software house from a strong product development focus across to industry-focused business groups and application development.

Table 10.11 Case Study Limitations.

Red Hat	Zaplet	NetDynamics	Kana	Intuit QuickBooks	Trilogy
Single substantial interview, plus supporting material from several secondary sources.	**Two substantial interviews with two interviewees.**	**Possible benefits from additional interviews including the NetDynamics Chief Scientist who was a key decision-maker and hands-on developer of the NetDynamics application. Possible benefits from interviews with developers from Sun Microsystems and Netscape who were all developing applications with each other and NetDynamics.**	**Possible benefits from additional insights from current Kana employees who were also with Kana during 1996-2000. Possible additional interviews with subset of Kana employees from companies acquired or from merger partners.**	**Benefits from gaining insights from selected third-party developers and service providers on Intuit's views and actual implementation of their online strategies.**	**More insights on Trilogy's transition to straight Internet-based applications during 1997-99 would have been desirable.**

There are also substantial challenges in researching an area where there is rapid change in terms of the turnover of the respondents. While there was excellent co-operation and support from all respondents in the project, great difficulties were encountered in the final validation of updated material, as a number of respondents had moved on to other companies, projects and activities.

A more formalised research project with funding and support from the industry or communities associated with Internet-based software houses may have enabled stronger and longer access to more respondents, and more triangulation of data sources. However, given the resource constraints associated with the project, quite substantial development of research methodology and contributions was achieved.

A number of research software application limitations were identified. While the NVivo application enabled the effective, rapid and dynamic development of cognitive maps, there were limitations in searching for and coding key words and concepts.

There were also limitations to the translation of digital recording inputs. At the time of the interview stage of the project (early 2002) the first digital recording systems came on the market – and all available transcription services could only process cassettes from conventional recorders and Dictaphones.

Advancing speech-recognition software is improving to the point where, within about two years (around 2007-08), digitally recorded interviews could be directly transcribed without transcription services – and perhaps input directly to appropriate analytical and mapping software applications. There is further discussion on this area in the next section of future research.

New information collection applications have the capacity to improve data collection and to analyse relatively unstructured information such as stories or accounts of decision-making. Earlier discussion in development of the composite DSA model in this chapter, highlighted emerging applications for advanced analysis based on a Semantic Web approach, which could assist in producing stronger insights, and perhaps different types of cognitive maps from those produced in this project.

Various recommendations for future research will address many of the research limitations noted for each case study and for the whole project.

There are also limitations associated with the context, and choices of sources of information beyond case-specific limitations. The research project used a specific context of Internet-based software houses which were generally small or growing "start-up" businesses.

The project would have been much more complex to set up and execute with multiple rounds of interviews and accounts with executives in a larger software house, or for instance, a larger multi-faceted services provider or vendor (for example, IBM Consulting or IBM Global Services, Infosys, Oracle, or SAP).

The project would also be more challenging to run in an information services context, or, for example, a financial services context – although it could be effectively applied to financial services application developers.

There are issues with selection of sources for the project. All sources chosen were decision-makers within the software house at the centre of an application development effort. There were no accounts from decision-makers associated with application development from third-parties working with each software house in the case studies in the research project.

Whether the third parties were ISVs, SIs, ISDs, channels partners, or customers, they may well have contributed different accounts – or even have provided additional etic inputs for a fuller view of the decision-making processes.

Overall, the author of this report believes that the proposed storytelling methodology developed for the project is best suited for contexts where there is real innovative application or product development, complemented by entrepreneurial activities associated with start-up companies.

10.6.3 Recommendations for Future Research

There are several opportunities for future research emerging from the current research project. Areas identified include application development research; applying research methodology to new marketing contexts; open-source research agenda; new software applications for cognitive research; combining human and system cognitive analysis; incorporation of emergence and complexity principles in cognitive research.

10.6.3.1 Application Development Research

There are opportunities to extend the current research to cover more software house case studies. Variations of the advanced hermeneutic analysis framework for analysing software houses could include writing up updated case studies and accounts of existing cases or completely new case studies. Advanced hermeneutic analysis could possibly be reframed, with changed or additional levels addressing the analysis of extended sensemaking much more to incorporate third-party perspectives in application research.

Another extension is to develop a cognitive theory for knowledge creation and management using software as a medium for such creation. At this point in time such creation would be likely to be through extending the current perspectives of application development and delivery.

Further research could be conducted on exploring the sensemaking associated with developing full online (or computer-mediated) offerings, as the current research project only provides an introductory glance into this area.

10.6.3.2 Applying Research Methodology to Other Contexts

The storytelling research methodology developed for the current project could be extended to analyse case studies and accounts for other industries and contexts. Examples could be computer hardware, media content, financial services, and various business-to-business contexts.

The methodology could be applied to other more conventional areas of B2B marketing decision-making contexts, providing useful research outputs.

The methodology could also be applied to B2C marketing contexts for new and innovative products and services, where accounts of customer experiences could perhaps be analysed against accounts of decision-making to develop and deliver the relevant products or services.

Case study researchers in the field of marketing, but also in management and business research and perhaps other academic fields, could take the storytelling method developed for the project, and use it to develop a comprehensive body of hermeneutic analysis complemented by strong DSA – resulting in really deep insight accounts, stories, scripts, scenarios, and even Semantic Nets, which would be similar to the gists of stories referred to by Schank (1998).

10.6.3.3 Open-source Research Agenda

A specific extension area emerging from the research project would be to set a research agenda exploring "open-source" philosophies in business and marketing contexts.

The research methodologies applied in the current project could be applied to a range of "open-source" software houses, developers, and even the whole "open-source" community.

There would be substantial value in exploring the sensemaking associated with business application users selecting and using "open-source" applications. There is also great potential to explore a concept of "open-source" marketing.

10.6.3.4 New Software Applications for Cognitive Research

Several benefits and limitations associated with the current software applications used for cognitive research were discussed in the project. Some quite exciting and revolutionary applications are under development or recently released that have the potential to stretch cognitive research analysis and mapping to new dimensions. New concept identifier applications based on semantic web analysis tools point the way toward much more efficient ways to summarise, code, and map concepts and issues. Copernic is a pointer to these new applications.

There is a large range of tools used for mapping computer networks and concepts that could be applied to cognitive analysis and mapping. New forms of mapping and visualisation will emerge out of "mapping cyberspace". For a substantial discussion of mapping and visualisation tools used in "mapping cyberspace", see Martin & Dodge (2001a & 2001b).

There is also a set of emerging technologies that will greatly improve triangulated data collection.

New digital recording and recognition technologies and applications will enable much faster collection and processing of interview data.

A significant extension of "interview data" will be in the form of newly released applications enabling direct digital video – and direct coding of video content for input into other applications (see ResearchWare 2003).

It is quite likely that during 2005, many of the existing software applications for cognitive analysis and mapping will be superseded by these new "semantic visual" application suites.

A further recommendation related to cognitive analysis would be for the development of applications to manage and map multiple-level hermeneutic analysis and interpretation

10.6.3.5 Combining Human and System Cognitive Analysis

Extended DSA analysis was applied in the current research project. However, a logical extension is to combine extended DSA with system decision analysis (SDA), that is, combine human sensemaking with computer sensemaking and mapping. Pattinson (2003) explored the conceptualisation and early application of "meta-DSA" with reference to online recommendations. There is a lot of potential for a wide and varying range of research using a "meta-DSA" or "DSA-SDA" concept.

A further extension of this concept is to recast the "meta-DSA" into a "human-system" hermeneutic analysis framework.

10.6.3.6 Incorporation of Emergence and Complexity Principles in Cognitive Research

A further and quite interesting area for further research is the incorporation of emergence and complexity principles in the sensemaking associated with decision-making. The Complex Systems Research Centre (CSRC) at the University of Technology, Sydney, is exploring the analysis of unstructured information using storytelling and sensemaking methods with a complexity/emergence perspective (see also Maani & Maharaj 2004).

Each of the hermeneutic analyses from the six case studies could be recast as representations of complex adaptive systems, and either analysed as those types of systems, or be subjected to re-analysis by the methods and tools associated with complex systems analysis.

10.7 Endnote

"**Mapping Implemented Strategies of Bringing Internet Products to Market Using Storytelling Research Methods"** is a project that has contributed significantly to B2B marketing research through the application of new and enhanced storytelling methods for analysing decision-making, with the development and delivery of application software for and within a new computer-mediated environment (CME).

The project represents a three-year journey into the world of rapid application development, of new and emerging online, open-source, enterprise Web Services environments, and reformation, after major upheavals in the broader high-technology digital information infrastructure and supporting industries.

Exploring the minds of application entrepreneurs was an exhilarating experience and a front-row seat for observing the great optimism about creating new businesses that are driving the twenty-first century information-based techno-economic infrastructure and paradigm (see Perez 2002; Freeman 2002; Arthur 2002).

The current research project represents a starting point for another journey – one that could travel into many new areas of cognitive strategic research associated with both human and computer-mediated environments. New sensemaking awaits those who choose to take the journey.

Appendices

Appendix I – Interview Package

Interview packages were developed for each case study and contain a full set of one DSA model, one event history map (Renamed to events chronology map after the field collection process), three cognitive maps; and extension questions.

Interview packages were adapted for different participants and for multiple interviews. For example, some interviews covered the DSA models and event history maps in one session, with cognitive maps addressed in follow-up interviews; interviews were a mix of face-to-face and follow-up digitally recorded interviews. Some participants focused on specific application development issues and cognitive maps, especially where they offered additional insights, or were not directly involved in the central activities in some cognitive maps.

Adapted interview packages were sent to all participants before the scheduled interview. All but two interviews were recorded either using voice-activated recorders, or in the case of telephone interviews, with digital telephone recording software. Permission to record the interview was gained from the participant in all cases prior to the interview's commencement. All interview notes and recordings were transcribed into Word documents that represent primary research input for the project.

Updated versions of the interview package with revised models and maps were sent to a subset of participants for subsequent validation.

The Interview Package Template is presented below:

Appendix I
Interview Package

Interview packages were developed for each case study containing a full set of one DSA Model, one Event History Map (Renamed to Events Chronology Map after the Field Collection Process), three Cognitive Maps; and Extension Questions

Interview packages were adapted for different participants and for multiple interviews. For example some interviews covered the DSA Models and Event History Maps in one session, with Cognitive Maps addressed in Follow-up interviews; interviews were a mix of face-to-face (see Appendix II) and follow-up digitally recorded interviews. Some participants focused on specific application development issues and Cognitive Maps, especially where they offered additional insights, or were not directly involved in the central activities in some Cognitive Maps.

Adapted Interview packages were sent to all participants before the scheduled interview. All but two interviews were recorded either using voice activated recorders, or in the case of telephone interviews, with digital telephone recording software. Permission to record the interview was gained from the participant in all cases prior to the interview's commencement. All interview notes and recordings were transcribed into Word Documents that represent primary research input for the project.

Updated versions of the interview package with revised models and maps were sent to a subset of participants for subsequent validation.

The Interview Package Template is presented below:

Hugh Pattinson – PhD Project

"Mapping Multi-Party Shared Vision And Internet-Based Software Applications Using Storytelling Methods"

<<Face-To-Face>> or <<Telephone>> Interview Package

<u>Case:</u>

Respondent Name:

Respondent Title:
Date of Interview:
Interview Location:

Introduction

Thank you for agreeing to participate in this research. As previously noted in my letter to you I am undertaking a study for my Ph.D. dissertation on exploring the development and marketing of Internet-based software applications. I am particularly interested in retrospective comments and perspectives of key persons involved in developing and delivering these new applications.

I would like to explore with you development and delivery issues associated with *<<Company and Application>>.*

I have developed a set of diagrams plus questions describing these issues based on a previously published case study on <<*Case Study Title*>>, which are included in this package.

All five diagrams relating to <<*Company*>> are included in this package. However, the diagrams that will be discussed during the telephone interview will be:

Figure 1: *<<DSA Model>>*
Figure 2: *<<Events Chronology (History) Map>>*
Figure 3: <<– 1^{st} Cognitive Map>>
Figure 4: <<2^{nd} Cognitive Map>>
Figure 5: <<3^{rd} Cognitive Map>>

Key questions relating to each diagram that will be covered during the telephone call:

- ***(For Figure 1) Does this diagram represent a sufficiently detailed description of the decision-making processes associated with the development and delivery of <<Company>> applications?***

- ***(For Figure 2) Does this Event History Map represent a sufficiently detailed description of a history of the development and delivery of <<Application>>?***

- ***(For Figure 3) Does this Cognitive Map represent a sufficiently detailed description of <<1st Cognitive Map Title>>?***

- ***(For Figure 4) Does this Cognitive Map represent a sufficiently detailed description of <<2nd Cognitive Map Title>>?***

Depending on time constraints, there will be a set of extension questions relating to:

- ***Shared vision associated with the development and delivery of <<Company>> technology/applications***
- ***Dominant Logic present in development and delivery of <<Company>> technology/applications***
- ***Major change or leverage points in the diagrams***
- ***Identification of strategic marketing issues associated with in development and delivery of <<Company>> technology/applications***

I look forward to discussing with you development and delivery issues associated with <<Company>> Technology and Applications.

