Software Quality Model - Comparative Study

Falak khursheed[1], Mohd Haroon[2], Jameel Ahmad[3]

[2, 3]*Dept. of CSE, Integral University, Lucknow, India*

***Abstract*— Actually, software products are increasing inside a fast way and they are used in nearly all activities of people life. As a consequence measuring and evaluating the standard of a software product is now crucial task for most companies. Several models are anticipated to help different kinds of users having quality issues. This paper analyses our ability to measure software excellent. The analysis is dependent on the representational hypothesis of measurement. We dissect three assumptions on application quality measurement and suggest a higher measurement focus with establishing experimental relational programs (models of what we should measure) in application quality related perform. Actually, software products are increasing in a fast way and are used in almost all activities of human life.**

***Keywords*— Software Quality, Software Quality Model, Quality Factors.**

I. Introduction

Study on software quality will be as old as software production plus the apprehension for high quality products arises with the design of error-free programs and also efficiency when employed. Research to improve the quality of software is generated due to users demand intended for software products along with increasing quality. Really, this is considered an engineering self-discipline. Over the very last decade, the general focus with the software industry features shifted from providing a lot more functionality to improving what continues to be coined as the consumer experience. The user experience identifies characteristics such since ease-of-use, security, balance and reliability. Improvements in such areas lead to a improved quality as perceived from the end users. The goal of this paper is to identify the requirements for just a software quality model being used as a new foundation to Computer software Quality Engineering.

II. Software Quality

What actually constitutes the grade of a product is normally "the degree to which a couple of inherent characteristics satisfies requirements". Software quality is dependent upon a set associated with quality factors. It is dependent upon the user satisfaction and with the errors or unexpected behavior with the software. These definitions are most often based on a similar "intuition" of exactly what Software quality is usually. For example, they often share the view that software quality would be the degree of meeting anyone needs. The difference may possibly lay in whether or not they consider user needs in the form of (1) requirements and implied needs, (2) user satisfaction or (3) the amount of incorrect behavior with the software. From your viewpoint of dimension theory these variances are significant, and result in different empirical relational techniques.

A. Quality As A Set Of Quality Factors

An example of "quality as a couple of quality factors" is the definition in ISO 8402-1986 which in turn defines quality seeing that: The totality connected with features and qualities of a services or products that bear on its ability to meet stated as well as implied needs. Types of such "features and characteristics" (quality factors) are generally efficiency, flexibility, honesty; inter operability, maintainability and portability. The above sort of definition has, a few, the following implications to the empirical relational technique of software excellent:

- The relation "better quality than" ought to be interpreted as an even better ability to fulfill stated and meant needs.
- In order to measure and review software quality we must formulate an empirical connection involving the quality factors and also the software quality themselves; i. e. computer software quality is ultimately measured.

From some sort of measurement theory point of view, this means that this quality factor based definitions will not enable measurement connected with software quality. One might argue that these quality factor classifications suggest a way of measuring of quality factors as a way to predict the computer software quality. In this situation an accurate connection involving the quality factors and also the software quality seriously isn't needed. On one other hand, in order to get a meaningful prediction technique we still will need an empirical relational system including an acknowledged understanding of this relation "higher excellent than", i. at the. An accepted comprehension of "a higher ability to meet user needs". Currently, such an empirical relational system does not exist.

B. Quality as user satisfaction

A common approach, for example frequently applied from the quality framework Total Quality Management (TQM), is to define or understand computer software quality as the level of user satisfaction. This knowledge of software quality features, among others, these consequences for the particular empirical relational technique of software quality:

- There must be a commonly accepted method of identifying the consumer satisfaction.

