

Agile adoption and improvement model

Asif Qumer, Brian Henderson-Sellers , Tom McBride
University of Technology

Abstract

Agile software development methods focus on the rapid and iterative delivery of a software product in small increments. Over the last decade, the software industry has shown a substantial interest in agile practices but there is no standard guiding vision model or framework to adopt and then use to assess or improve the agile method in a software development organization; indeed, the absence of a guiding vision model could result in the failure of the agile implementation. The purpose of this paper is to present an Agile Adoption and Improvement Model (AAIM) for the adoption, assessment and improvement of an agile software development process. We have analysed the results of several agile software process assessments, industrial case studies on the adoption of an agile approach and feedback from both researchers and the software industry for the construction of the AAIM. The AAIM can be used as a gradual road map for the adoption of an agile approach so that the required agile level can be achieved and improved over a period of time. The AAIM has been organized in three agile blocks, six agile stages (AS) and an embedded agility measurement model (to quantitatively measure the degree of agility). In AAIM, each stage specifies goals that must be achieved to attain a particular business value through the use of an agile software development approach.

Keywords: *Agile Methods/Methodology, Agile Measurement Model*

INTRODUCTION

In the last few years, several agile methods have been proposed and adopted by the software industry. However, benefits from agile software processes and the applicability of agile methods in large and complex software development organizations is still of significant concern to practitioners. Two fundamental problems have been identified: firstly, the inability of the organizations to construct, execute and manage agile software development processes and, secondly, the absence of a model to guide agile adoption and improvement. Researchers and practitioners have developed a number of assessment tools and frameworks (Kitchenham and Jones 1997, Williams et al. 2004, Tran et al. 2004, Qumer and Henderson-Sellers 2006a) for the assessment of agile software development methods but there is no single framework that can be used as a roadmap or guiding vision model to construct, execute or manage agile software development methods. Therefore, a standard framework or a model is required for the implementation and improvement of an agile approach (how, and how well an organization is practising and applying the agile practices for a software development) in a software development organization.

Previously, a 4-Dimensional Analytical Tool (4-DAT) (Qumer and Henderson-Sellers 2006a,b) for the assessment of agile methods has been developed but it did not discuss agile adoption and improvement in any detail. The AAIM has been more recently developed as a model that could be used together with 4-DAT for the adoption and

improvement of an agile method rationally in a currently non-agile software development organization. The AAIM helps to assess the current state, set goals and define a course for the adoption and continuous improvement of an agile approach. The key to agile success is to build on incremental success, project by project. This paper is organised as follows: Section 2 describes the research methodology and the AAIM construction process with a brief summary of our previous work in this context. Section 3 describes the AAIM. Section 4 explains the enactment and the key features of the AAIM followed by a conclusion in Section 5.

Research Methodology and Model Construction Process

The AAIM has been constructed by the application of an iterative, inductive and interactional mechanism of data collection, instantaneous analysis and emergent interpretation by using a Grounded Theory (Glaser and Strauss 1967; Strauss and Corbin 1990, Pandit 1996) research methodology. The extant frameworks, industrial reports and case studies that have relevance to the emerging model, data and conceptual categories of AAIM have been systematically selected, reviewed and analysed. This paper presents version 1.0 of the AAIM which has been developed in three iterations. In each iteration, the model has been reviewed and updated after the feedback and analysis of the data. Here is a brief summary of the AAIM construction process (see Section 2.1).

The AAIM builds on previous work in agile software process assessment (Qumer and Henderson-Sellers, 2006a, b), the concepts of agility and agile software development methods (Henderson-Sellers and Serour 2005, Qumer and Henderson-Sellers 2006c,); the concepts of an agile management approach (Anderson 2004), the concepts of an agile software development organization (Chau and Maurer 2004), the concepts of an agile approach in a large organization (Lindvall et al. 2004), the concepts of people-orientation (Cockburn et al. 2001), the concepts of an agile way of documenting software (Dickerson 2004), the six agile values and the concepts of agile rationalization (Agile Manifesto 2001, Qumer and Henderson-Sellers 2006d).

The AAIM Construction Process

The following are the three main iterative steps or phases in constructing the Agile Adoption and Improvement Model (AAIM): (1) data collection and coding, (2) theoretical sampling and (3) model development (theory development). Figure 1 shows this process.