Thank you,

Hugh Pattinson
Coordinator E-Business Marketing Programs
University Of Technology, Sydney

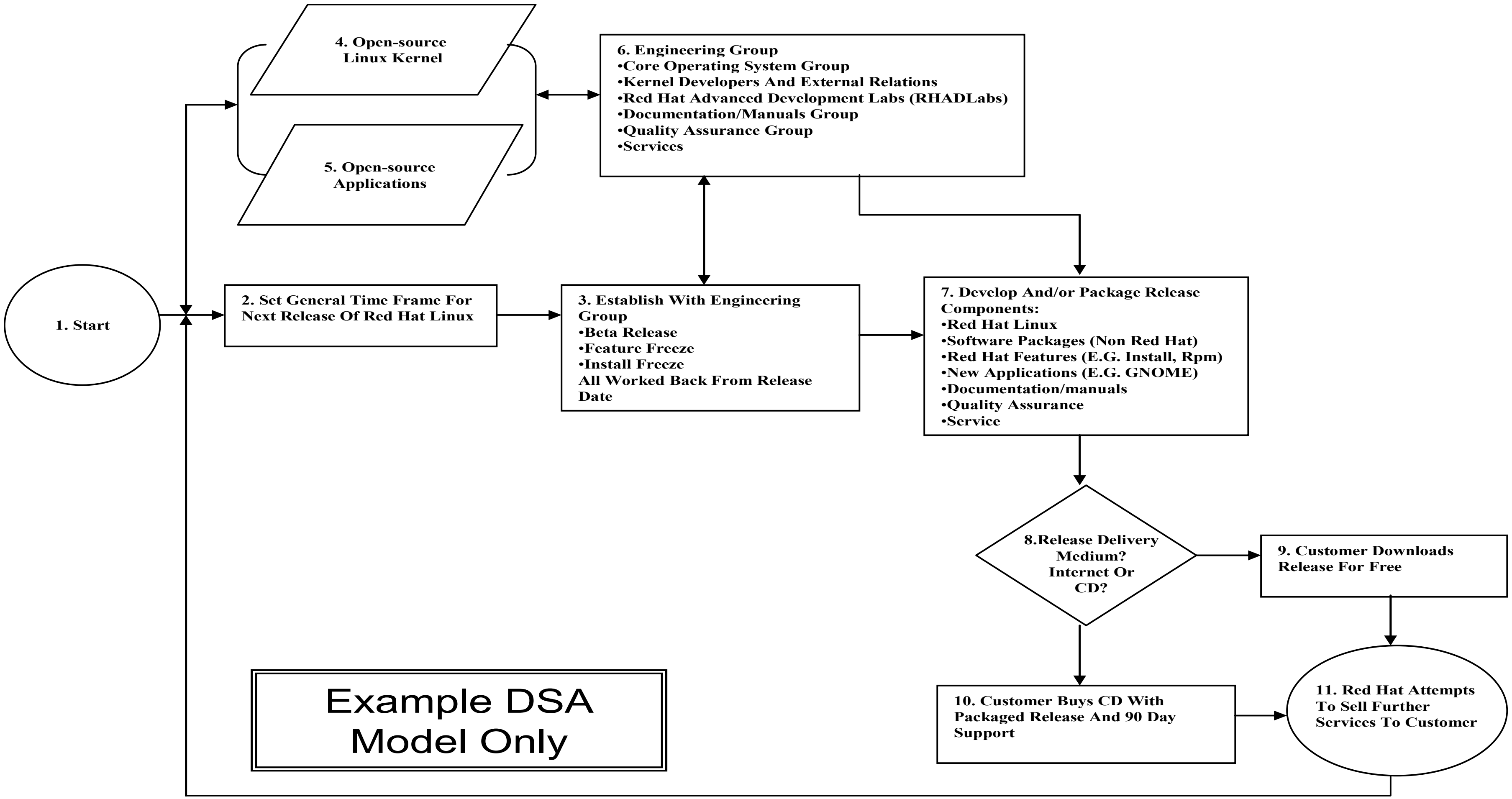


Figure 1 - Summary DSA Model For Red Hat Application Development 1998 – Notes on Each Point In This Figure Can Be Found in Table 1

Adapted from: MacCormack, Alan, Kerry Herman (2000), "Red Hat And The Linux Revolution", Harvard Business Case No. 9-600-009

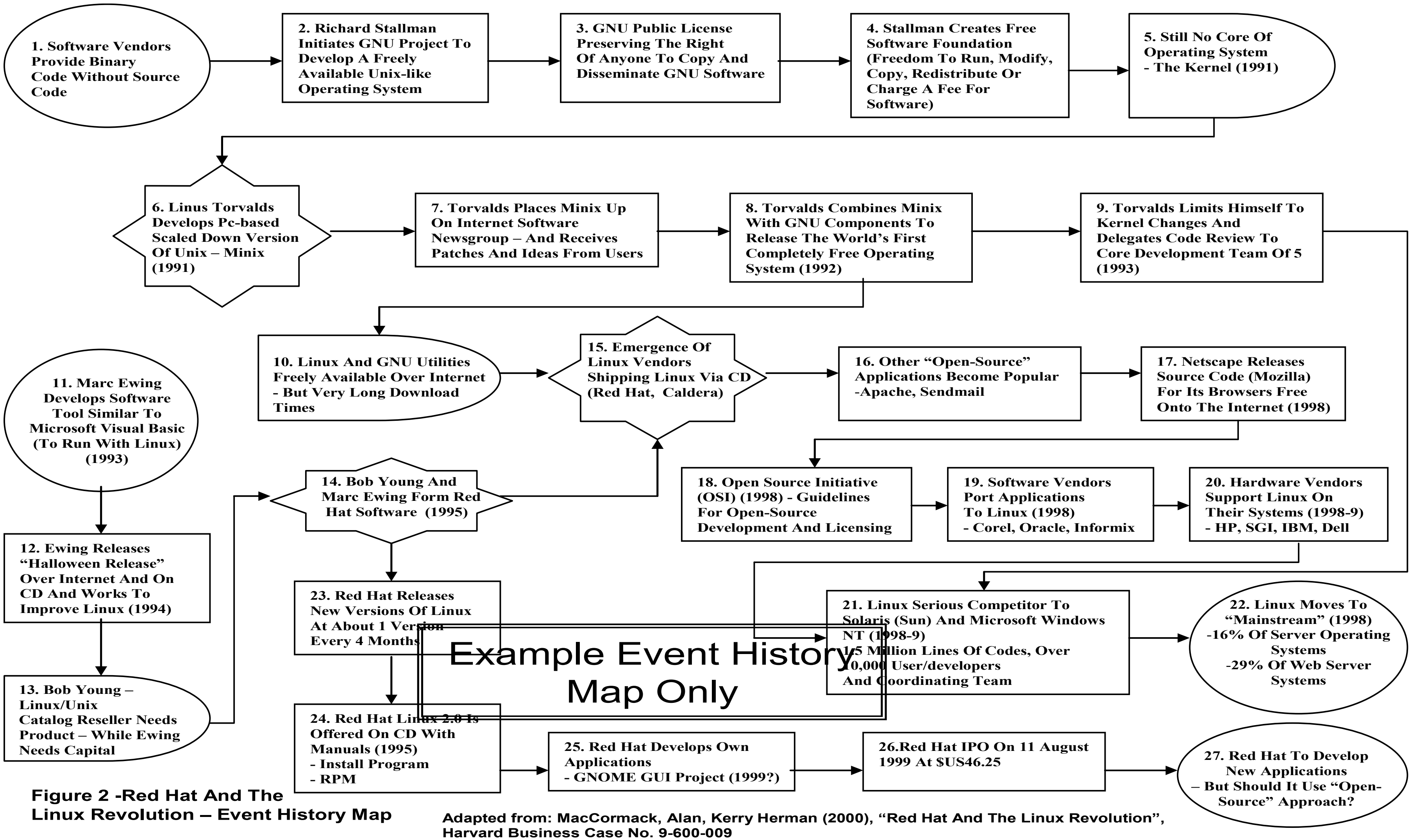


Figure 2 -Red Hat And The Linux Revolution – Event History Map

Adapted from: MacCormack, Alan, Kerry Herman (2000), "Red Hat And The Linux Revolution", Harvard Business Case No. 9-600-009

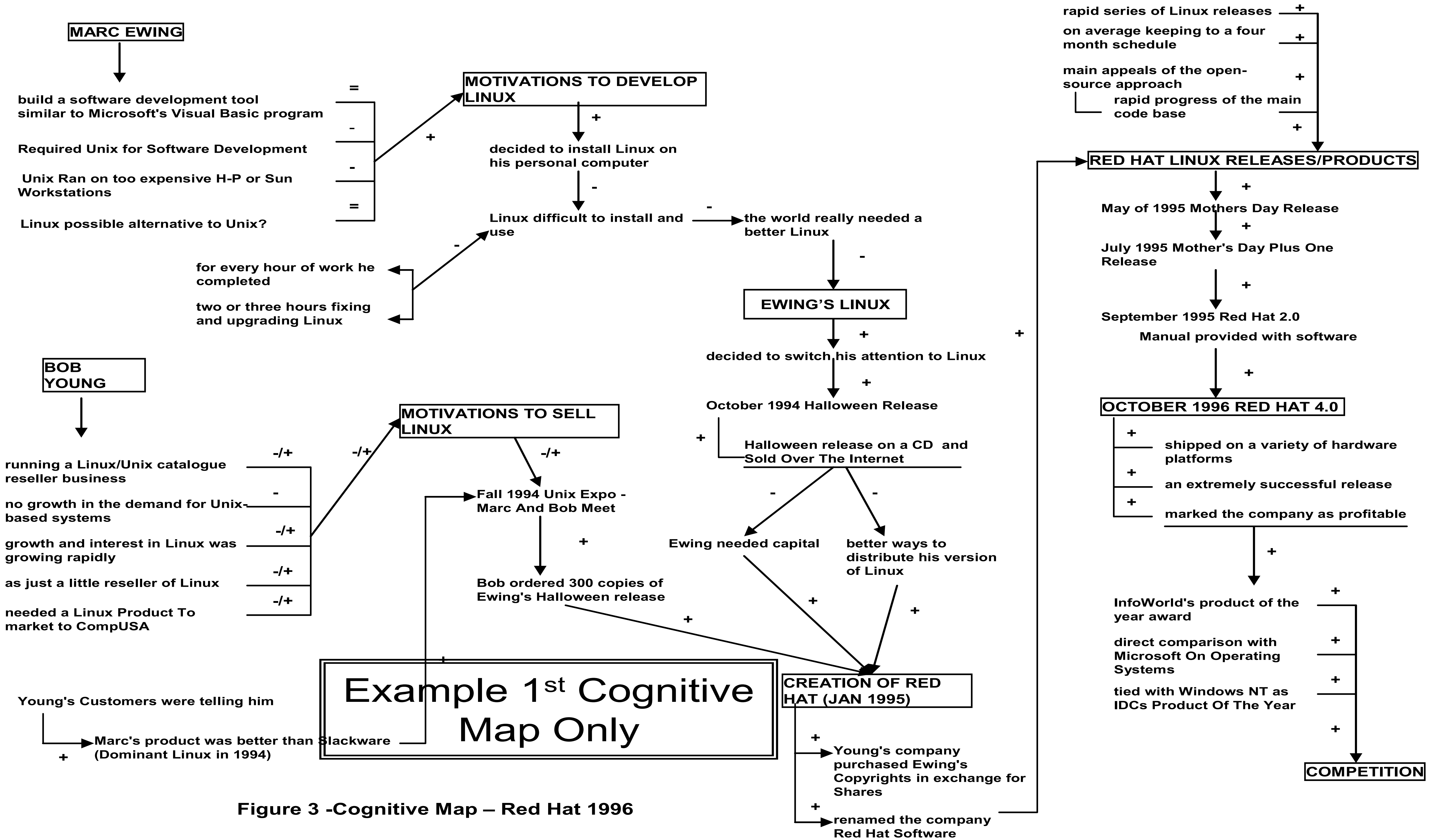


Figure 3 -Cognitive Map – Red Hat 1996

Adapted from: MacCormack, Alan, Kerry Herman (2000), "Red Hat And The Linux Revolution", Harvard Business Case No. 9-600-009

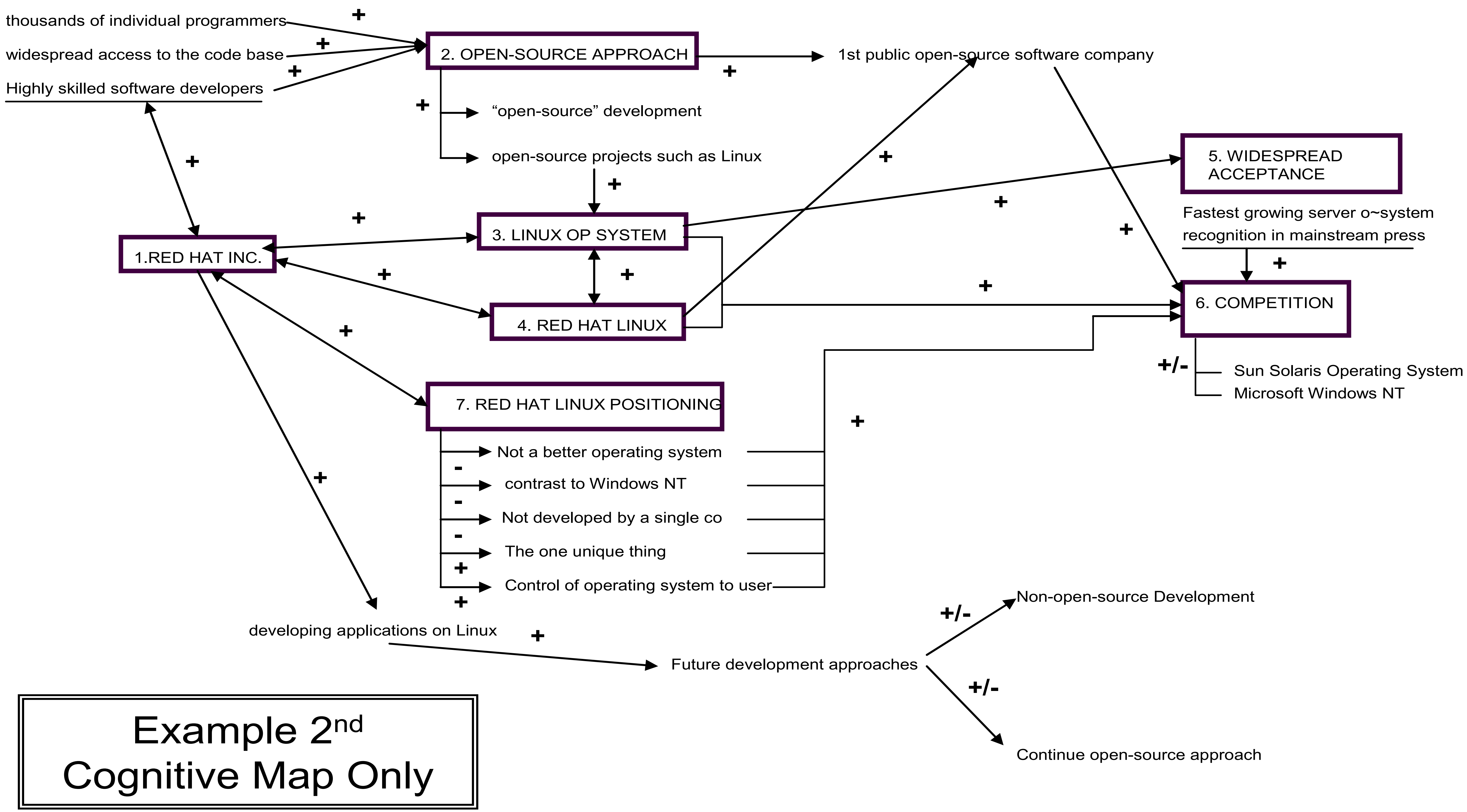


Figure 4 - Cognitive Map – Red Hat 1999

Adapted from: MacCormack, Alan, Kerry Herman (2000), "Red Hat And The Linux Revolution", Harvard Business Case No. 9-600-009