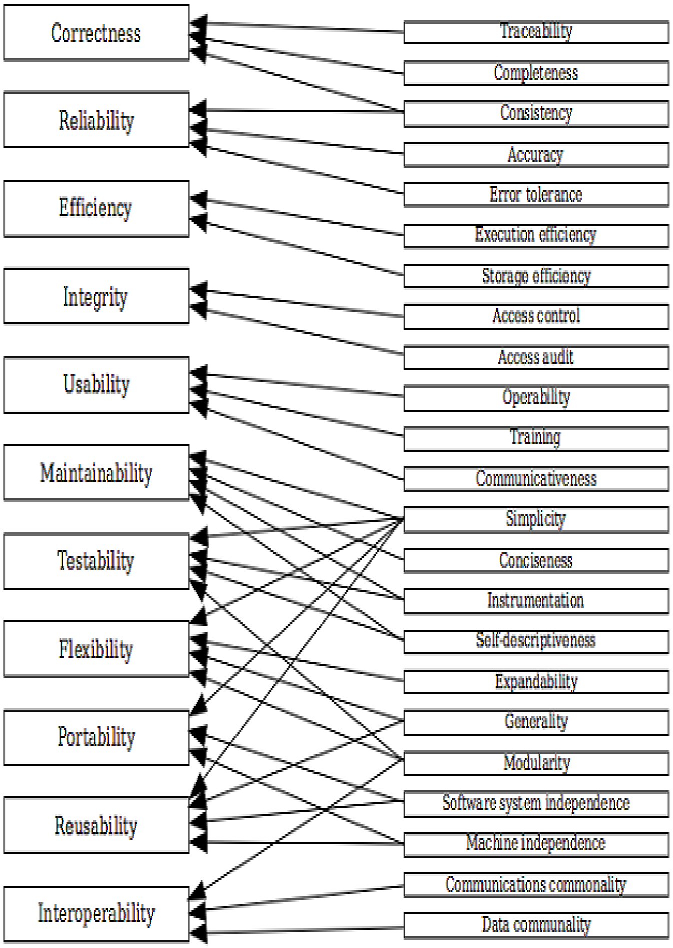
- There must be a commonly accepted meaning on the relations "same quality as" and "better quality than" according to user satisfaction. Natural meats argue that it is possible because:

A of identifying the user satisfaction may be to question them. "same/higher quality than" can be understood as the particular same/higher proportion associated with satisfied users.

III. Quality Models

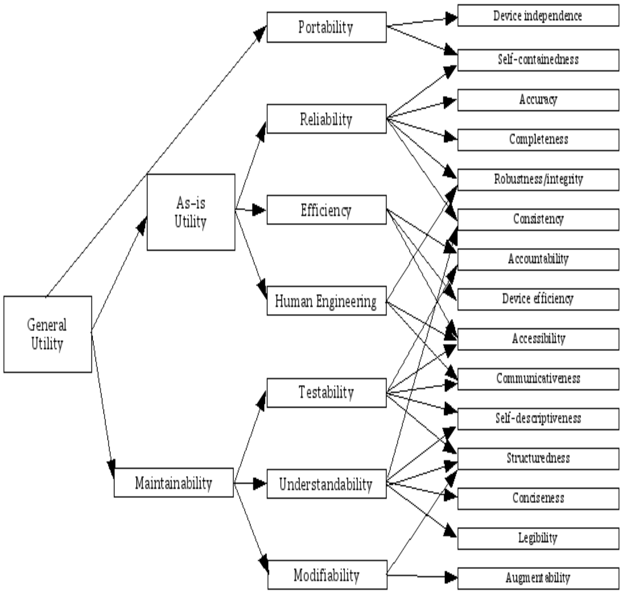
A. Mc Call Model

Your Mc Call design established product good quality through several features. These were arranged into three perspectives: Product Review (maintenance, flexibility, and testing), Product Operation (correct, trusted, efficient, integrity and usability) and Product Transition (portability, re usability and inter operability). The major contribution on the McCall method was to take into account relationships between good quality characteristics and metrics. This model seemed to be used as base for your creation of people quality models. This model seriously isn't applicable depending on criteria outlined inside the IEEE Standard for any Software Quality Metrics Methodology for any top to bottom approach to quality engineering. Furthermore, it emphasizes the merchandise perspective of quality for the detriment of additional perspectives. It is therefore not suited as being a foundation for