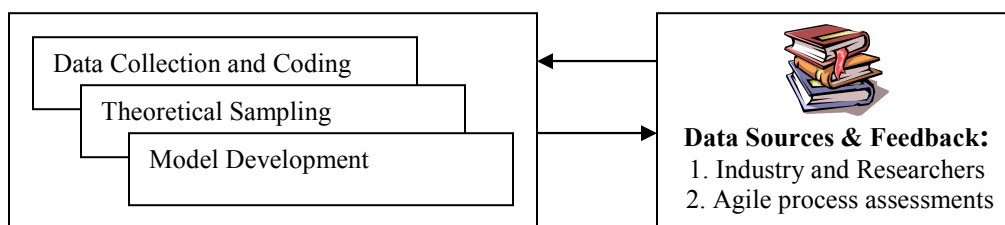


Figure 1: AAIM Construction Process

Data Collection and Coding

Data have been collected from different sources such as the existing agile frameworks (e.g. Beck 2000, Baskerville and Pries-Heje 2001, Schwaber and Beedle 2002, Auer and Miller 2002, DSDM 2003a,b, Aydin 2004 et al. 2004, Koch 2005), industrial agile adoption case studies (e.g. Boehm and Turner 2003, Leffingwell and Muirhead 2004, Nielsen and McMunn 2005, Leffingwell and Smits 2005, McMunn and Nielson 2005, Smits 2006, Elssamadisy 2006, Pettit 2006a,b, Lawrence and Yslas 2006, Sliger 2006, Gat and Martens 2006, Ambler 2006, Barnett 2006, Meadows and Hanly 2006, Qumer and Henderson-Sellers 2007), and agile process assessment (Qumer and Henderson-Sellers 2006a,b).

The “Open Coding” and “Theoretical Coding” (Glaser 1978) techniques have been applied iteratively to identify the different categories and their properties; and then to establish the relationship among them (identified categories). The main identified categories are: flexibility, speed, responsiveness, communication-oriented, people-oriented, executable-artifact, learning and lean.

The agile process assessment results and the interactions among the identified categories/ concepts have been used to define the agile blocks and AAIM levels (AAIML) (see in Section 3).

Theoretical Sampling

The “Theoretical Sampling” (Glaser and Strauss 1967) technique has been used to further develop the properties of the categories in each agile block and agile adoption and improvement model levels (Figure 2).

Model Development

Finally, the relationships and interactions among the different categories of grounded data are used to develop the AAIM, in an iterative manner, which is then tested using the collected data and feedback (arrows in Figure 1).

Agile Adoption and Improvement Model (version 1.0)

The AAIM (Figure 2) has been organized and ordered in three agile blocks, from basic to advanced: an agile-prompt, an agile-crux and an agile-apex. At each block, the degree of agility of an agile process is measured quantitatively by using the agility measurement modelling approach (Qumer and Henderson-Sellers 2006a,b). Furthermore, the AAIM is structured in six agile stages (from stage 1 to stage 6). These stages are embedded in the three agile blocks. Each block and stage have a name and specify the agile practices to follow in order to achieve the particular AAIM level (AAIML). The agile-prompt has the AAIML 1: agile infancy. The agile-crux consists of the core of the AAIM levels, which are AAIML 2: agile initial, AAIML 3: agile realization and AAIML 4: agile value. Finally, the agile-apex block presents the AAIML 5: agile smart and the AAIML 6: agile progress. Each level establishes the agile practices in the agile software development process/method, which in turn enable the organization to achieve the desired AAIML over the period of a time.

Agile Block: Agile-Prompt

Agile-prompt is a point for an organization to start with a basic agile process practice for software development. It consists of the AAIML 1 called agile infancy.

AAIML 1: agile infancy

At this level, a software development organization does not apply an agile method off-the-shelf; rather, the focus of this level is only to introduce and establish the basic agile properties (speed, flexibility and responsiveness) in a software development process/method in practice. Speed enables the quick development of a quality useable software product, a situation-specific emergent software process (agile practices from various agile methods can be combined), a situation-specific emergent development team(s) and a situation-specific emergent development environment (tools) and planning (release planning and iteration planning) by using an iterative and an incremental test-driven approach (test first); flexibility encourages the acceptance and accommodation of changes (generated from an internal environment or by a customer) in a software product, process, plan, development team(s) and development environment (new tools and technology, tools configuration changes); and, finally, responsiveness refers to the fact that not only does it become easy to accept the changes but the changes must be reflected and visible. These are the three properties that establish a foundation to achieve the rest of the agile levels as we cannot achieve them in one go.