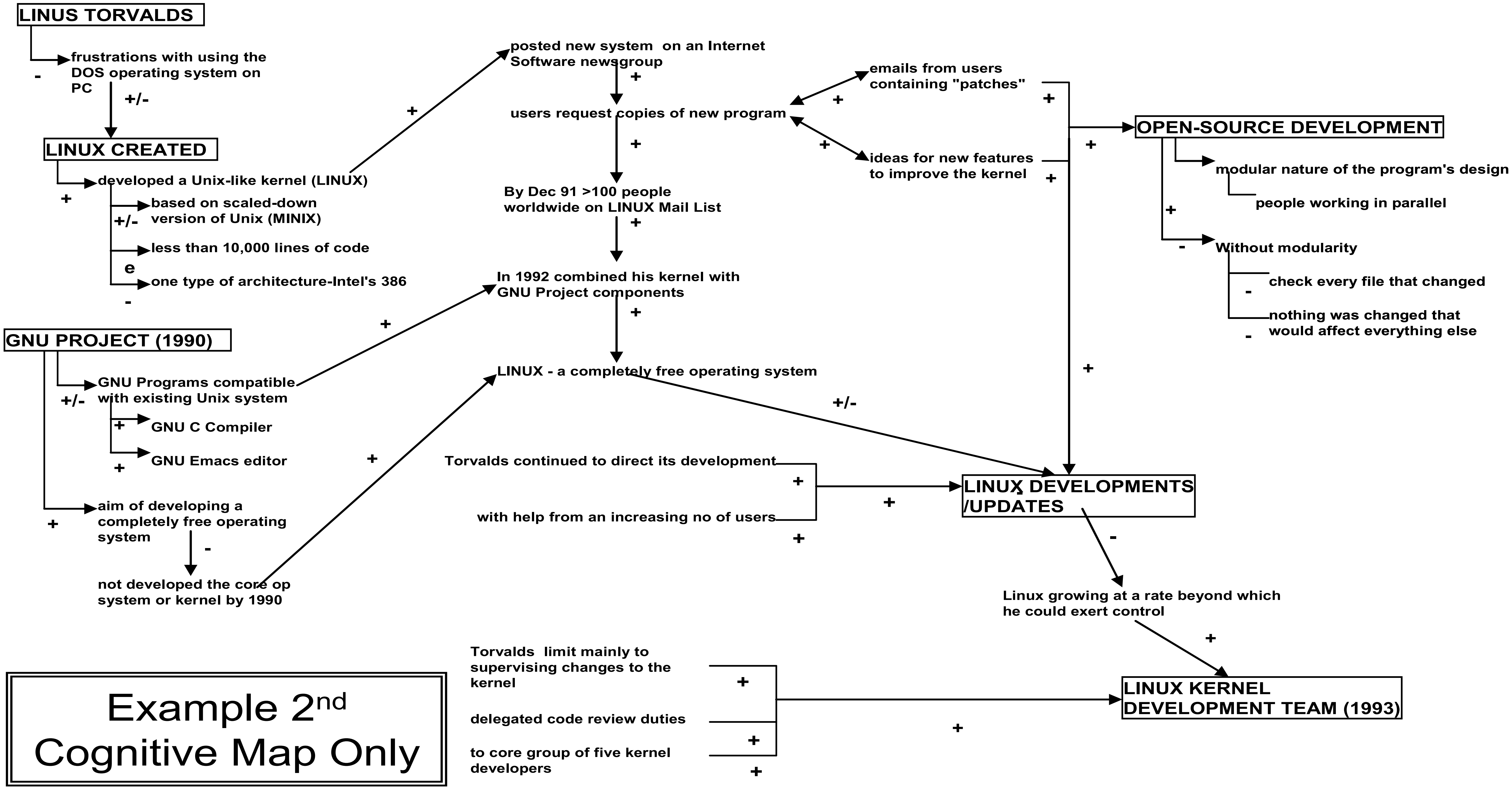


Figure 5 - Cognitive Map – Torvalds Creates LINUX

Adapted from: MacCormack, Alan, Kerry Herman (2000), "Red Hat And The Linux Revolution", Harvard Business Case No. 9-600-009

Appendix II- Face-to-Face (In-depth) Interview Package

Hugh Pattinson – PhD Project

"Mapping Multi-Party Shared Vision And Internet-Based Software Applications Using Storytelling Methods"

In-Depth Interview Package

Check-List

- **Interview Consent**
- **Introduction**
- **Walkthrough DSA Model**
- **Walkthrough Event History Map**
- **Walkthrough Cognitive Map(s)**
- **Extension Research Questions**
- **Further Comments**
- **End Of Interview**

UNIVERSITY OF TECHNOLOGY, SYDNEY
CONSENT FORM - PhD RESEARCH PROJECT

I ________________________agree to participate in the research project ***"Mapping Multi-Party Shared Vision And Internet-Based Software Applications Using Storytelling Methods"*** being conducted by ***Hugh Pattinson*** of the University of Technology, Sydney, for the purpose of his PhD degree.

I understand that the purpose of this study is to explore the development and marketing of Internet-based software applications. There will be an emphasis on retrospective comments and perspectives of key persons involved in developing and delivering these new applications.

I understand that my participation in this research will involve a brief interview in person where the researcher will write notes related to the questions presented in the interview, supported by audiotape recording. I understand that I can request the cessation of audio taping at any stage during the interview.

I am aware that I can contact ***Hugh Pattinson*** or his/her supervisor ***Professor Ken Miller (UTS) or Professor Arch Woodside (at Boston College)*** if I have any concerns about the research. I also understand that I am free to withdraw my participation from this research project at any time I wish and without giving a reason.

I agree that ***Hugh Pattinson*** has answered all my questions fully and clearly.

I agree that the research data gathered from this project may be published in a form that does not identify me in any way.

____ ____/____/____
Signed by

____ ____/____/____
Witnessed by

NOTE:
The University of Technology, Sydney Human Research Ethics Committee, has approved this study. If you have any complaints or reservations about any aspect of your participation in this research that you cannot resolve with the researcher, you may contact the Ethics Committee through the Research Ethics Officer, Ms Susanna Davis (ph: 61 2 - 9514 1279, Susanna.Davis@uts.edu.au). Any complaint you make will be treated in confidence and investigated fully and you will be informed of the outcome.

Interview Record

Case: ______________________________

Respondent Name: ______________________________

Respondent Title: ______________________________

Date of Interview: ______________________________

Interview Location: ______________________________

Models Or Maps To Be Presented To The Interview Participant:

(This Table Must Be Completed Before Interview, but May be Modified If The Participant Has An Unexpected Role In Developing Specific Models Or Maps):

	YES	NO	YES (DURING INTERVIEW)
1. DSA Model			
2. Event History Map			
3. Cognitive Map 1 (Description)			
4. Cognitive Map 2 (Description)			
5. Cognitive Map 3 (Description)			
6. Research Extension Questions			

Introduction

Thank you for agreeing to participate in this research As previously noted in my letter and/or telephone call and/or e-mail to you I am undertaking a study for my Ph.D. dissertation on exploring the development and marketing of Internet-based software applications. I am particularly interested in retrospective comments and perspectives of key persons involved in developing and delivering these new applications.

I would like to explore with you development and delivery issues associated with <<Application>>.

I have developed a set of diagrams plus questions describing these issues based on a previously published case study on <<Application>> that I would like to discuss with you. Can we go through these diagrams and questions now?

Go through in order the sections ticked YES in the Table above only – except where they may be sufficient evidence in one section to suggest that an additional section should be covered.

1. *DSA Model*

This diagram attempts to describe decision-making associated with the development and delivery of <<Application>>.

<<Show DSA Model >><<See Sample as Figure 1>>

Does this diagram represent a sufficiently detailed description of the decision-making processes associated with the development and delivery of <<Application>>?

<<Walkthrough every step of the DSA Model and take notes on comments about the Model>>

Note: is there any evidence to suggest that the participant should comment on other Maps that they were not initially marked for? If YES, then extra models or maps may be added and noted in order of this running sheet schedule.

2. Event History Map

This diagram attempts to describe the chronology of events associated with the development and delivery of <<Application>>.

<<Show Event History Map>><<See Sample as Figure 2>>

Does this diagram represent a sufficiently detailed description of a Chronology of Events associated with the development and delivery of <<Application>>?

<<Walkthrough every step of the Events chronology mapand take notes on comments about the Model>>

Note: is there any evidence to suggest that the participant should comment on other Maps or Models that they were not initially marked for? If YES, then extra models or maps may be added and noted in order of this running sheet schedule.

3. Cognitive Map 1

I have attempted to develop a mental or cognitive associated with the specific area of <<Note on issue or area that the Cognitive Map is based on inserted here >>. All or most or a significant part of this Cognitive Map is based on published perspectives of your views of this issue or area.

<<Show Cognitive Map 1>><<See Sample as Figure 3>>

Does this Cognitive Map represent a sufficiently detailed description of <<the area or issue noted in the previous paragraph>>?

<<Walkthrough every step of Cognitive Map 1 and take notes on comments about the Map>>

Note: is there any evidence to suggest that the participant should comment on other Maps or Models that they were not initially marked for? If YES, then extra models or maps may be added and noted in order of this running sheet schedule.

4. **Cognitive Map 2**

I have attempted to develop a mental or cognitive associated with the specific area of <<Note on issue or area that the Cognitive Map is based on inserted here >>. All or most or a significant part of this Cognitive Map is based on published perspectives of your views of this issue or area.

<<Show Cognitive Map 2>><<See Sample as Figure 4>>

Does this Cognitive Map represent a sufficiently detailed description of <<the area or issue noted in the previous paragraph>>?

<<Walkthrough every step of Cognitive Map 2 and take notes on comments about the Map>>

Note: is there any evidence to suggest that the participant should comment on other Maps or Models that they were not initially marked for? If YES, then extra models or maps may be added and noted in order of this running sheet schedule.

5. Cognitive Map 3

I have attempted to develop a mental or cognitive associated with the specific area of <<Note on issue or area that the Cognitive Map is based on inserted here >>. All or most or a significant part of this Cognitive Map is based on published perspectives of your views of this issue or area.

<<Show Cognitive Map 3>><<See Sample as Figure 5>>

Does this Cognitive Map represent a sufficiently detailed description of <<the area or issue noted in the previous paragraph>>?

<<Walkthrough every step of Cognitive Map 3 and take notes on comments about the Map>>

Note: is there any evidence to suggest that the participant should comment on other Maps or Models that they were not initially marked for? If YES, then extra models or maps may be added and noted in order of this running sheet schedule.

6. Extension Questions

I have a set of questions designed to expand on some ideas and presented in the diagrams we have walked through and discussed. Can we go through these questions now?

6a. Based on the diagrams we have discussed, what type or level of shared vision was associated with the development and delivery of <<Application>>? Who was involved in developing share vision associated with the development and delivery of <<Application>>?

<<The participant may refer to the Models or Maps that were previously discussed with them>>

6b. Who do you think dominated the decision-making processes or the strategic thinking associated with the Models and Maps we have discussed in this interview?

<<The participant may refer to the Models or Maps that were previously discussed with them>>

6c. Where do you think were the major change or leverage points in the Models and Maps discussed in this interview

<<The participant may refer to the Models or Maps that were previously discussed with them>>

6c. What do you are think were the strategic marketing issues associated with the Models and Maps covered in this interview?

<<The participant may refer to the Models or Maps that were previously discussed with them>>

7. **Further Comments**

Do you have any further comments regarding the Models and Maps discussed in this interview, or on this project?

Note: if there are some key participants who declined or are missing from this case, then ask the participant if there may be other people who should be contacted regarding the Case.

End Of Interview

Thank you for your time today. As token of my appreciation for participating in this project, I have two books for you, the first is <<book name and author>> and the second is <<book name and author>>

I will be updating the Models and Maps used in our discussion and writing comments on extension issues in my PhD dissertation.

I understand that I can contact you if I require further clarification of points raised in this interview.

I will also provide a summary to you of the findings for this case and the research project.

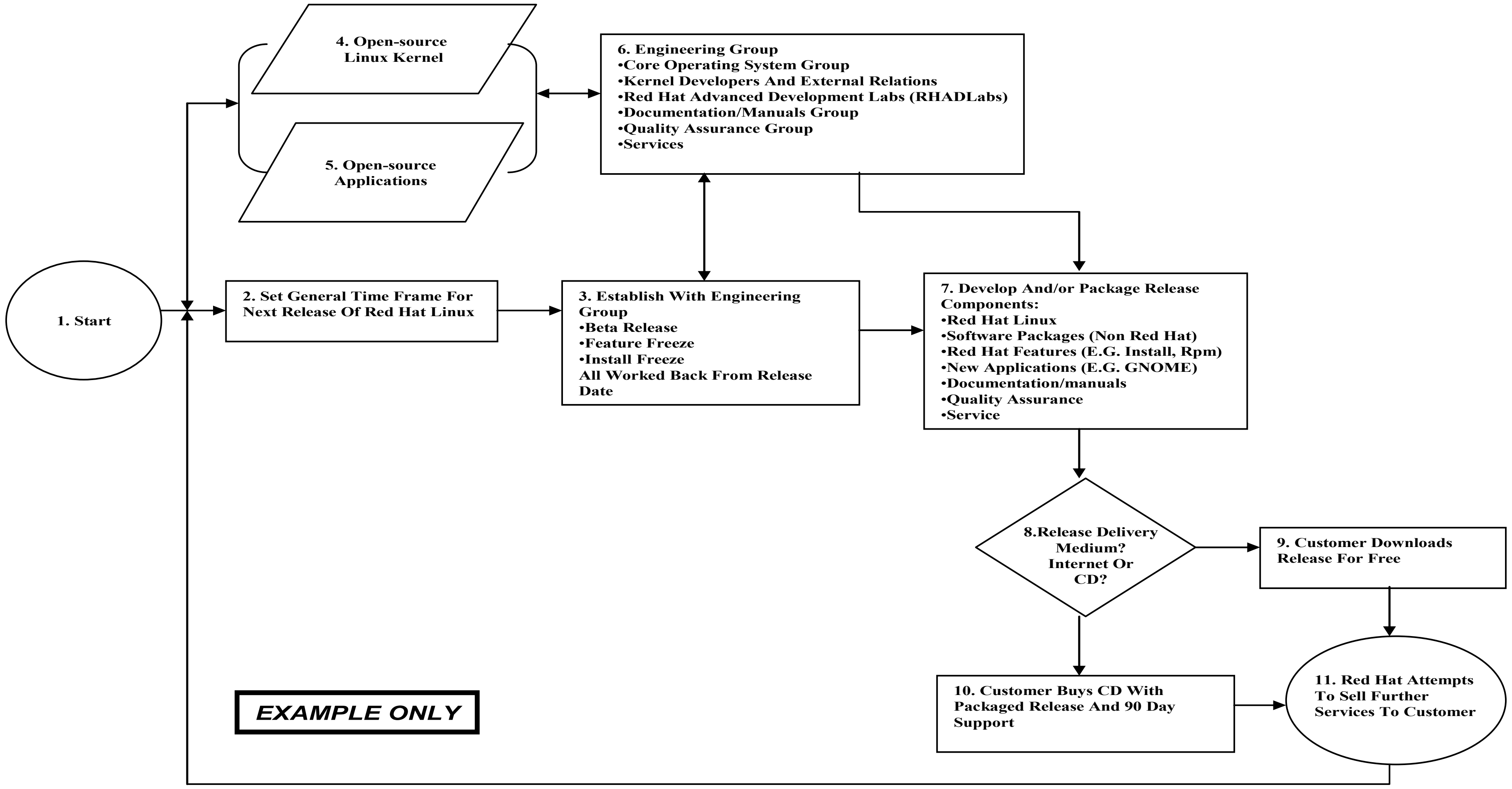


Figure 1 - Summary DSA Model For Red Hat Application Development 1998 – Notes on Each Point In This Figure Can Be Found in Table 1

Adapted from: MacCormack, Alan, Kerry Herman (2000), "Red Hat And The Linux Revolution", Harvard Business Case No. 9-600-009

Table 1 - Notes on Summary DSA Model 1998 (Figure 1)

1./2. Time Frames For Red Hat Linux releases are dependent on developments of Linux Kernel and other Open-Source Applications. New versions of Kernel and applications are frequency, some are updated daily

3. Specific application development lifecycle stages are negotiated with the Engineering Group and worked back from a release date

4/5. The Linux Kernel and associated Open-source applications are developed collectively by over 10,000 contributors of code, patches and ideas. There is a core group of 5 developers based around the globe, who review code development. Several programmers within various Red Hat Engineering Groups are part of the Open-source community.

