Measurements used in Software Quality Evaluation

Bernard Wong
bernard@it.uts.edu.au
University of Technology, Sydney
Faculty of Information Technology
PO Box 123, Broadway,
NSW 2007, AUSTRALIA.

Abstract

The paper presents the results of two quantitative studies, which look at the measurements used in software evaluation. These measurements are extremely important, as they are the means for conducting the evaluation of the software. The first study collected a list of measurements, which were then tested by the second study. The results showed that in software evaluation, stakeholders do not differ greatly in the measurements used; though in previous studies it was clearly shown that stakeholders differ in the level of importance placed on the software characteristics. This means that the same measurements can be used for the evaluation regardless of the stakeholders though the algorithms used to calculate the quality and the predicted effects on desired consequences and values may differ for each stakeholder. The results of this study are important as it identifies the metrics, perceived by stakeholders as essential for applying the Software Evaluation Framework to software evaluation.

Keywords: Software evaluation, Software Quality, Metrics, Measurements, Empirical Software Engineering, Human Factors

1. Introduction

It has been said that there are as many definitions of quality as writers on the subject. Perhaps, fortunately, the latter have been remarkably few in number considering the obvious importance of the concept and the frequent appearance of the term quality in everyday language. Though the topic of software quality has been around for decades, software product quality research is still relatively immature, and today it is still difficult for a user to compare software quality across products. Researchers are still not clear as to what is a good measure of software quality because of the variety of interpretations of the meaning of quality, of the meanings of terms to describe its aspects, of criteria for including or excluding aspects in a model of software, and of the degree to which software development procedures should be included in the definition. In a recent paper by Wong & Jeffery [1], a framework for Software Quality evaluation is introduced.

This framework for software evaluation, gives the rationale for the choice of characteristics used in the evaluation, whilst also supplying the underpinning explanation for the multiple views of quality. The framework has its theoretical foundations on value-chain models, found in the disciplines of cognitive psychology and consumer research, and introduces the use of cognitive structures as a means of describing the many definitions of quality, whilst also showing the rationale behind these differences.

A number of papers over the past few years have covered different aspects of this framework ([1], [2], [3]). In 2001, the first paper was published [2]. The paper's focus was on exploring, through a qualitative study, the use of cognitive structures as a means of describing Gutman's means-end chain relationship [2]. This means-end chain relationship is the underpinning theory on which the software evaluation framework is based. In the second paper, the results of a larger quantitative study, on the appropriateness of Gutman's model in software quality evaluation was reported [3]. And during the third paper, the software evaluation framework SEF is introduced. The paper validates the cognitive structures introduced in the earlier qualitative study [2], whilst also discussing potential significance for this framework.

This paper investigates the measurements used in software quality evaluation. Whilst the earlier papers have described the characteristics relevant to software evaluation, they fail to address the measurements used in the evaluation process and how they appraise these characteristics.

The paper reports on two quantitative studies of stakeholders' definition and understanding of software quality. The first study involved 118 subjects whilst the

second study involved 403 subjects. Both studies involved different stakeholders, programmers, analyst programmers, systems analysts, managers, technical engineers and software users.

2. Software Quality Evaluation Framework

During the past thirty years there have been many studies on the topic of software quality, however there have been none on a framework for software quality, which considers the motivation behind the evaluation process, other than the earlier version of this framework introduced by Wong & Jeffery [1]. The recent studies of Wong & Jeffery ([1], [2], [3]) provide the premise to the framework introduced here in this paper. As described in an earlier paper [3], this framework is based on the notion that software evaluators are influenced by their job roles. This is supported by earlier studies ([4], [5]) where stakeholders with different job roles were found to focus on different sets of software characteristics when evaluating software quality. What motivates these differences is found within the broader context of value, where studies have shown that values are a powerful force in influencing the behavior of individuals ([6], [7]).