Agile-block: Agile-Crux

The agile-crux block (core of the AAIM) consists of 3 levels. The focus of this block is on the establishment of the key agile practices and properties in a software process / method, which differentiate an agile process from a traditional software development approach. The AAIM levels (in this block) are presented in the following sub-sections.

AAIML 2: Agile Initial

At this level of the AAIM, the focus is to enable the communication and collaboration (communication-oriented) among the people by establishing good communication and cooperation protocols within the organization (communication among/within the development teams) and outside the organization (communication with customers and with relevant organizations/stakeholders). It has been noticed that communication and cooperation is very important for working with co-workers and establishing accurate requirements and feedback from customers.

AAIML 3: Agile Realization

This level emphasizes the production of the executable artifacts with a minimal and reduced documentation. The software documentation (non-executable) is used for communication purposes and can be reduced by using other means of communications (verbal or face-to-face communication) and tools. It has been observed that the documentation (non-executable artifacts) can be reduced if there is a well-established communication-oriented culture in the organization (as suggested by the AAIML 2). The AAIML 1, 2 and 3 establish a platform to achieve AAIML 4.

AAIML 4: Agile Value

At this level of AAIM, the practices are established and focused to value the people (people-oriented) both within the organization (developers) and outside the organization (customers) without ignoring the importance of the software development tools and processes. We have observed and noticed that in an agile team, highly skilled people should be indulged (as the agile developers are not only the developers but also the decision makers and they are allowed to do whatever they want to do to achieve a desired business value).

Agile Block: Agile-Apex

The agile-apex block consists of the AAIML 5 and 6. The focus is on the establishment of a learning and quality production environment while consuming minimal possible resources with overall continuous progress in the establishment of an agile environment. The following are the details of these two levels. The consideration of the quality factor does not mean that the rest of the levels do not care about the quality but, here, the stress is to further reduce the production cost while maintaining or improving the production quality (quality should not be compromised while reducing the production cost).

AAIML 5: Agile Smart

At this level, the focus is on the establishment of a learning environment. The learning of the people (involve in a software development), software process (before, after and during the execution of a software process), product (before, during and after the production) and tools (the new tools and a technology) lead toward overall organization learning and improvement.

AAIML 6: Agile Progress

At this level, the practices are focused on the establishment of a lean production environment (the quality production with minimal resources and within a minimum timeframe) and to keep the process agile.