6. Red Hat's Engineering Group (Director Erik Troan) has five main divisions plus a Services Function as outlined below:

<u>Core Operating System</u> (Cristian Gafton)
•Shipped Versions of Red Hat Linux
•Install Code
•Red Hat Package Program Manager (RPM)

<u>Red Hat Advanced Development Labs (RHADLabs)</u>
Mike Fulbright)
•Think-tank for Red Hat Linux Development
•New Applications
•GNOME – GUI for Linux
•GNOME was an open-source project managed outside Red Hat by Miguel de Icaza in Mexico City

<u>Kernel Developers And External Engineering Relations</u>
(Doug Ledford)
•Linux Kernel Development
•Members are prominent Linux Programmers around the globe but with little accountability to Red Hat
•Work with Hardware and Software Vendors on porting and engineering issues

<u>Quality Assurance</u> (Mike Wangsmo)
•Worked With Other Divisions to Ensure Code Quality
•Maintained Databases Associated With Automated Testing And Bug Tracking

<u>Documentation/Manuals</u> (Ed Bailey)
•Produced All the Documentation And Manuals for Products (There were 18 Manuals With Red Hat Linux6.0, 2 which were shipped with the product)

EXAMPLE ONLY

<u>Red Hat's Services Offerings in 1999</u> included:
•Consulting And Professional Services (pre and post-installation advice and consulting)
•Training And Certification (Delivering courses to non-Red Hat Engineers)
•Technical Support at Three Levels
 •Level 1 - basic 24x7 telephone support
 •Level 2 - more complex problems that could not be fixed by Independent Providers
 •Level 3 – issues requiring changes to the code
•20 Support Engineers provided Level 2 and 3 Support by Summer 1999

7. The Engineering Group develops and packages Red Linux "Product" which includes Kernel, Red Hat Enhancements and Applications And open-source applications. Red Hat 6.0 consisted of 645 different software packages of which Red Hat's contributed only 35 packages. Red Hat's RPM package enabled a user to keep "pristine" applications sources separate to patches and recent updates – and then to integrate them and created updated package versions.

8/9/10. Red Hat Linux is available at no charge over the Internet, but in 1999 would take about 36 hours to download. Red Hat Linux is also available in CD-ROM form through OEMs such as DELL (as an operating system option). Red Hat Linux is typically sold on a CD-ROM with two online manuals plus 90 day support.. Red Hat became a significant supplier of Linux as it one of the earliest companies to offer Linux via CD, plus it added vital utility applications such as an install program and RPM, which greatly assisted users to install and manage their Linux operating environments

11. Red Hat faces a major challenge/opportunity to develop revenue streams through selling further Services to its customers

<u>Further Observations:</u>

•Marketing and selling as words, terms, or function are not mentioned at all in the case
•Red Hat is an active participant in the open-source community – and is highly dependent on the developments within that community
•The Internet is a revolutionary distribution and communication channel for application development and delivery

Adapted from: MacCormack, Alan, Kerry Herman (2000), "Red Hat And The Linux Revolution", Harvard Business Case No. 9-600-009

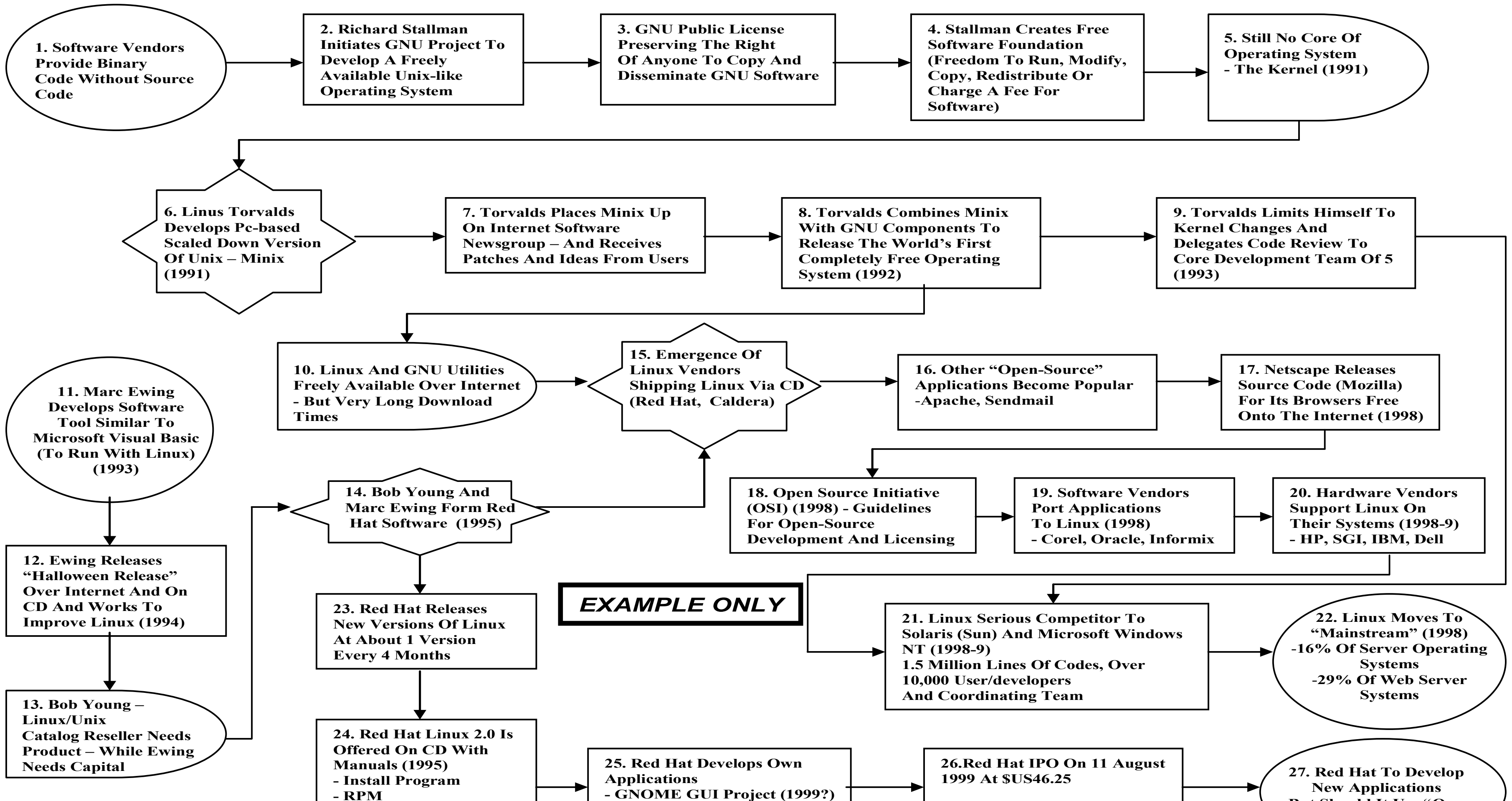


Figure 2 -Red Hat And The Linux Revolution – Event History Map

Adapted from: MacCormack, Alan, Kerry Herman (2000), "Red Hat And The Linux Revolution", Harvard Business Case No. 9-600-009

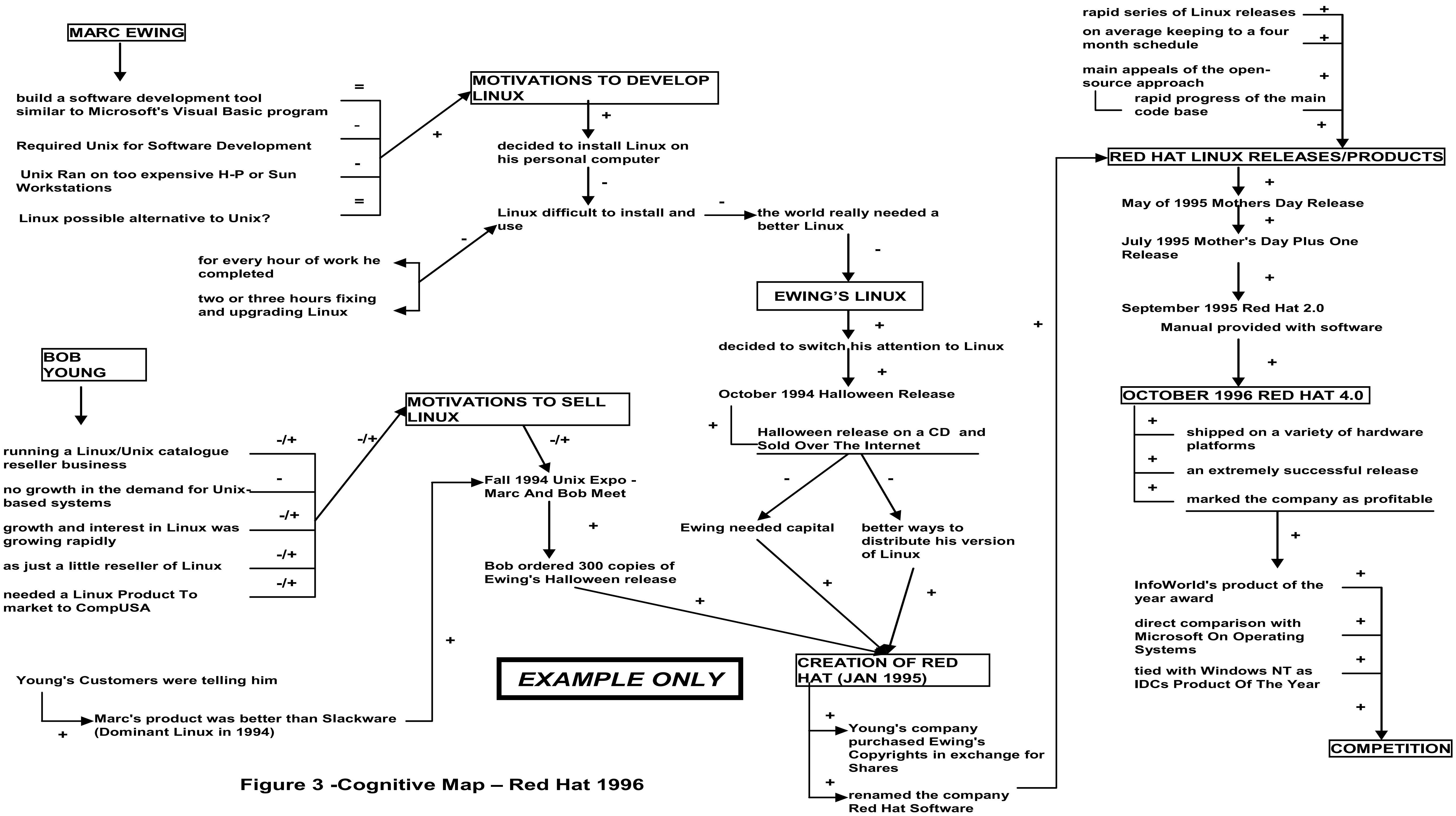


Figure 3 -Cognitive Map – Red Hat 1996

Adapted from: MacCormack, Alan, Kerry Herman (2000), "Red Hat And The Linux Revolution", Harvard Business Case No. 9-600-009