The theoretical basis for developing such a framework was based on the theory found in cognitive psychology, and adopted Gutman's Means-End Chain Model ([8], [9], [10], [11], [12]), which posits that linkages between product characteristics, consequences produced through usage, and personal values of users underlie the decision-making process, or in this case, the software quality evaluation process. It is proposed in this research, that a framework be introduced for software quality evaluation, which focuses on the relationships between the characteristics, the consequences and the values, as introduced by Gutman's Model. It is the aim of the framework to not only show the relationships between the characteristics and software quality, but also show that there are relationships between the characteristics and the desired consequences, and between the characteristics and the sought after values.

As highlighted in the literature, the benefit of utilizing Gutman's model in the framework is that it shows how the desired values influence the behaviors of individuals in all aspects of their lives ([6], [7], [8]). Gutman's Model suggests that the desired consequences and the values sought are the motivators behind the choice of characteristic for software evaluation. In addition to this, the framework also highlights the significants of this relationship through the relationships between characteristics and consequences and also between the characteristics and value. It is through these relationships that allow the possibility of using the characteristics to evaluate each consequence and value.

The framework distinguished individual responses in terms of three broad classes of elements: characteristics, consequences and value. This framework provided a good foundation for developing relevant hypothesis for this study. The results elicited the various combinations of characteristics, consequences and values for each person.

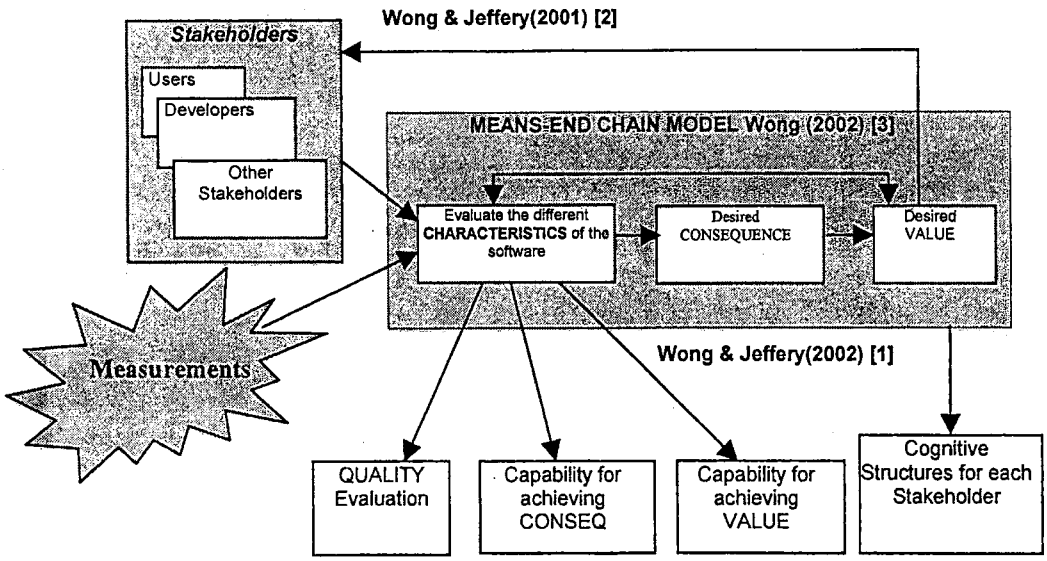

Figure 1 SEF: Software Evaluation Framework

The framework shown in figure 1 is based on Gutman's Means-End Chain Model. As can be seen in this diagram, the framework consists of a number of boxes describing the three elements of Gutman's Model, the stakeholders who evaluate the software quality, the outcome for the Quality evaluation, and the arrows linking these elements, whilst also describing the direction of the influence. The Means-End Chain Model has been placed in the main box, as it is proposed, in this framework, to be the central influence for the choice of characteristics used in software evaluation, and the influence for the differences found between stakeholders. Parts of this framework have already been investigated and have been reported in recent papers ([1], [2], [3]). An exploratory study by Wong & Jeffery [2], utilized a qualitative approach to explore the influence of value on the choice of characteristics, and to determine whether cognitive structures could be used as a tool to represent the links between the characteristics, consequences and values. This study also supported earlier pilot studies on stakeholder differences ([4], [5], [6]), identifying different cognitive structures for users and developers. A more recent paper by Wong [3] reported on a large quantitative study, which tested the appropriateness of utilizing Gutman's Model in software evaluation.