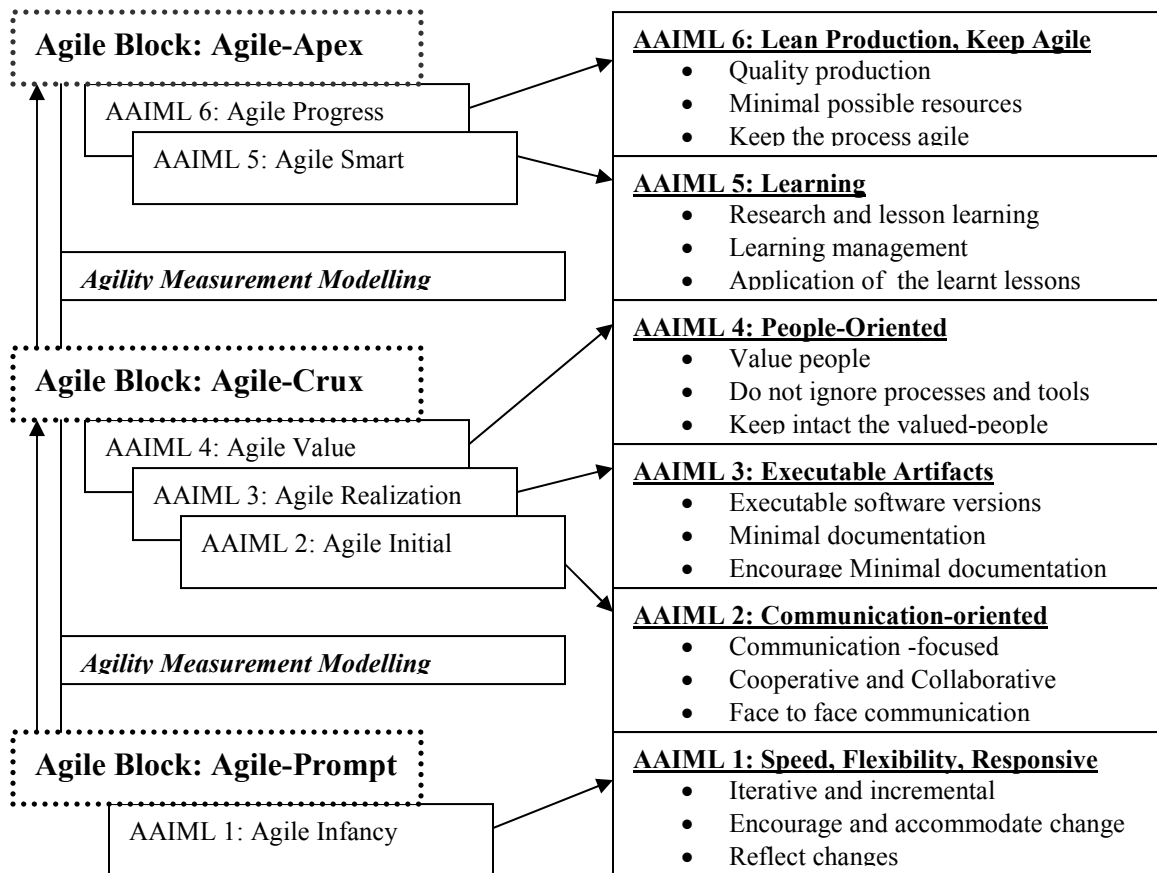


Figure 2. Agile Adoption and Improvement Model

The AAIM Enactment and Key Features

The adoption and improvement of agile practices is a continuous and evolutionary process and takes time, depending upon various factors. The AAIM is a method-independent model that can guide a software development organization to adopt and improve agile practices for a specific situation or project. We had applied AAIM on one of our pilot projects (Qumer and Henderson-Sellers 2007) in the industry (a large software development organization) and found that the success of the agile transition substantially depends on the leading role of the CIO and executive management using AAIM. Such a person should champion the adoption of agile methods and take the responsibility for eliminating any impediments to effective development and delivery of business value through agility. Agile practices could be adopted in different ways by different organizations. The AAIM lays the groundwork for the implementation of agility and the software development organization may tailor or customize AAIM according to their local organizational structure, culture, size and development environment. The transition to an agile software development practice is challenging and, therefore, it is a

good idea to gradually introduce an agile approach in a traditional software development environment/ organization. The results of the various agile adoption case studies have been analysed and it has been noticed that an agile approach requires a different mindset, process, people, environment and tools (Qumer and Henderson-Sellers 2006d) for the successful implementation of an agile approach. The AAIM will help to establish such an environment to successfully follow an agile approach. The AAIM can be used for the assessment of a particular level of agility adoption and to advance to the next level. If the assessment is positive, an organization may proceed with the next level. Otherwise, it should stay at the same level for a specified period of a time as suggested by an assessor. An IT consultant, agile coach, software engineering director, process manager or process quality manager may use this model to introduce and assess an agile process/method for a particular situation. The following are the key features of this model.

- The AAIM helps to assess how and how well an agile process/ method is (in practice) is being followed within a software development organization.
- The AAIM helps to assess the current agile level of an organization.
- The AAIM helps to measure and assess (quantitatively) the degree of agility of a software development process.
- The AAIM provides a roadmap for the establishment of a systematic agile software development environment and the systematic use of agile practices within that environment.
- The AAIM combines the concepts from both theory and practice (data and feedback from both researchers and software industry).