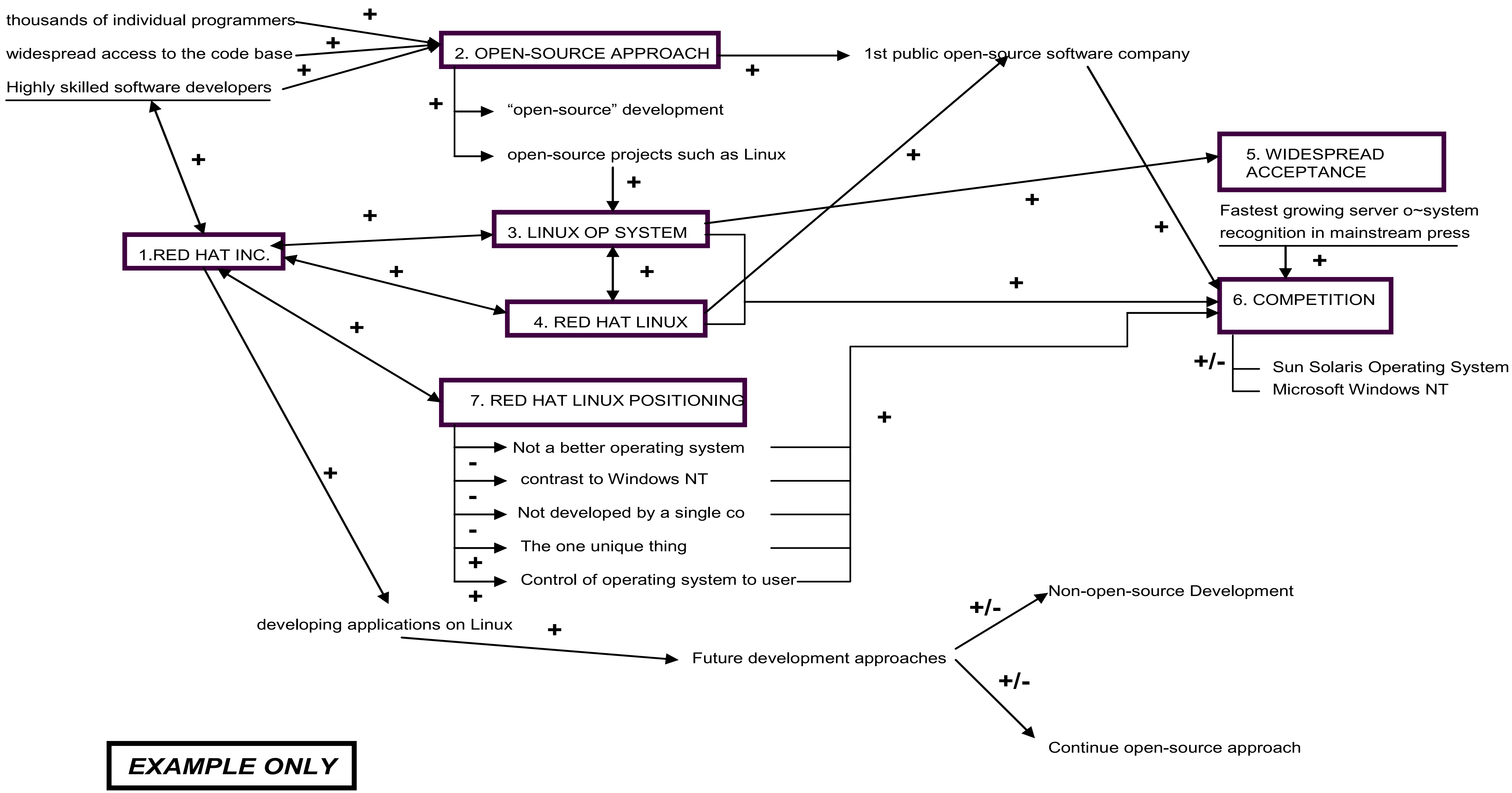


Figure 4 - Cognitive Map – Red Hat 1999

Adapted from: MacCormack, Alan, Kerry Herman (2000), "Red Hat And The Linux Revolution", Harvard Business Case No. 9-600-009

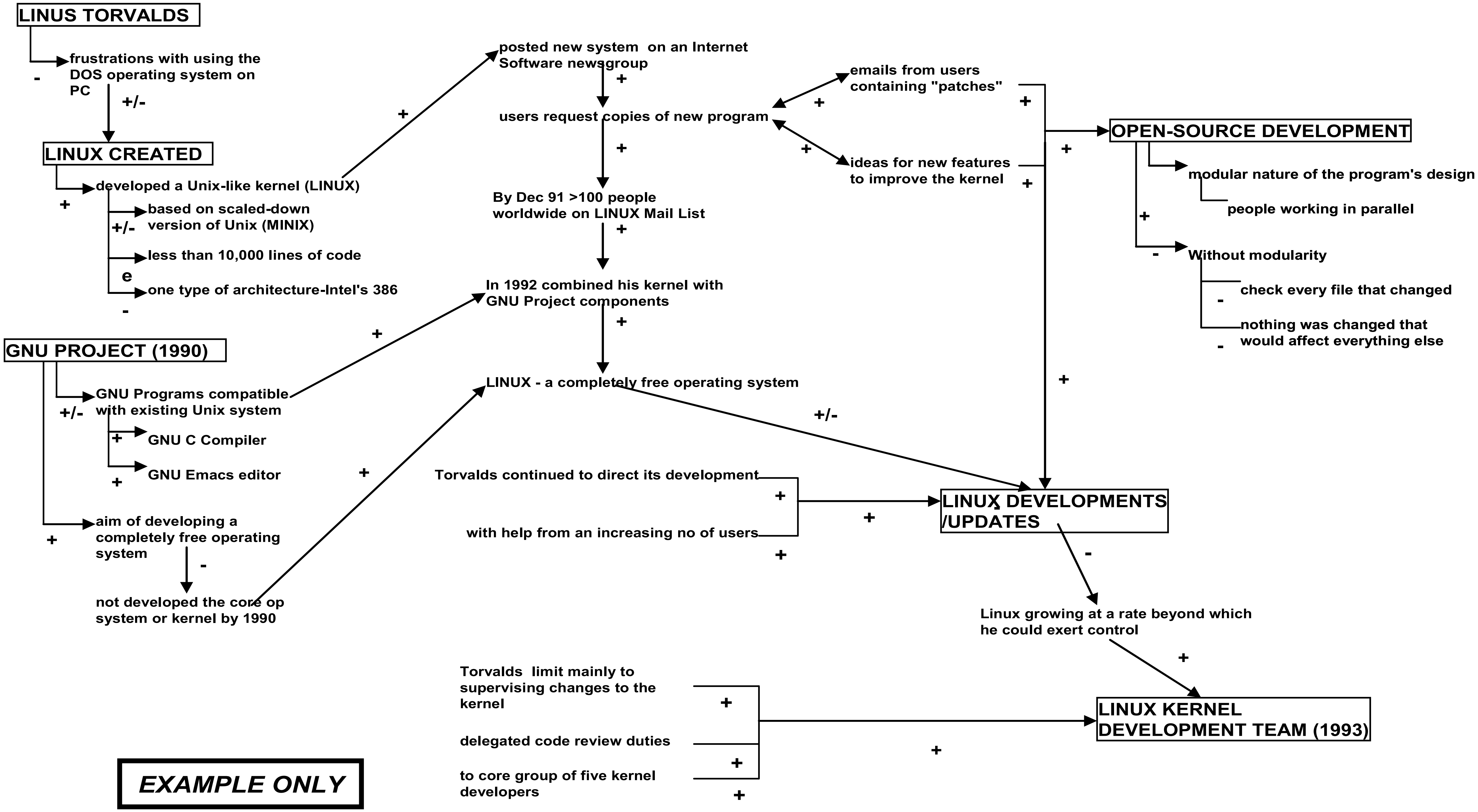


Figure 5 - Cognitive Map – Torvalds Creates LINUX

Adapted from: MacCormack, Alan, Kerry Herman (2000), "Red Hat And The Linux Revolution", Harvard Business Case No. 9-600-009

Appendix III - External Auditor's Evaluation Form (EAEF) and Package

Appendix III

External Auditor's Evaluation Form (EAEF) and Package

The External Auditor's Evaluation Form (EAEF) was developed for evaluation and validation of the cognitive mapping development process employed for the research project.
All selected auditors were sent a hardcopy version of the EAEF package, followed by an electronic version sent by e-mail. Responding auditor's returned electronically completed EAEF packages.

The EAEF Package is presented below:

Hugh Pattinson – PhD Project

"Mapping Multi-Party Shared Vision And Internet-Based Software Applications Using Storytelling Methods"

External Auditor's Evaluation Form (EAEF)

Introduction

Thank you for agreeing to participate in this research. I have undertaken a set of case studies exploring decision-making associated with the development and delivery of Internet-based software applications.

The case study methodology to explore and map decision-making includes the following elements:

1. A Harvard Business School Case Study that addresses aspects of internet-based application development for a selected software house. All initial models and maps are derived from this source document

2. A Decision Systems Analysis (DSA) Model for the software house's application development cycle. The DSA Model is developed from relevant extracts from the source document.

   DSA Model construction was based on methodology outlined by Howard, Hulbert & Farley (1975) and extended by Woodside & Wilson (2000) with consideration of third-party participation in product development (see Biemans 1989). (See also updated perspectives on these contributions in Hulbert 2003, Woodside & Wilson 2003 and Biemans 2003).

3. An Event History Map covering key events set in the timeframe that the source document refers to. The Event History Map was based on methodology used by Woodside & Wilson (2000)

External Auditor's Evaluation Form and Package (Cont'd)

1. A set of Cognitive Maps developed from extracts from the source document. Typically for each Case Study there are three cognitive maps – one from early in the source document, one from late in the source document, and one on a key issue in the source document.

   Methodology for creation of cognitive maps was developed from Huff (1990); Huff, Huff & Barr (2000); and Huff & Jenkins (2002). Coding schema was adapted from Huff, Narapeddy & Fletcher (1990). Initial Cognitive Map formats were adapted from Nath & Newell (1998), and extended with nested section formats

All models and maps were reviewed through multiple interviews with subjects either directly named in the source documents, or with subjects who were actively involved in the application development referred to in the source documents. All models and maps were updated, and in some cases, additional models and maps were created. Section boxes were numbered in the updated cognitive maps, after and in response to the interviews.

And important process for research based on humanistic inquiry is to subject elements of the research project to expert evaluation. In particular, I believe that there is substantial value in subjecting sets of cognitive maps from the case study to expert evaluation. Hirschman (1986) suggests that such research should be evaluated for:

1. Credibility
2. Transferability
3. Dependability
4. Confirmability

I have prepared a package of material for expert evaluation against these criteria, including:

- One Case Study Abstract summarising the original Harvard Business School Case Study source document plus additional insights and perspectives gained from the Case Study in my Dissertation
- One Cognitive Map of the raw mapped output from the NVivo Model Explorer application
- One Cognitive Map translated into a format similar to Nath & Newell (1998)
- One Cognitive Map updated from multiple interviews with subjects either directly named in the source documents, or with subjects who were actively involved in the application development referred to in the source documents.

External Auditor's Evaluation Form and Package (Cont'd)

I am asking you to review all of the material in the package against the question outlined below:

CREDIBILITY

1. Are the Cognitive Maps Believable When Reading And Reflecting On Material Presented?

TRANSFERABILITY

2. Are the processes described in the case abstract and cognitive maps useful metaphorical information that may help executives improve their sensemaking skills?

DEPENDABILITY

3. Does the process data presented in the case abstract and cognitive maps show consistency, fit, in what would be expected from reading one and comparing it to the other, and vice versa, including going from the abstract to the maps and back again?

External Auditor's Evaluation Form and Package (Cont'd)

CONFIRMABILITY

1. Do the data in the case abstract and maps support each other within and between the two sets of information?

_____
_____
_____
_____
_____

THICKNESS

2. Is deep knowledge, nuance filled knowledge, captured and apparent in the research procedure, the abstract, and cognitive maps?

_____
_____
_____
_____
_____

JUSTIFICATION

3. Has the research justified the use of revisiting prior case study environments/information and confirming and adding to the data reported by these cases and building theories in use?

_____
_____
_____
_____

4. Are you satisfied that new, unique and valuable contributions have been made by this study?

_____
_____
_____
_____
_____

External Auditor's Evaluation Form and Package (Cont'd)

FURTHER COMMENTS:

Are there any further comments you wish to make about this research project?

Thank you very much for your comments and support for this project.

Hugh Pattinson.

<u>**External Auditor's Evaluation Form and Package (Cont'd)**</u>

NetDynamics Case Study

Abstract

The NetDynamics case study extends an account of rapid application development with **decision systems analysis** (DSA) complemented by **cognitive mapping** of strategic thought.

The account explores decision-making for a start-up software house with a concept for application possessing potential to become a fundamental component of an emerging computer-mediated environment (CME – in the case study – the World Wide Web) - and a foundation for future enterprise e-commerce applications.

The case study provides insights into a group of three application developers working, testing and updating each other's applications, with a common vision to rapidly produce an advanced multimedia CME.

<u>External Auditor's Evaluation Form and Package (Cont'd)</u>

NetDynamics Case Study

Abstract (Continued)

The case study provides insights into a group of three application developers working, testing and updating each other's applications, with a common vision to rapidly produce an advanced multimedia CME.

Key findings from the case study highlight issues with rapid application development across a number of new application technologies; variations in actual relationships between developers working with each others' applications; roles for System Integrators as extended developers and marketers for small software houses; challenges with managing rapidly growing enterprise customer bases; and whether start-up software houses with enterprise-based applications can grow to a critical mass to provide enterprise customer support, or end up being acquired by larger vendors with established enterprise customer support infrastructure.

External Auditor's Evaluation Form and Package (Cont'd)

Exhibit 1 – Text Extract - NetDynamics Selection of Java

The single most important technical decision taken during this period was the choice of scripting language. By the end of December, the company had reviewed five alternatives: a proprietary language (nicknamed "E"), Visual Basic (a popular, but rather elementary language), C++ (another popular language, but with implementation drawbacks due to its compilation time, which would lengthen customers' development cycles), Python, and Java. The choice quickly came down to either "E" or Java, mainly because these would result in the maximum speed and simplicity with which a customer could develop their application. After much debate, the team finally chose Java.

What helped the team make such trade-off decisions was the clear original vision for the product developed in conjunction with lead users: Emphasize ease-of-use in application, and adopt an "open" platform philosophy (if a standard was emerging, that should take precedence over developing proprietary technology). However, while Java was ultimately chosen, the choice generated significant controversy, especially given its relative immaturity at the time and the limited information about its capabilities. "We knew Java was going to be big," recalled Malka, "but it was still only available as a 1.0 Beta version. This meant that the development tools which went along with it were either terribly buggy or nonexistent. Subsequently, we had to develop many of our own development tools."