What this paper will focus on will be the measurements used by the different stakeholders. The previous studies have shown that different stakeholders focus on different characteristics in their evaluation of software. The question to be asked in this study is whether the measurements for the characteristics also differ, and whether there are multiple measurements used for each characteristic. As such, figure 1 has been modified to include measurements as an input to characteristics.

3. Data Collection and Analysis

In order to conduct this investigation, two quantitative studies were conducted. In the first study, 118 subjects from different industry areas completed the survey. The study sought definitions for the meaning of "software quality" from a range of stakeholders, requesting the subjects to list the measurements they would use to evaluate the software quality.

The industries from which the respondents were drawn included betting and gaming, finance, meat and livestock, telecommunications, pharmaceutical & medical suppliers, hospitals, food manufacturing, engineering, airline, advertising, oil marketing, transport, government services, and computing companies. The organizations ranged in size from about 25 employees, to organizations having several thousand employees. The subjects came from different jobs and backgrounds, programmers, analyst programmers, systems analysts, managers, technical engineers and software users. A spread of people with different backgrounds was chosen so that the survey could cover a wider range of stakeholders. Of those surveyed, there were 42 developers, 24 end users, 13 managers and 39 who did not match the above areas.

The analysis began with content analysis of the different words and phrases used in the lists of measurements and characteristics given by those surveyed. They were grouped, based on their meanings. The data was then analyzed statistically to ascertain the level of popularity for each characteristic. This was achieved by counting the occurrence of each measurement. The data was then reorganized into four groups according to the backgrounds of the subjects. These four groups were programmers, managers, users (including students) and others (who were mainly technical people). The result of this first study gave a list of measurements.

In the second quantitative study, 403 subjects were surveyed. The organizations involved in this study were from the telecommunications industry banking industry, insurance industry, airline industry and the dot.com industry. All organizations supplied users and developers for the survey. Subjects were asked to evaluate their in-house developed software applications.

The questionnaire was distributed to over 600 users and developers who had a close relationship with their in-house software applications. With the aid of organizations from different industries, contacts were made with appropriate managers, to seek assistance from their staff. Respondents were sought from those who develop or maintain in-house applications in the participating firms or those who use in-house developed applications in their daily job.

The end result of this study was most pleasing, with 210 users and 193 developers responding. Though there were 530 attempts at the survey, 127 were not usable, either because no attempt was made at any of the questions, or only a small percentage of questions were attempted. In total, 403 usable results were collected from 22 organizations.

The questionnaire used the results from the first study. Questions of similar design were put together, and distinctive typefaces were used for questions, answers and directions [14]. The questions were highly structured in order to make completion and comparative analysis easier. Because open questions are more likely to cause problems of categorization and so lead to false conclusions, or to over-represent the more convinced and more articulate [15], the great majority of responses involved selecting an appropriate box.

Unfortunately, the limitation on the length of this paper prohibits further details of the research instrument and method.

4. Results

As described at the beginning of this paper, the results of the first study form the basis to the survey used for the second study. A list of measures perceived as important to the stakeholders for evaluating software were collected from this study, and are listed in table 1. The table lists not only the measurements, but also the characteristic type it refers to and the percentage of subjects who listed the measurement. As earlier studies have identified seven characteristics, Usability, Functionality, Operational, Technical, Institutional, Service and Economic, this study has begun with the assumption that each metric is associated with one of these characteristics, and that a number of metrics can be associated with the same characteristic. It should be pointed out at this time that the characteristic technical refers to the ISO9126 characteristics portability and maintainability and the characteristic operational refer to the ISO9126 characteristics reliability and efficiency.

Most of the participants of the survey who are in the "others" category, work in the technical/engineering areas. An assumption can easily be made that the technical/engineering areas are similar to the developers.