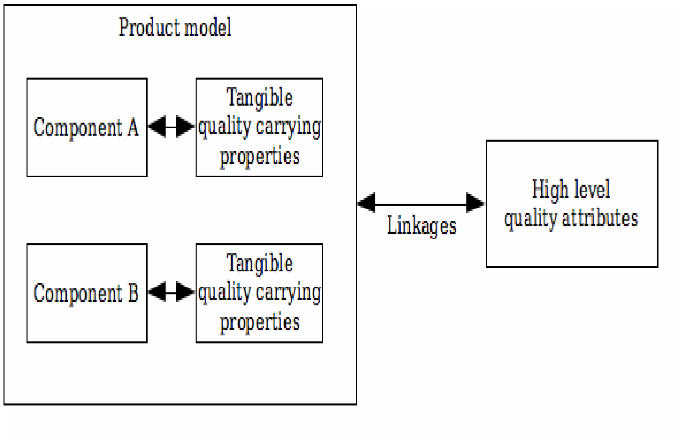
B. Boehm

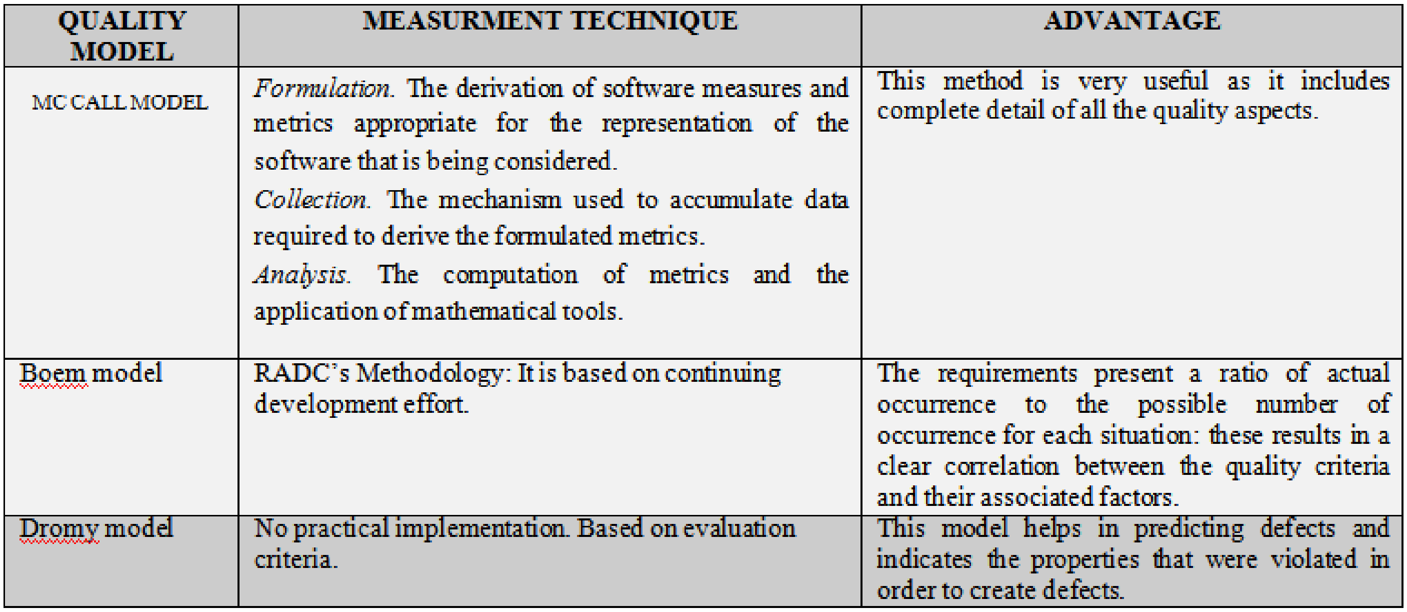
Boehm [8] ensures large-scale characteristics that constitute a marked improvement over the Mc Telephone model because create factors at diverse levels. The excessive -level factors usually are: a) Utility revealing the easiness, reliability and efficiency people of a computer software product; b) maintainability in which describe the facilities to modify, the testability along with the aspects of understanding; c) portability inside sense of having the capacity to continue being used with a change connected with environment.