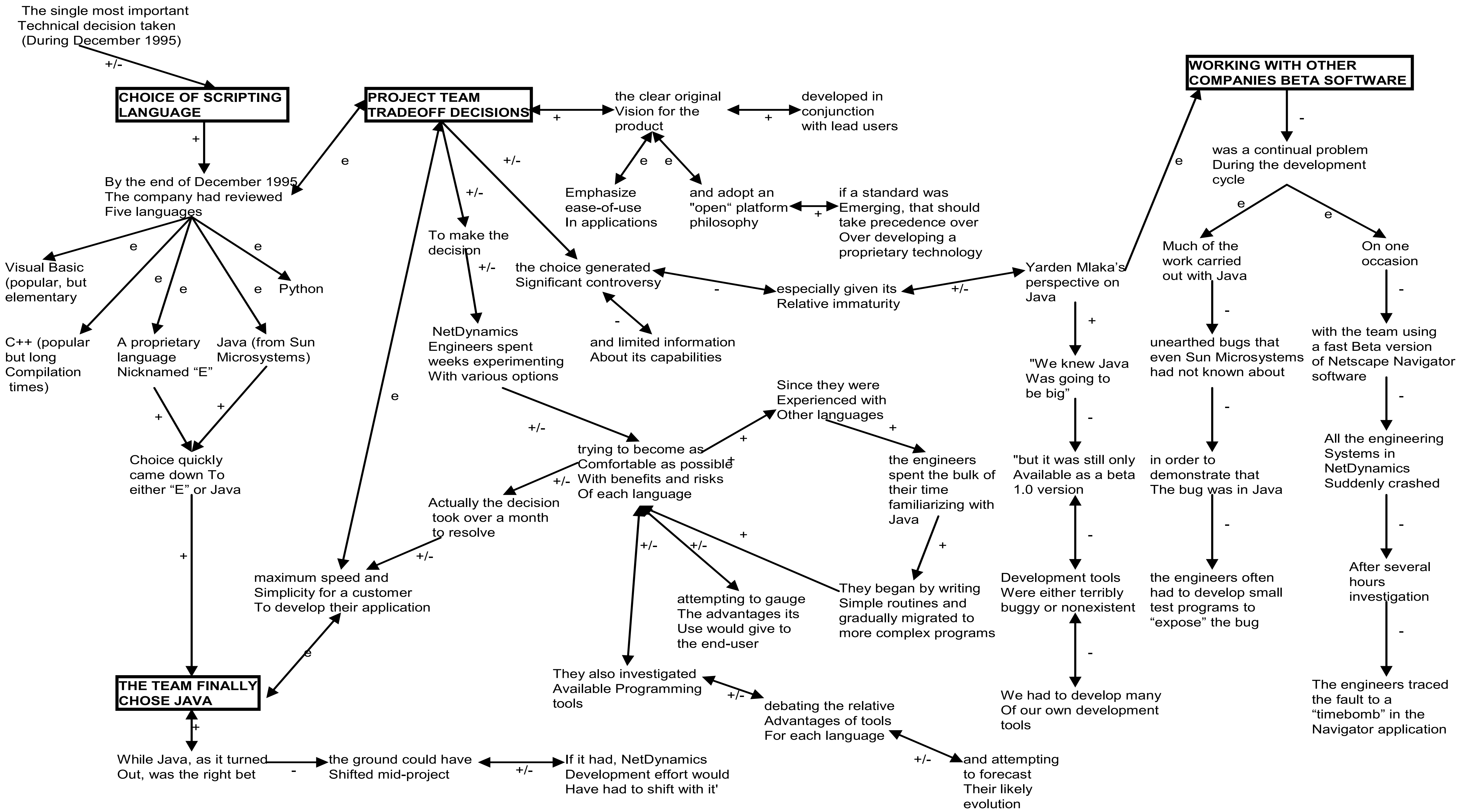
External Auditor’s Evaluation Form and Package (Cont’d)

Exhibit 1 – Text Extract - NetDynamics Selection of Java (Continued)

To make the decision, NetDynamics engineers spent weeks experimenting with various options, trying to become as comfortable as possible with the benefits and risks of each language. Since they were experienced with other approaches, the engineers spent the bulk of their time familiarizing themselves with Java. They began by writing simple routines, and gradually migrated to more complex programs, attempting to gauge the advantages its use would give to the end-user. They also investigated available programming tools, debating the relative advantages of each and attempting to forecast their likely evolution. With such detailed experimentation, the decision took over a month to resolve.

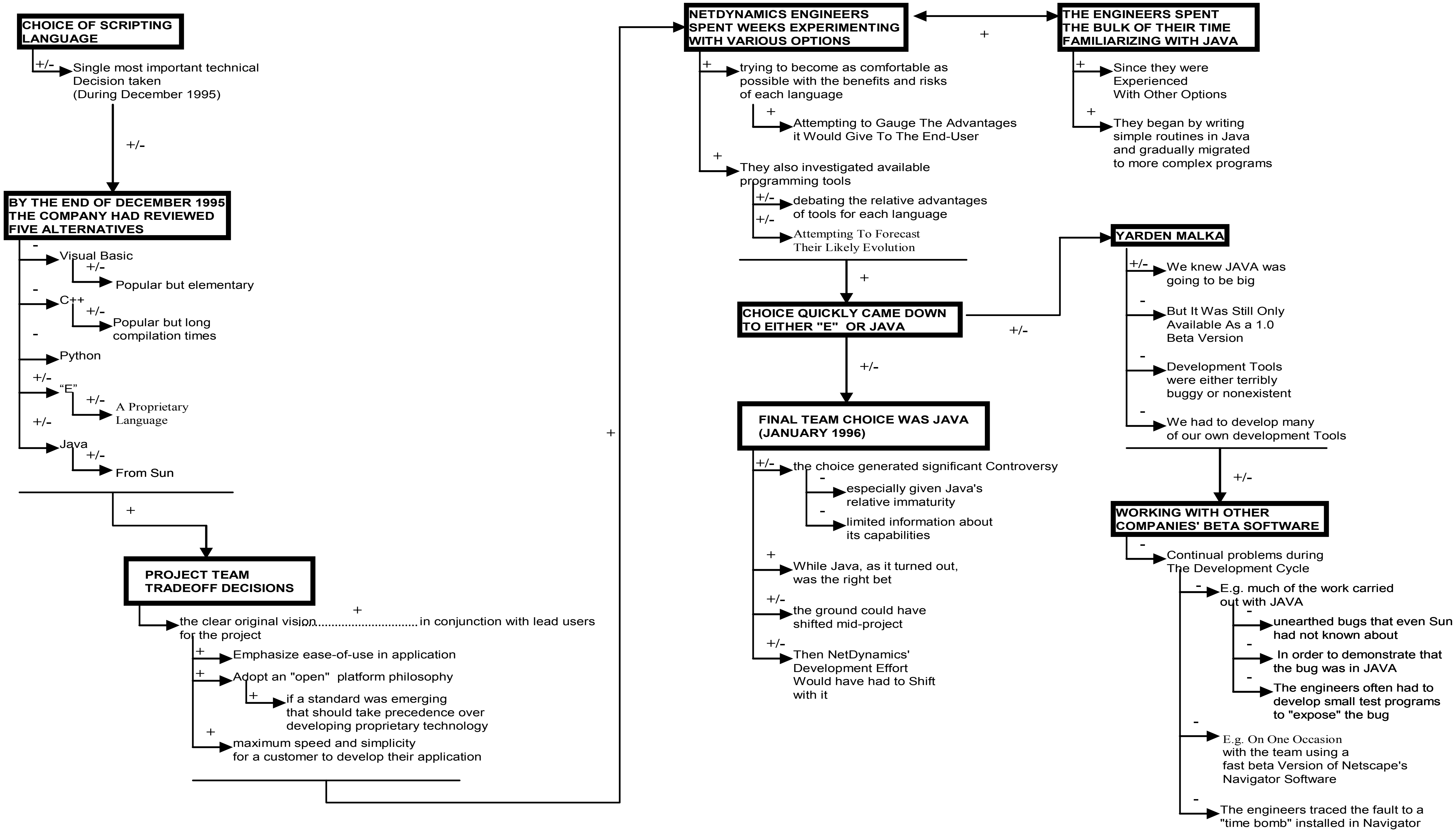
While Java, as it turned out, was the right bet, the ground could have shifted mid-project. If it had, NetDynamics' development effort would have had to shift with it.
Working with other companies' beta software was a continual problem during the development cycle. For example, much of the work carried out with Java unearthed bugs that even Sun Microsystems had not known about. However, in order to demonstrate that the bug was in Java, not their own code, the engineers often had to develop small test programs to "expose" the bug. On one occasion, with the team using a fast beta version of Netscape's Navigator software, all the engineering systems at NetDynamics suddenly crashed. After several hours investigation, the engineers traced the fault to a "timebomb" installed in Navigator, to stop unauthorized use past a certain date.

Extracted from Iansiti & MacCormack (1996,7)



External Auditor's Evaluation Form and Package (Cont'd)

Figure 1 Cognitive Map (NVivo Model) - Selection of Java (November-December 1995)
Created from data in Iansiti & MacCormack (1996)



External Auditor's Evaluation Form and Package (Cont'd)

Figure 2 Cognitive Map - Selection of Java (November-December 1995)
Created from data in Iansiti & MacCormack (1996)

Exhibit 3.1: External Auditor's Evaluation Form and Package (Cont'd)

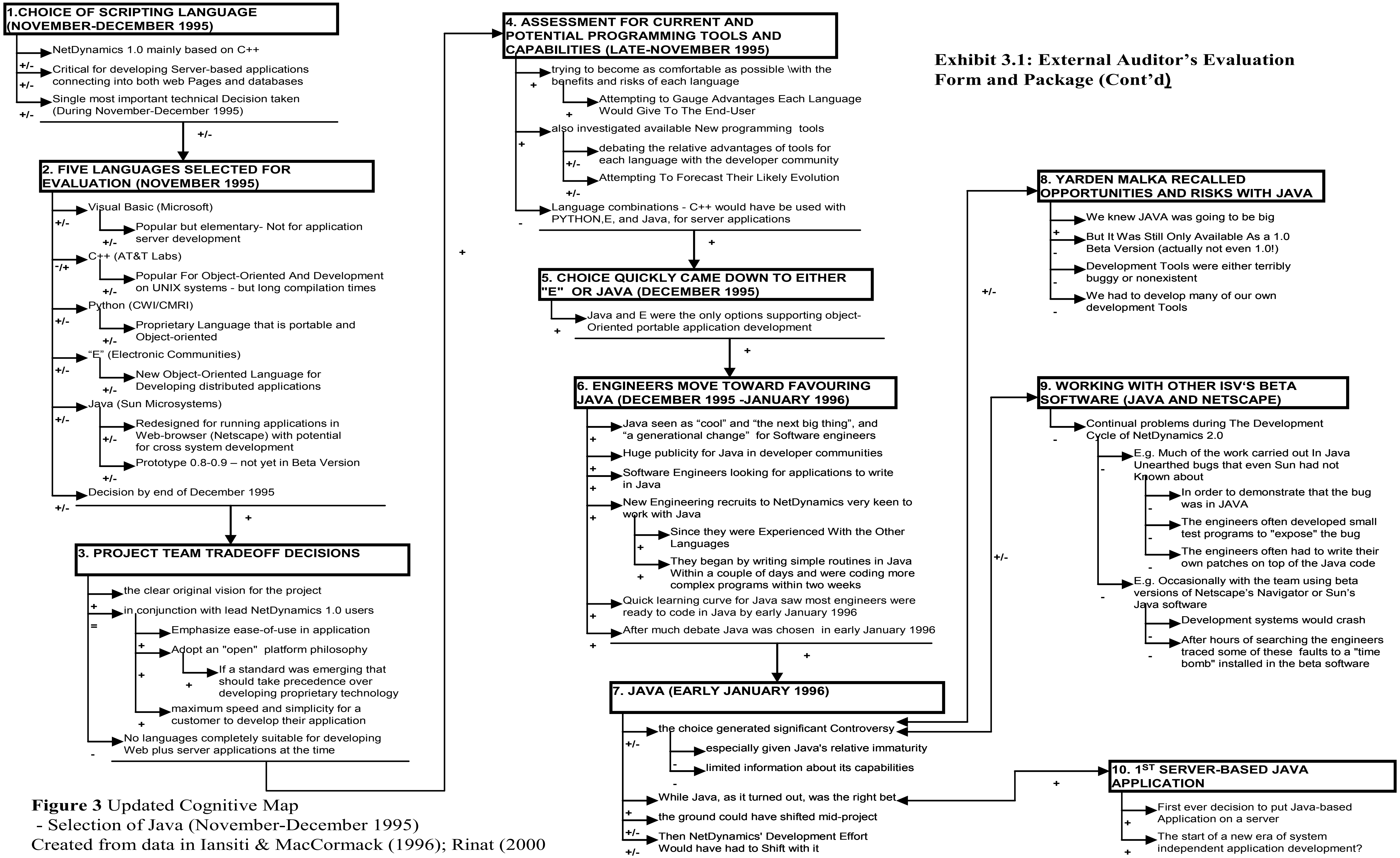


Figure 3 Updated Cognitive Map
- Selection of Java (November-December 1995)
Created from data in Iansiti & MacCormack (1996); Rinat (2000 a&b); Ebbs (2002 a&b)

External Auditor's Evaluation Form (EAEF)

References (I)

Biemans, Wim G. (1989), *Developing Innovations Within Networks*. Groningen: University of Groningen Press.

Biemans, Wim G. (2003), "A Picture Paints A Thousand Numbers: A Critical Look At B2B Product Development Research," *Journal Of Business & Industrial Marketing*, 18 (6/7), 514-28.

Ebbs, Kenneth (2002), Personal Interview (Telephone) with Hugh Pattinson, Sydney/Santa Clara CA. (7 May).

Ebbs, Kenneth (2002), Personal Interview (Telephone) with Hugh Pattinson, Sydney/Santa Clara CA. (10 May)

Gummesson, Evert (2003), "All Research Is Interpretative!" *Journal Of Business & Industrial Marketing*, 18 (6/7), 482-92.

Hirschman, Elizabeth C. (1986), "Humanistic Inquiry in Marketing Research: Philosophy, Method, and Criteria," *Journal of Marketing Research*, 23 (August), 237-49.

Howard, John A., James Hulbert, and John U. Farley (1975), "Organization Analysis And Information Systems Design: A Decision-Process Perspective," *Journal Of Business Research*, 3 (2 (April)), 133-48.

External Auditor's Evaluation Form (EAEF)

References (II)

Hulbert, James (2003), "Organizational Analysis And Information Systems Design: A Road Revisited," *Journal Of Business & Industrial Marketing*, 18 (6/7), 509-13.

Huff, Anne Sigismund (1990), *Mapping Strategic Thought*. John Wiley And Sons, New York.

Huff, Anne Sigismund, Vijaya Narapeddy, and Karen E. Fletcher (1990), "Coding the Causal Association of Concepts," in *Mapping Strategic Thought*, Anne Sigismund Huff, Ed. New York: John Wiley & Sons.

Huff, Anne Sigismund, James Oran Huff, with Pamela S. Barr (2000), *When Firms Change Direction*. Oxford University Press, New York.

Huff, Anne Sigismund. Mark Jenkins (Eds.) (2002), *Mapping Strategic Knowledge*. Sage, London.

Iansiti, Marco and Alan MacCormack (1996), "Living On Internet Time: Product Development at Netscape, Yahoo!, Net Dynamics and Microsoft", Harvard Business School, Boston, 9-697-052

Iansiti, Marco, Alan MacCormack (1996), "Living On Internet Time: Product Development at Netscape, Yahoo!, Net Dynamics and Microsoft", Harvard Business School, Boston, 9-697-052

Nath, Deepika and Stephanie E. Newell (1998), "Organizational Responses to a Hypercompetitive Environment: A Case Study of Pepsi Canada," *Journal Of Business Research*, 41, 41-48.

External Auditor's Evaluation Form (EAEF)

References (III)

Rinat, Zack (2002), Personal Interview (Face-To-Face) with Hugh Pattinson, South San Francisco CA. (20 February).

Rinat, Zack (2002), Personal Interview (Telephone) with Hugh Pattinson, Sydney/South San Francisco CA. (25 April).

Woodside, Arch G., Elizabeth J. Wilson (2000), "Constructing Thick Descriptions of Marketers' and Buyers' Decision-Processes in Business-to-Business Relationships," *Journal Of Business and Industrial Marketing*, 15 (4/5), 354-69.