Characteristic Type	Measurement	Developers(%)	End Users(%)	Managers(%)	Others (%)	Total(%)
Economic	Cost	36	25	57	34	36
Usability	Docs	31	25	43	34	32
Support	Support	21	13	29	24	21
Usability	Online Help	2	25	7	13	11
Operational	Hardware Req	12	8	7	11	10
Usability	Flexibility	10	13	14	5	9
Institutional	Popularity	5	4	14	5	6
Institutional	Brand Name	7	4	14	3	6
Usability	Easy to Install	0	0	0	18	6
Usability	Looks Good	0	13	0	8	5
Technical	Size	2	4	0	8	4
Operational	Effectiveness	2	4	0	5	3
Usability	Presentation	7	0	0	3	3
Institutional	Reputation	2	8	0	3	3
Operational	Modern Tech	5	4	0	0	3
Technical	Potential for Growth	2	4	0	3	3
Functional	Additional Functions	5	0	0	0	3
Operational	Availability	0	4	0	3	2
Functional	Features	2	0	0	3	2
Technical	Future Reuse	5	0	0	0	2
Technical	Good Memory	2	0	0	5	2
Usability	Good Screen Design	0	8	0	0	2
Usability	Intuitive	0	0	0	5	2
Economic	Value for Money	2	4	0	0	2
Operation/Technical	Well Designed	5	0	0	0	2
Usability	Clarity	2	0	0	0	1
Usability	Consistency	0	0	0	3	1
Technical	Development	0	0	7	0	1
Operational	Good Backups	2	0	0	0	1
Institutional	Good Reviews	0	4	0	0	1
Technical	Modular Design	2	0	0	0	1
Operational	Powerful	0	4	0	0	1
Functional	Productive	2	0	0	0	1

Technical	Referential Integrity	2	0	0	0	1
Operational	Storage	0	4	0	0	1
Operational	Tested Effectively	2	0	0	0	1
Functional	Utilities	0	4	0	0	1
Institutional	Vendor Stability	2	0	0	0	1
Functional	Version	0	4	0	0	1

Table 1: Percentage of subjects selecting the above measurements.

The results from the first study showed that the stakeholders do not vary in the measurement used for the evaluation. The results do not show any possible patterns for the stakeholders, but shows that all measurements, regardless of stakeholders are appropriate for software evaluation. This means that the same measurements can be used for the evaluation regardless of the stakeholders though the algorithms used to calculate the quality and the predicted effects on desired consequences and values may differ for each stakeholder. However the results show that some of the measurements are more popular than others, however it is difficult to determine whether a measurement is more appropriate for one stakeholder or less appropriate for another stakeholder. The results also show that managers do not use measurements in the technical or operational characteristics, and that the metrics used for evaluation lends it to belong to the economic or institutional characteristics. This is supported by the previous study of Wong & Jeffery ([4], [5]).

The results of the second study were first analyzed for reliability. Cronbach's Alpha was used to test that the items used to measure each variable was reliable. For each scale, inter-item correlation coefficients were examined, and Cronbach's alpha coefficients were calculated. This was conducted on each of the seven characteristic groups.

Scale	Cronbach's Alpha
Support	alpha=0.9352
Economic	alpha=0.8563
Institutional	alpha=0.7933
Technical	alpha=0.9666
Functional	alpha=0.8457
Usability	alpha=0.9537
Operational	alpha=0.8821

Table 2 Reliability Coefficients

Cronbach's alpha measures the reliability or internal consistency of a scale. Nunnally [16], suggested reliability in the 0.5 to 0.6 range in the early stages of research to 0.95 as being the desirable standard for research in applied educational settings. In business settings, there are no generally accepted guidelines [17], although Van de Ven and Ferry [18] had suggested a range of 0.55 to 0.90 for constructs with narrow to moderately broad conceptual scope. This 0.55 to 0.90 range will be used for assessing the reliability of the measures here.

In order to provide some evidence for the discriminant validity among constructs used in the model, a factor analysis was also conducted. Three main steps were used to conduct the factor analysis. First, an assessment of the suitability of the data, second, the factor extraction, and third, factor rotation and interpretation.