Conclusion

This paper presents an overview of the Agile Adoption and Improvement Model version 1.0. The proposed AAIM has been developed to aid the introduction, assessment and improvement of the agile software development approach (processes or methods) in a software development organization. We have tested this model on one of our pilot projects in industry. We applied this model to transform a large non-agile software development organization to adopt an agile approach. In this pilot project (case study), first the current state of the case study software company was assessed and then an agile product-enhancement process (APEP: hybrid agile practices) was engineered rather than a whole methodology. Secondly, the engineered process (APEP) was adopted by using AAIM. Currently, the case study organization is operating at the AAIML 1 and is successfully practising agile practices for the desired business value. In future, the company is passionate to establish a communication-corporation and a less document-oriented environment at a large scale, which will enable them to achieve AAIM level 2 and 3. The results of the case study (agile transition) highlighting two things: firstly, a step-by-step approach may be considered reasonable for a gradual, successful agile transition or adoption, rather than all at once, which may pose several risks and problems; and, secondly, the appropriateness of agile practices for large and complex projects. We intend to improve the model as we further proceed in our research and get feedback from the software community.

Acknowledgments

We wish to thank the Australian Research Council for financial support under the Linkage Grants Scheme. This is a contribution number 07/07 of the Centre for Object Technology Applications and Research (COTAR). We are also thankful to the people from both the research community and the software industry who helped us with their valuable feedback and experience.

References

- AgileManifesto. 2001. Manifesto for Agile Software Development.
- Ambler, S.W. 2006. 'Scaling Agile Development via Architecture'. Agile Journal
- Anderson, D. J. 2004. Agile Management for Software Engineering. Pearson Education Inc.
- Auer, K. and Miller, R. 2002. Extreme Programming Applied: Playing to Win. Addison Wesley, Pearson Education, Indianapolis.
- Aydin, M. N., Harmsen, F., Van Slooten, K. and Stegwee, R. A. 2004. 'An Agile Information Systems Development Method in Use'. In Turkish Journal of Electrical Engineering & Computer Sciences, Vol. 12 Scientific and Technical Research Council of Turkey, pp. 127-138.
- Baskerville, R. and Pries-Heje, J. 2001. 'How Internet is redefining information systems development methodology'? Realigning research and practices in IS development, 49-68.
- Barnett, L., Learning From Others. 2006. 'Best Practices for Large-Scale Agile Development'. Agile Journal.
- Beck K. 2000. Extreme Programming Explained, Addison-Wesley Pearson Education, Boston.
- Boehm, B. and Turner, R. 2003. 'Observations on Balancing Discipline and Agility'. Proceedings of the Agile Development Conference, IEEE Computer Society.
- Chau, T. and Maurer, F. 2004. 'Tool Support for Inter-Team Learning in Agile Software Organizations'. Proceedings of the Workshop on Learning Software Organizations, Banff, Springer.
- Cockburn, A., Highsmith, J. and Boehm, B. 2001. 'Agile Software Development: The People Factor'. In Computer, Vol. 34, pp. 131.
- Dickerson, C. 2004. 'Documentation Dilemma'. In InfoWorld, Vol. 26, InfoWorld Publishing Group.
- DSDM. 2003a. DSDM Consortium, Dynamic Systems Development Method Ltd.
- DSDM. 2003b. Guidelines For Introducing DSDM Into An Organisation Evolving to a DSDM Culture, DSDM Consortium.
- Elssamadisy, A. 2006. 'Getting Beyond "It Depends!" Being Specific but Not Prescriptive About Agile Practice Adoption'. Agile Journal, www.agilejournal.com.
- Gat, I. and Martens, R. 2006. 'CASE STUDY: How BMC Is Scaling Agile Development'. Agile Journal, www.agilejournal.com
- Glaser, B.G., 1978. Theoretical sensitivity: Advances in the methodology of grounded theory. Sociology Press, Mill Valley, CA.
- Glaser, B. G. and Strauss, A. L. 1967. The Discovery of Grounded Theory: Strategies for Qualitative Research. Aldine Publishing Company, New York.
- Henderson-Sellers, B. and Serour, M.K. 2005. 'Creating a dual agility method - the value of method engineering'. J. Database Management, Vol. 16, No. 4, pp.1-24.
- Kitchenham, B. A. and Jones, L. 1997. 'Evaluating Software Engineering Methods and Tool part 5: the Influence of Human Factors'. ACM SIGSOFT Software Engineering Notes, Vol. 22, No. 1, pp.13-15.
- Koch, A. S. 2005. Agile Software Development: Evaluating the Methods for Your Organization. London, Artech House, Inc.
- Lawrence, R. and Yslas, B. 2006. 'Three-way Cultural Change: Introducing Agile within Two Non-Agile Companies and a Non-Agile Methodology'. Proceedings of AGILE 2006 Conference, Minneapolis, USA: IEEE Computer Society.
- Leffingwell, D. and Smits, H. 2006. A CIO's Playbook for Adopting the Scrum Method of Achieving Software Agility. Rally Software Development Corporation and Ken Schwaber-Scrum Alliance.
- Leffingwell, D. and Muirhead, D. 2004. Tactical Management of Agile Development: Achieving Competitive Advantage. Rally Software Development Corporation.