Woodside, Arch G., Elizabeth J. Wilson (2003), "Case Study Research Methods For Theory Building," *Journal Of Business & Industrial Marketing*, 18 (6/7), 493-508.

References

Ambrosini, Veronique and Cliff Bowman (2002), "Mapping Successful Organisational Routines," in *Mapping Strategic Knowledge*, Anne Sigismund Huff, Mark Jenkins, Ed. London: Sage Publications.

Armour, Phillip G. (2000), "The Case for a New Business Model – Is Software a Product or a Medium?" *Communications of the ACM,* 43 8(August) 19-22

Arnold, Stephen J. and Eileen Fischer (1994), "Hermeneutics and Consumer Research," *Journal of Consumer Research*, 21 (June), 55-70.

Arthur, W. Brian (2002), "Is The Information Revolution Dead?" *Business 2.0*, 3 (3), 65-72.

Austin, Robert D. (1998), "Trilogy (A)", Harvard Business School, Boston, 9-699-034

Axelrod, R. (1976), *Structure Of Decision*. Princeton NJ: Princeton University Press.

Banxia Software (2002), Decision Explorer, V 3.2, Windows XP, www.banxia.com

Bazeley, Patricia and Lyn Richards (2000), *The NVivo Qualitative Project Book*. London: Sage Publications.

Berners-Lee, Tim (1999), *Weaving The Web*, London: Orion Business Books.

Bettis, Richard A. and C.K. Prahalad (1995), "The Dominant Logic: Retrospective And Extension," *Strategic Management Journal*, 16 (January), 5-14.

Biemans, Wim G. (1991), "User and Third-Party Involvement in developing Medical Equipment Innovations," *Technovation*, 11(3), 163-82.

Biemans, Wim G. (2003), "A Picture Paints a Thousand Numbers: A Critical Look at B2B Product Development Research," *Journal of Business & Industrial Marketing*, 18 (6/7), 514-28.

Brinberg, David and Elizabeth C. Hirschman (1986), "Multiple Orientations for the Conduct of Marketing Research: An Analysis of the Academic/Practitioner Distinction," *Journal of Marketing*, 50 (4), 161-173.

Brookes, Cyril H. P., Phillip J. Grouse, D., Ross Jeffery, and Michael j. Lawrence (1982), *Information Systems Design*, Sydney: Prentice-Hall of Australia.

Brooks, F. P. (1975), *The Mythical Man Month: Essays on Software Engineering,* Addison-Wesley, Reading Ma.

Burawoy, Michael, Alice Burton, Ann Arnett Ferguson, Kathryn J. Fox, Joshua Gamson, Nadine Gartrell, Leslie Hurst, Charles Kurtzman, Leslie Salzinger, Joseph Schiffman, and Shiori Uri (1991), *Ethnography Unbound: Power and Resistance in the Modern Metropolis*, Berkeley CA: University of California Press.

Capon, Noel and James Hulbert (1975), "Decision Systems Analysis In Industrial Marketing," *Industrial Marketing Management*, 4, 143-60.

Chandra, Mala (2001), Personal Interview (Telephone) with Hugh Pattinson, Sydney/Redwood Shores, California (13 December).

Chesbrough, Henry W. (2003a), "The Era of Open Innovation," *MIT Sloan Management Review*, 44 (3), 35-41.

Chesbrough, Henry W. (2003b), *Open Innovation: The New Imperative for Creating and Profiting from Technology*, Boston: Harvard Business School Press.

Christensen, Clayton M. (1997), *The Innovator's Dilemma: When New Technologies Cause Great Firms To Fail*, Boston: Harvard Business School Press.

Christensen, Clayton M., and Michael E. Raynor (2003), *The Innovator's Solution: Creating And Sustaining Successful Growth*, Boston: Harvard Business School Press.

Clark, K, B. (1985), "The Interaction of Design Hierarchies and Market Concepts in Technological Evolution," *Research Policy,* 14, 235-51.

Clark, K, B. and Takahiro Fujimoto (1991), *Product Development Performance.* Boston MA: Harvard Business School Press.

Clemen, Robert T. and Terence Reilly (2001), *Making Hard Decisions With DecisionTools - 2nd Edition* (2nd Rev Ed.), Pacific Grove CA.: Duxbury.

Clausing, D. (1993), *Total Quality Development: A Step by Step Guide to World Class Concurrent Engineering,* New York: ASME Press

CNRI (2003), "Corporation for National Research Initiatives", http://www.cnri.reston.va.us/ , Viewed on 1 November 2003

Cohen, Michael D., James G. March, and Johan P. Olsen (1972), "A Garbage Can Model of Organisation Choice," *Administrative Science Quarterly*, 17, 1-25.

Cooper, R. G. (1983), "A Process Model for Industrial New Product Development," *IEEE Transactions on Engineering Management,* EM-30 (1) 128-152

Copernic (2005), “Copernic Summariser”, http://www.copernic.com/en/products/summarizer/index.html,.

Cusumano, Michael A. and David B. Yoffie (2000), *Competing On Internet Time: Lessons From Netscape and Its Battle With Microsoft* (Touchstone Edition), New York: Touchstone.

CWI (2003), "Stitching Mathematisch Centrum", http://www.cwi.nl/ , Viewed on 1 November.

Cyert, Richard M., Herbert A. Simon, and Donald B. Trow (1956), "Observation of a Business Decision," *Journal of Business*, 29(4), 237-48.

Danneels, Erwin (2002), "The Dynamics of Product Innovation And Firm Competences," *Strategic Management Journal*, 23, 1095-121.

DeLacey, Brian and Dorothy Leonard (2001), "Zaplet, Inc. (A)", Harvard Business School, Boston, 9-601-165

Dodge, Martin and Rob Kitchin (2001), *Mapping Cyberspace*, London: Routledge.

Dodge, Martin and Rob Kitchin (2001), *Atlas of Cyberspace*, Harlow, UK: Pearson Education/Addison Wesley.

Dougherty, Deborah (1989), "Interpretive Barriers to Successful Product Innovation,” *Organization Science*, 3(May) 179-201.

Dougherty, Deborah (1992), "A Practice-Centered Model Of Organisational Renewal Through Product Innovation," *Strategic Management Journal*, 13, 77-92.

Ebbs, Kenneth (2002a), Personal Interview (Telephone) with Hugh Pattinson, Sydney/Santa Clara CA. (7 May).

Ebbs, Kenneth (2002b), Personal Interview (Telephone) with Hugh Pattinson, Sydney/Santa Clara CA. (10 May)

Eden, Colin and Fran Ackerman (2002), "A Mapping Framework For Strategy Making," in *Mapping Strategic Knowledge*, Anne Sigismund Huff, Mark Jenkins, Ed. London: Sage Publications.

Eisenhardt, Kathleen M. and B. Tabrizi (1995), “Accelerating Adaptive Processes – Product Innovation in the Global Computer Industry”, *Administration Science Quarterly,* 40(Mar) 84-110

Faleiro, Brian, Dana Porter, Siddarth Rastogi, Vitlay Shub, Christine Stokes, Lanchi Venator, Christopher L. Tucci (2003), "Hotmail: Free Mail For Sale," in *Internet*

Business Models And Strategies: 2nd Edition, Allan Afuah, Christopher L. Tucci, Ed. New York: McGraw-Hill.

Fletcher, Karen E. and Anne Sigismund Huff (1990), "Strategic Argument Mapping: A Study of Strategy Reformulation at AT&T," in *Mapping Strategic Thought*, Anne Sigismund Huff, Ed. New York: John Wiley & Sons.

Fletcher, Karen E. and Anne Sigismund Huff (1990), "Argument Mapping," in *Mapping Strategic Thought*, Anne Sigismund Huff, Ed. New York: John Wiley & Sons.

Franke, David (2002), Personal Interview (Face-to-Face) with Hugh Pattinson, Austin Texas (13 February).

Franke, David (2002), Personal Interview (Telephone) with Hugh Pattinson, Sydney/Austin Texas (17 April).

Fraunhofer Institute – IITB (2004), "Wissenschaliche Hilskrafte/Praktikantinnen: Intelligente Softwareagenten, Semantic Web, UML, Java, J2EE, CORBA", http://www.iitb.fraunhofer.de/ , Viewed 20 February 2005 (Note: there is no English translation for this site).

Freeman, Christopher and Francisco Louca (2002), *As Time Goes By: From The Industrial Revolution to The Information Revolution*, Oxford: Oxford University Press.

Gainey, Mark, Michael Wolfe (2002), Personal Interview (Face-to-Face) with Hugh Pattinson, Menlo Park, California (21 February).

Gainey, Mark, (2002), Personal Interview (Telephone) with Hugh Pattinson, Sydney/Los Altos, California (23 April).

Geertz, Clifford (1973), "Chapter 1: Thick Description: Toward an Interpretative Theory of Culture," in *The Interpretation Of Cultures: Selected Essays*, Clifford Geertz, Ed. New York: Basic Books.

Gergen, Kenneth J., and Mary M. Gergen (1986), "Narrative Form and the Construction of Psychological Science," in *Narrative Psychology: The Storied Nature of Human Conduct*, Theodore R. Sarbin, ed., New York: Praeger, 22-44.

Giddens, A. (1984), *The Constitution of Society: Outline of the Theory of Structuration*, Berkeley CA: University of California Press.

Gladwell, Malcolm (2002), *The Tipping Point: How Little Things Can Make a Big Difference*, Boston: Back Bay Books.

Goetz, Thomas (2003), "Open Source Everywhere," *Wired*, 11 (11), 158-67 & 208-11.

Hackley, Chris (2003), “Divergent Representational Practices in Advertising and Consumer Research: Some Thoughts on Integration,” *Qualitative Market Research: An International Journal,* 6 (3), 175-183

Hall, Roger (1976), “A System Pathology of an Organisation: the Rise and Fall of the Old Saturday Evening Post, *Administrative Science Quarterly*, 21 (2), 185-211.

Hawryszkiewycz, Igor T. (1998), *Introduction to Systems Analysis and Design: 4th Edition*, Sydney: Prentice Hall.

Hendry, John (2000), “Strategic Decision-making, Discourse, and Strategy as Social Practice,” *Journal of Management Studies,* 37 (7), 955-973.

Hicks, Terry (2002a), Personal Interview (Face-to-Face) with Hugh Pattinson, Mountain View California (21 February).

Hicks, Terry (2002b), Personal Interview (Telephone) with Hugh Pattinson, Sydney/Mountain View California (8 May).

Hickson, David J., Richard J. Butler, David Cray, Geoffrey R. Mallory, and David C. Wilson (1986), *Top Decisions: Strategic Decision-making in Organisations*, San Francisco: Jossey-Bass.

Hirschman, Elizabeth C. (1986), "Humanistic Inquiry in Marketing Research: Philosophy, Method, and Criteria," *Journal of Marketing Research*, 23 (August), 237-49.

Hodgkinson, Gerard P., A. John Maule (2002), "The Individual in the Strategy Process," in *Mapping Strategic Knowledge*, Anne Sigismund Huff, Mark Jenkins, Ed. London: Sage Publications.

Hoffmann, Donna L. and Thomas P. Novak (1996), "Marketing Hypermedia Computer Mediated Environments: Conceptual Foundations," *Journal Of Marketing*, 60 (July 1996), 50-68.

Hohn, Jessica (2001), "KANA Solidifies eCRM Leadership with Launch of Fully Integrated iCARE Suite; Exceeds Integration Milestones", http://www.kana.com/newsevents/pressrelease/default.aspx , Viewed on 7 November 2003

Holbrook, M.B. and J. O'Shaughnessy (1988), "On the Scientific Status of Consumer Research and the Need of for an Interpretative Approach to Studying Consumer Behaviour," *Journal of Consumer Research,* 15 (December), 398-402

Howard, John A. and William M. Morgenroth (1968), "Information Processing Model of Executive Decision," *Management Science*, 14 (7), 416-28.

Howard, John A., James Hulbert, and John U. Farley (1975), "Organisation Analysis and Information Systems Design: A Decision-Process Perspective," *Journal Of Business Research*, 3 (2), 133-48.

Huff, Anne Sigismund (1990), *Mapping Strategic Thought*, New York: John Wiley & Sons.

Huff, Anne Sigismund (1990a), "Mapping Strategic Thought," in *Mapping Strategic Thought*, Anne Sigismund Huff, Ed. New York: John Wiley & Sons.

Huff, Anne Sigismund (1990b), "Bias and Sensemaking in Good Times and Bad," in *Mapping Strategic Thought*, Anne Sigismund Huff, Ed. New York: John Wiley & Sons.

Huff, Anne Sigismund, James Oran Huff, with Pamela S. Barr (2000), *When Firms Change Direction*. New York: Oxford University Press.

Huff, Anne Sigismund, James Oran Huff, with Pamela S. Barr, J. Larry Stimpert (2000), "The Structuration Of Industries," in *When Firms Change Direction*, Anne Sigismund Huff, James Oran Huff, with Pamela S. Barr, Ed. New York: Oxford University Press.

Huff, Anne Sigismund and Mark Jenkins Eds. (2002), *Mapping Strategic Knowledge*, London: Sage.

Huff, Anne Sigismund, Vijaya Narapeddy, and Karen E. Fletcher (1990), "Coding the Causal Association of Concepts," in *Mapping Strategic Thought*, Anne Sigismund Huff, Ed. New York: John Wiley & Sons.

Hulbert, James (2003), "Organisational Analysis And Information Systems Design: A Road Revisited," *Journal Of Business & Industrial Marketing*, 18 (6/7), 509-13.

Hulbert, James, John U. Farley, and John A. Howard (1972), "Information Processing and Decision-making in Marketing Organisations," *Journal of Marketing Research*, 9 (February), 75-7.

Hyams, Chris (2002), Personal Interview (Face-to-Face) with Hugh Pattinson, Austin Texas (13 February).

Hyams, Chris (2002), Personal Interview (Telephone) with Hugh Pattinson, Sydney/Austin Texas (3 May).

Iansiti, Marco (1995a), "Shooting the Rapids: Managing Product Development in Turbulent Environments," *California Management Review*, 38 (Fall), 37-58.

Iansiti, Marco (1995b), "Technology Integration: Managing Technological Evolution in a Complex Environment, *Research Policy*, 24, 521-42

Iansiti, Marco (1998), *Technology Integration: Making Critical Choices in a Dynamic World*. Boston: Harvard Business School Press.

Iansiti, Marco and Alan MacCormack (1996), "Living On Internet Time: Product Development at Netscape, Yahoo!, Net Dynamics and Microsoft", Harvard Business School, Boston, 9-697-052.