There are two main issues to consider in determining whether a particular data set is suitable for factor analysis: sample size and the strength of the relationship among the items. While there is little agreement among authors concerning how large a sample should be, the recommendation generally is: the larger, the better. In small samples the correlation coefficients among the variables are less reliable, tending to vary from sample to sample. Tabachnick and Fidell [19] review this issue and suggest that 'it is comforting to have at least 300 cases for factor analysis'. As the sample size for this study is 403, this criteria is easily satisfied. Though a number of researchers ([20], [21]) suggest that the sample size is important, there have been suggestions that it is not the overall size that is of concern, but rather the ratio of subjects to items. Nunnally [16] recommends a 10 to 1 ratio, that is 10 cases for each item to be factor analyzed. Others suggest that 5 cases for each item is adequate in most cases [19]. The study has 403 cases with 56 items, which gives close to a 7 to 1 ratio.

The other issue to address is whether the data is factorable. To determine whether the data is appropriate for factor analysis, the Kaiser-Meyer-Olkin Measure (KMO) is used. The Kaiser-Meyer-Olkin Measure is a measure of sampling adequacy. Values of 0.60 and above are required for good Factor Analysis (Tabachnick and Fidell[1996]) and the Kaiser-Meyer-Olkin Measure of Sampling Adequacy (KMO) for this data set was 0.948, well above the recommended requirement. Therefore the data is suitable for factor analysis.

Factor extraction was then performed using principal components analysis. According to Pallant [21], this approach is the most commonly used. The Kaiser

criterion was used to determine the number of factors, where only the factors with eigenvalues greater than 1.0 are retained. The factor analysis was conducted for both users and developers. Three analyses were performed, to analyze users and developers together, to analyze only users, and then to analyze only developers. In each, five factors were found to satisfy this condition, which supports the results described in earlier studies ([1], [2], [3]).

And finally the factors are rotated. This does not change the underlying solution, but rather it presents the pattern of loadings in a manner that is easier to interpret.

The result of this paper shows that the characteristics in the framework can be assessed through the use of the metrics identified in this study. Unfortunately, limitation on the length of the paper prohibits the listing of the three factor analysis, however, it should be highlighted that the results support the notion that the metrics, associated with the same characteristics measure the characteristic in similar ways. The cronbach's alpha, listed in table 2, shows that the metrics associated with the characteristic, give reliable measures for the characteristic. The result also shows through the factor analysis that the metrics can be grouped according to the characteristics proposed in the framework. All three matrices support the grouping of the metrics according to the characteristics of the framework. However, it should be noted that the list of metrics is not complete, and that further measures are still to be studied. It should also be highlighted that whilst groups of metrics appear to be able to measure the same characteristic, it is not conclusive as to whether one metric can replace the other in the measure of that characteristic. Further studies are required.

5. Conclusion

The benefit of the framework till now, has only allowed the identification of the links between the characteristics and software quality, and the characteristics, which have significant effect on the individual consequences, and values. With the results of this study, the framework can introduce a set of metrics appropriate for each characteristic. The framework can now be applied to industry as a software evaluation tool. Of course, further work is still required in the study of these metrics, and also in the study of the processes, which work with the framework. This is extremely important in the evaluation of software, as it introduces measurements of the characteristics in order to calculate the software quality and the effects of the software on each desired consequence and value. What is also important is that this framework, along with the metrics addresses software evaluation in terms of the perceptions held by the different stakeholders.

The result of this study also introduces a valuable addition to the Goal Question Metric Method (GQM) ([22]). The most notable feature of GQM, is the central role of a goal, focusing on the fact that there must be a reason for measuring, and that without goals, patterns are unlikely to be visible, since the data collected will be unfocused. Basili rejects the notion of fixed sets of metrics, which many of the earlier models of quality are based on, but instead offers a method to assist tailoring sets of metrics to specific goals. However, it has been generally accepted that the progression from the goals to the questions is the most difficult aspect of GQM. The method provides little guidance, relying instead upon the judgment, experience and insight of those involved with measurement to identify useful questions. There exists a multiplicity of questions that could be asked about virtually any goal. The research described in this paper, solves this problem for software quality evaluation through the use of cognitive structures. Values sought by the stakeholders are appropriate goals, with the cognitive structures supplying related questions, which are associated with these goals. This results in creating the purpose for measurement, setting in place the progression from the goals, to the questions and finally to the metrics, which are associated with the software characteristics. The new addition of "metrics" to the framework completes the mapping to the GQM Method. Not only does it assist in identifying potential Goals and Questions, it now links to the Metrics.