- Lindvall, M., Muthig, D., Dagnino, A., Wallin, C., Stupperich, M., Kiefer, D., May, J. and Kahkonen, T. 2004. 'Agile Software Development in Large Organizations'. *Computer*. 37: 26-34.
- Meadows, L. and Hanly, S. 2006. 'Agile Coaching in British Telecom'. *Agile Journal*, www.agilejournal.com.
- McMunn, D. and Nielsen, J. 2005. 'Showcasing Agile Development: A Powerful Tool for "Crossing the Chasm"'. *Digital Focus*.
- Nielsen, J. and McMunn, D. 2005. 'The Agile Journey: Adopting XP in a Large Financial Services Organization. Proceedings of the XP2005, Springer-Verlag Berlin Heidelberg.
- Pandit, N.R. 1996. The Creation of Theory: 'A Recent Application of the Grounded Theory Method'. <http://www.nova.edu/ssss/QR/QR2-4/pandit.html>.
- Pettit, R. 2006a. 'An "Agile Maturity Model?"'. *Agile Journal*, www.agilejournal.com.
- Pettit, R. 2006b. 'Scaling Up: An Approach to Organizational Agile Adoption'. *Agile Journal*, www.agilejournal.com.
- Qumer, A. and Henderson-Sellers, B. 2006a. 'Measuring agility and adoptability of agile methods: A 4-Dimensional Analytical Tool'. *Procs. IADIS International Conference Applied Computing* (eds. N. Guimarães, P. Isaias and A. Goikoetxea), IADIS Press, 503-507
- Qumer, A. & Henderson-Sellers, B. 2006b. 'Comparative evaluation of XP and Scrum using the 4D Analytical Tool (4-DAT)'. *Proceedings of the European and Mediterranean Conference on Information Systems 2006 (EMCIS2006)* (eds. Z. Irani, O.D. Sarikas, J. Llopis, R. Gonzalez and J. Gasco), CD, Brunel University, West London
- Qumer, A. & Henderson-Sellers, B. 2006c. 'Crystallization of agility: back to basics'. *ICSOFT 2006 Proceedings of the First International Conference on Software and Data Technologies, INSTICC Press, Volume 2*, 121-126
- Qumer, A. & Henderson-Sellers, B. 2006d. 'A Framework to Support Non-fragile Agile Agent-Oriented Software Development'. *Frontiers in Artificial Intelligence and Applications. New Trends in Software Methodologies, Tools and Techniques. Procs. Fifth SoMeT_06* (eds. H. Fujita and M. Mejri), IOS Press, 84-100
- Qumer, A. and Henderson-Sellers, B., 2007. 'Construction of an Agile Software Product-Enhancement Process by Using an Agile Software Solution Framework (ASSF) and Situational Method Engineering'. *COMPSAC, IEEE*. (Submitted)
- Schwaber K and Beedle M. 2002. *Agile Software Development with SCRUM*. Prentice Hall.
- Sliger, M. 2006. *A Project Manager's Survival Guide to Going Agile*. Rally Software Development Corporation.
- Smits, H. 2006. *5 Levels of Agile Planning: From Enterprise Product Vision to Team Stand-up*. Rally Software Development Corporation.
- Strauss, A. and Corbin, J. 1990. *Basics of Qualitative Research: Grounded Theory, Procedures and Techniques*. Newbury Park, CA, Sage Publications. Engineering. Edinburgh, pp.11-20.
- Tran, Q.-N. N., Low, G. and Williams M-A. 2004. 'A Preliminary Comparative Feature Analysis of Multi-agent Systems Development Methodologies'. *6th International Bi-Conference Workshop on Agent-Oriented Information Systems*, New York, USA, pp.386-398.
- Williams, L., Krebs, W. Layman, L., Anton, A.I. and Abrahamsson, P. 2004. 'Toward a Framework for Evaluating Extreme Programming'. *Empirical Assessment in Software*