Iansiti, Marco and Alan MacCormack (1998), "Product Development on the Internet," in *Sense and Respond: Capturing Value in the Network Era*, Stephen P. and Richard L. Nolan Bradley, Ed. Vol. Ch.8. Boston: Harvard Business School Press

Intuit, Corporation (2002), "Intuit Unveils New QuickBooks 2003 Financial Product Line." Mountain View.

Intuit, Corporation (2003), "Intuit Inc Annual Report 2003", Intuit Inc., Mountain View Ca.

Intuit, Corporation (2003), "Intuit Developer Network", http://www.developer.intuit.com/ , Viewed on 26 November.

James, Gail E. (2000), "Industry-Level Learning and the Social Construction of Recipes," in *When Firms Change Direction*, Anne Sigismund Huff, James Oran Huff, with Pamela S. Barr, Ed. New York: Oxford University Press.

Jasinski, Dale W. and Anne Sigismund Huff (2002), "Using a Knowledge-Based System to Study Strategic Options," in *Mapping Strategic Knowledge*, Anne Sigismund Huff, Mark Jenkins, Ed. London: Sage Publications.

Kidder, Tracy (1981), *The Soul Of A New Machine*, New York: Avon Books.

Kirsner, Scott (2002), "How Do You Spell Value?" *Newsweek* (10 June) Vol. 139

Klein, Gary (1999), *Sources of Power: How People Make Decisions*, Cambridge MA: The MIT Press.

Klein, Harald (2003), "Text Analysis Info Page and Website", http://www.textanalysis.info , Viewed: 18 December 2003.

Kochikar, Purnima, Rajiv Lal (2000), "Intuit QuickBooks", Harvard Business School, Boston, 9-501-054.

Krishnan, V. and Karl T Ulrich (2001), "Product Development Decisions: A Review of the Literature," *Management Science* 47 1(January) 1-21.

Lakoff, G., M. Johnson (1980), *Metaphors We Live By,* Chicago University Press, Chicago.

Langley, Ann, Henry Mintzberg, Patricia Pitcher, Elizabeth Posada, Jan Saint-Macary (1995), "Opening Up Decision-making: The View from the Black Stool," *Organisation Science*, 6 (3), 269-79.

Lyles, M. A. (1990), "A Research Agenda for Strategic Management in the 1990s," *Journal Of Management Studies*, 27 (4), 363-75.

Maani, Kambiz E., and Vandana Maharaj (2004), "Links between Systems Thinking and Complex Decision-making," *System Dynamics Review*, 20 (1), 21-48.

Medlin, Christopher John (2004), "Interaction in business relationships: A time perspective," *Industrial Marketing Management*, 33 (3), 185-93.

MacCormack, Alan, and Kerry Herman (2000), "Red Hat and The Linux Revolution", Harvard Business School, Boston, 9-600-009

McCracken, G. (1988). *The Long Interview*, Newbury Park: Sage, USA.

McNamara, Robert S., James G. Blight, Robert K. Brigham, Thomas J. Biersteker, Herbert Y. Schandler (1999), *Argument without End*: In Search of Answers to the Vietnam Tragedy, New York: Public Affairs.

Mandel, Marc and Robert D. Austin (2000), "Trilogy (B)", Harvard Business School, Boston, 9-600-123

Microsoft Corporation (2002a), MS-EXCEL, V XP (2002), Windows XP, www.microsoft.com

Microsoft Corporation (2003a), MS-EXCEL, V 2003, Windows XP, www.microsoft.com

Microsoft, Corporation (2002b), PowerPoint, V XP (2002), Windows XP, www.microsoft.com

Microsoft, Corporation (2003b), PowerPoint, V 2003, www.microsoft.com

Mintzberg, Henry (1979), "An Emerging Strategy of 'Direct' Research," *Administrative Science Quarterly*, 24 (4), 582-90.

Mintzberg, Henry (1990), "The Design School: Reconsidering the Basic Premises of Strategic Management," *Strategic Management Journal*, 11, 171-295.

Mintzberg, Henry, B. Ahlstrand, J. Lampel (1998), *Strategy Safari*, New York: Free Press.

Mintzberg, Henry, Duru Rasinghani, and Andre Theoret (1976), "The Structure of "Unstructured" Decision Processes," *Administrative Science Quarterly*, 21 (June), 246-75.

Mohrmann, G.P., C. J. Stewart, D.J. Ochs (Eds) (1973), *Explorations in Rhetorical Criticism,* Pennsylvania State University Press, Philadelphia

Morgenroth, William M. (1964), "A Method for Understanding Price Determinants," *Journal of Marketing Research*, 2 (August), 17-26.

Nath, Deepika, and Stephanie E. Newell (1998), "Organisational Responses to a Hypercompetitive Environment: A Case study of Pepsi Canada," *Journal Of Business Research*, 41, 41-48.

Pattinson, Hugh M., Arch G. Woodside, and Kenneth E. Miller (2002), "Combining Decision Systems Analysis And Cause Mapping Of Strategic Thought: Making Sense Of The Dynamics In B2B Strategies," in Proceedings of The Center For Business And Industrial Marketing (CBIM) - Eighth Annual Academic Workshop, Michael K. Rich (Ed.). Atlanta, GA.

Pattinson, Hugh M. (2003), ""Click Here For Your Recommendations": An Exploration of Online Personalization Applications," Presented at World Marketing Congress, Perth June 2003, Work-In-Progress.

Perez, Carlotta (2002), *Technological Revolutions and Financial Capital*, Cheltenham UK: Edward Elgar.

Phillips, N., J.L. Brown (1993), “Analyzing Communication in and Around Organisations: a Critical Hermeneutic Approach,” *Academy of Management Journal,* 36, 1547-76.

Prahalad, C. K., and Richard A. Bettis (1986), "The Dominant Logic: a New Linkage Between Diversity and Performance," *Strategic Management Journal*, 7 (6), 485-501.

PYTHON (2003), "History of PYTHON", http://www.python.org/dev/doc/devel/lib/node636.html , Viewed on 3 November.

QSR International (2002a), NVivo, V 1.3, Windows XP, www.qsr.com.au

QSR International (2002b), NVivo, V 1.3, Windows XP, www.qsr.com.au

ResearchWare (2003), HyperResearch, V 2.6, Windows XP, http://www.researchware.com

Rinat, Zack (2002a), Personal Interview (Face-to-Face) with Hugh Pattinson, South San Francisco California (20 February).

Rinat, Zack (2002b), Personal Interview (Telephone) with Hugh Pattinson, Sydney/South San Francisco California (25 April).

Ritter, Thomas, Ian F. Wilkinson, Wesley J. Johnston (2004), "Managing in Complex Business Networks," *Industrial Marketing Management*, 33 (3), 175-83.

Rivken, Gary (2003), "Leader of the Free World," *Wired*, 11 (11), 152-57 & 206-08.

Roberts, David (2002), Personal Interview (Face-to-Face) with Hugh Pattinson, Redwood Shores, California (20 February).

Rossiter, John R. Rossiter (2001), "What is Marketing Knowledge? Stage I: Forms of Marketing Knowledge," *Marketing Theory*, 1 (September), 9-26.

Rughase, Olaf G. (2002), "Linking Content to Process," in *Mapping Strategic Knowledge*, Anne Sigismund Huff, Mark Jenkins, Ed. London: Sage Publications.

Sarvary, Miklos (2000), "Kana Communications", Harvard Business School, Boston, 9-501-003.

Sawhney, Mohanbir (2003), "How to Keep Your Customers Satisfied ; To learn if end users are happy, you have to ask the right people- and the right questions.," *CIO Magazine*, 16 (14), 1.

Schank, Roger C. (1998), *Tell Me A Story: Narrative And Intelligence*, Evanston Il.: Northwestern University Press.

Senge, Peter (1990), *The Fifth Discipline: the Art and Practice of the Learning Organisation* (1st ed.), New York: Currency Doubleday.

Shane Scott A., and Karl T. Ulrich (2004), "50th Anniversary Article: Technological Innovation, Product Development, and Entrepreneurship in *Management Science*," *Management Science,* 50 2(February) 133-144

Stubbart, Charles I. and Arkalgud Ramaprasad (1988), "Probing Two Chief Executives' Schematic Knowledge of the U.S. Steel Industry Using Cognitive Maps," in *Advances In Strategic Management*, Vol. 5. Greenwich CN.: JAI Press.

SUN (2003), "NetDynamics - Awards", http://www.sun.com/products/netdynamics/awards.html , Viewed on 3 November

TACT (1997), http://www.indiana.edu/~letrs/help-services/QuickGuides/about-tact.html.

Thompson, Craig, Howard R. Pollio, and William B. Locander (1994), "The Spoken and the Unspoken: A Hermeneutic Approach to Understanding the Cultural Viewpoints that Underlie Consumers' Expressed Meanings," *Journal of Consumer Research*, 21 (December), 432-452.

Thompson, Craig J. (1997), "Interpreting Consumers: A Hermeneutical Framework for Deriving Marketing Insights for the Texts of Consumers' Consumption Stories," *Journal of Marketing Research*, 34 (November), 438-456.

Torvalds, Linus and David Diamond (2001), *Just For Fun: The Story Of An Accidental Revolutionary*, New York: Texere.

Treacey, Michael and Fred Wiersma (1996), *The Discipline of Market Leaders*. London: HarperCollins.

Trilogy (2003a), Website, notes on testimonials and examples of Trilogy applications: Ford (http://www.trilogy.com/Sections/Extras/Downloads/Forrester.pdf); British Airways (http://www.trilogy.com/Sections/News/Press_Releases/PR_2003-05-12_0194.cfm); Prudential Insurance Company of America (http://www.trilogy.com/Sections/News/Press_Releases/PR_2000-010-10_0151.cfm), www.trilogy.com , all viewed 23 July.

Trilogy (2003b) Website, comments from Jack Welch (http://www.trilogy.com/Sections/News/Business_Press.cfm), plus extract from CIO (http://www.cio.com/archive/050103/netgains.html - linked to Trilogy Website) , and NewsWeek (http://www.trilogy.com/Sections/News/?NewsID=AR_2002-06-10_0188). www.trilogy.com , all viewed 23 July.

Trilogy Corporation (2003), "Introducing The Trilogy Fast Cycle Time Methodology: Delivering E-Business Value In Internet Time", http://www.trilogy.com/Sections/Extras/Downloads/FCT.pdf , Viewed on 28 November 2003

Tuomi, Illka (2003), *Networks of Innovation: Change and Meaning in the Age of the Internet*, Oxford: Oxford Press.

University of Toronto, (1993), TACT, V 2.1, MS-Dos (Runs In MS-Dos Prompt in Windows), www.chass.utoronto.ca

Veritape Ltd. (2002), Veritape, V 1.3, Windows XP, www.veritape.com

Veritape Ltd. (2003), Veritape, V 2.0, Windows XP, www.veritape.com

Villanueva, Tim (2002a), Personal Interview (Face-to-Face) with Hugh Pattinson, Sunnyvale California (18 February).

Villanueva, Tim (2002b), Personal Interview (Telephone) with Hugh Pattinson, Sydney/Sunnyvale California (9 May).

Von Hippel, Eric (1986), "Lead users: A source of novel product concepts, *Management Science.* 32 (July) 791-805

Von Hippel, Eric (2001), "Innovation By User Communities: Learning From Open-Source Software," *MIT Sloan Management Review*, 42 (4), 82-86.

Vyas, Niren, and Arch G. Woodside (1984), "An Inductive Model of Industrial Supplier Choice Processes," *Journal Of Marketing*, 48 (Winter), 30-45.

W3C Worldwide Consortium (2004), "Semantic Web", http://www.w3.org/2001/sw/, viewed on 20 February 2005.

Walsh, J. P. (1995), "Managerial and Organisational Cognition: Notes From a Trip Down Memory Lane," *Organisation Science*, 6 (3), 280-321.

Weick, Karl E. (1990), "Introduction: Cartographic Myths In Organisations," in *Mapping Strategic Thought*, Anne Sigismund Huff, Ed. New York: John Wiley And Sons.

Weick, Karl E. (1995), *Sensemaking in Organisations*, Thousand Oaks CA: Sage.

Weick, Karl E. (2001), *Making Sense of the Organisation,* Oxford: Blackwell Publishing.

Wheelwright, S., and K. B. Clark (1992), *Revolutionizing Product Development,* New York: The Free Pres

Whyte, William F. (1943), *Street Corner Society*, Chicago: Chicago University Press.

Wolfe, Michael (2002), Personal Interview (Telephone) with Hugh Pattinson, Menlo Park, California (20 April).

Woodside, Arch G. (2003), “Middle-Range Theory Construction of the Dynamics of Organisational Marketing-Buying Behaviour” *Journal of Business & Industrial Marketing*, 18 (4/5), 309-335.

Woodside, Arch G. (1996), "Rejecting Superior New Technologies," in *Marketing in Evolution: Essays in Honour of Michael J. Baker*, Susan A. and Neil Hood Shaw, Ed. London: Macmillan Press.

Woodside, Arch G. (1994), "Network Anatomy Of Industrial Marketing And Purchasing Of New Manufacturing Technologies," *Journal Of Business & Industrial Marketing*, 9 (3), 52-63.

Woodside, Arch G., Elizabeth J. Wilson (2003), "Case study Research Methods For Theory Building," *Journal Of Business & Industrial Marketing*, 18 (6/7), 493-508.

Woodside, Arch G. and David M. Samuel (1981), "Observations of Centralised Corporate Procurement," *Industrial Marketing Management*, 10, 191-205.

Woodside, Arch G. and Elizabeth J. Wilson (2000), "Constructing Thick Descriptions of Marketers' and Buyers' Decision-Processes in Business-to-Business Relationships," *Journal Of Business and Industrial Marketing*, 15 (4/5), 354-69.

Woodside, Arch G. and Glenn Voss (1999), "Causal Modelling Innovation, Manufacturing, Diffusion and Adoption/Rejection (IMDAR) Processes For New, Superior Medical Technologies," *International Journal of Healthcare Technology and Management*, 1 (1/2), 200-08.

Woodside and David M. Samuel (1981), "Observations of Corporate Procurement," *Industrial Marketing Management*, 10, 191-205.

Workman, John P. Jr. (1993), "Marketing's Limited Role in New Product Development in One Computer Systems Firm," *Journal of Marketing Research*, 30 (November), 405-21.

Xtreme Programming Organisation (2002), "Xtreme Programming Website," Vol. 2002.

Yoffie, David B., Mary Kwak (1998), “The Browser Wars--1994-98”, Harvard Business School, Boston, 9-798-094

Young, Rob (2001), Personal Interview (Telephone) with Hugh Pattinson, Sydney/Durham NC. (21 December)

Zaplet (2003), "About Zaplet Inc." http://www.zaplet.com/c/@Q3IQgcjFI7Ln-/articles/Z3Launch.htm, Viewed on 9 October.