Though further studies of the metrics is still required, the addition of the new metrics section to the framework, results in being able to apply the Software Evaluation Framework in practice.

References

1. Wong, B. & Jeffery, R.: A Framework for Software Quality Evaluation, Proceedings of the Fourth International Conference, PROFES 2002, pp 103-118.
2. Wong, B. & Jeffery, R.: Cognitive Structures of Software Evaluation: A Means-End Chain Analysis of Quality, Proceedings of the Third International Conference, PROFES 2001, 2001, pp 6-26.
3. Wong, B: The Appropriateness of Gutman's Means End Chain Model in Software Evaluation, Proceedings of the 2002 International Symposium on Empirical Software Engineering, ISESE 2002, pp 56-65.
4. Wong, B. & Jeffery, R.: Quality Metrics: ISO9126 and Stakeholder Perceptions, Proceedings of the Second Australian Conference on Software Metrics, 1995, pp 54-65.
5. Wong, B. & Jeffery, R.: A Pilot Study of Stakeholder Perceptions of Quality, Technical Report, CSIRO, 1996.
6. Rokeach: Beliefs, Attitudes and Values, San Francisco: Jossey Bass, 1968.

7. Yankelovich: New Rules, New York: Random House.
8. Gutman,J.: A Means-End Chain Model Based on Consumer Categorization Processes, Journal of Marketing, 46 (Spring): 60-72.
9. Gutman.J. : Means-End Chains as Goal Hierarchies, Psychology & Marketing, 14 (6):1997 545-560.
10. Bagozzi, R.: Goal-directed behaviours in marketing: cognitive and emotional, Psychology & Marketing, 14, Sept 1997, pp 539-543.
11. Valette-Florence, P.: A Causal Analysis of Means-End Hierarchies in a Cross-cultural Context: Metholodogical Refinements, Journal of Business Research, v 42 No 2 June 1998, pp161-166.
12. Bagozzi, R. and Dabholkar, P.: Discursive psychology: an alternative conceptual foundation to means-end chain theory, Psychology & Marketing, 17, July 2000, pp 535-586.
13. Kahle L, Beatty S and Homer P.: Alternative Measurement Approaches to Consumer Values: The List of Values (LOV) and Values and Life Style (VALS), Journal of Consumer Research, 13 December, 1986, 405-409.
14. Oppenheim A: Questionnaire Design, Interviewing and Attitude Measurement, London: Pinter, 1992.
15. Payne S: The Art of Asking Questions, New Jersey: Princeton University, 1951.
16. Nunnally J C: *Psychometric Methods*, 2nd ed McGraw-Hill, New York, 1978.
17. Peter J: Reliability: A Review of Psychometric Basics and Recent Marketing Practices, *Journal of Marketing Research*, **16**, February, 1979, pp. 6-17.
18. Van de Ven A H and Ferry D F: Measuring and Assessing Organisations, New York, NY: John Wiley, 1980.
19. Tabachnick B G and Fidell L S: *Using Multivariate Statistics*, 3rd Edition.. New York: Harper Collins, 1996.
20. Stevens J: *Applied Multivariate Statistics for the Social Sciences*, 3rd Edition. Mahwah, New Jersey: Lawrence Erlbaum, 1996.
21. Pallant J: *SPSS Survival Manual*, Allen & Unwin, 2001.
22. Rombach D and Basili V: Practical Benefits of Goal-Oriented Measurement, Proceedings Annual Workshop of the Centre for Software Reliability: Reliability and Measurement, Garmisch-Partenkirchen, Germany: Elsevier, 1990.