C. Dromey

Dromey has built a quality evaluation framework that analyzes the quality of software *components* through the measurement of tangible quality properties. Each artifact produced in the software lifecycle can be associated with a quality evaluation model. However, there is no discussion of how this can be done in practice, and this theoretical model is used to design others more specific models.


Product model
Component A
Tangible quality carrying properties
Component B
Tangible quality carrying properties
Linkages
High level quality attributes



QUALITY MODEL	MEASURMENT TECHNIQUE	ADVANTAGE
MC CALL MODEL	*Formulation*. The derivation of software measures and metrics appropriate for the representation of the software that is being considered. *Collection*. The mechanism used to accumulate data required to derive the formulated metrics. *Analysis*. The computation of metrics and the application of mathematical tools.	This method is very useful as it includes complete detail of all the quality aspects.
Boem model	RADC's Methodology: It is based on continuing development effort.	The requirements present a ratio of actual occurrence to the possible number of occurrence for each situation: these results in a clear correlation between the quality criteria and their associated factors.
Dromy model	No practical implementation. Based on evaluation criteria.	This model helps in predicting defects and indicates the properties that were violated in order to create defects.

D. FURPS/FURPS

The later, and probably somewhat less renown, model that can be structured in this is the same manner since the previous two excellent models (but still worth at least to be mentioned in this particular context) is your FURPS model originally presented by Robert Grady [15] (and expanded by Rational Computer software [16-18] - now IBM Rational Computer software - into FURPS+3). FURPS is short for:

- Functionality – which may include feature pieces, capabilities and safety?
- Usability - which may include human elements, aesthetics, consistency inside user interface, on the web and contextsensitive guide, wizards and agents, user documentation, along with training materials?
- Reliability - This may include frequency along with severity of failing, recoverability, predictability, accuracy and reliability, and mean moment between failures (MTBF?)
- Performance -- imposes conditions on functional requirements including speed, efficiency, availability, accuracy, throughput, answer time, recovery moment, and resource usage
- Supportability - which may include testability, extensibility, customization, maintainability, compatibility, configurability, serviceability, installability, localizability (internationalization)?

E. Comparative study:

The Comparative study is given in the figure at the top of the page.

IV. CONCLUSION

This paper is about exploring quality. The idea of quality is represented in various perspectives as each user satisfaction along with user demand. A comparative study of all quality models is actually briefly discussed along with their technique used to measure the coffee quality. Each model has a quality factors which are evaluated based in different criteria. Future research is usually extend and improve the methodology to extend metrics that were validated using unique criteria.

REFERENCES

[1] Adey, C. A. & Hill, G. K. (2000). *Quality / ISO 9000 as a Marketing Tool*, [En ligne]. http://www.smps.org/mrc/articles/0200qualityiso.pdf.

[2] Bazzana, G., Anderson, O., & Jokela, T. (1993). *ISO 9126 and ISO 9000: Friends or foes?* Presented at Software Engineering Standards Symposium.

[3] Biehl, R. E. (2001). *Six sigma for Software*. IEEE Software, 21(2), 68-70..

[4] Basili, V.R., & Selby, R.W. Calculation and use of an environment's characteristic software metric set, pp. 386-393, *Proceedings of the 8th Int.Conf. on Software Engineering*, London, 1985].

[5] Briand, L. et al. On the application of measurement theory in software engineering, *Empirical Software Engineering*,1996,vo.1.

[6] McCall JA, Richards PK, Walters GF. Factors in software quality, RADC TR-77-369:1977. (Rome: Rome Air Development Center)

[7] Boehm BW, Brow JR, Lipow M, McLeod G, Merritt M. Characteristics of software quality. North Holland Publishing. Amsterdam, the Netherlands; 1978.

[8] Grady, RB. Practical software metrics for project management and process improvement, Prentice Hall; 1992.

[9] Dromey RG. Concerning the Chimera (software quality). IEEE Software. 1996;1:33-43.

[10] Drown DJ, Khoshgoftaar TM, Seiya N. Evaluation any sampling and software qualitymodel of high assurance systems, IEEE Transaction on systems, Mean and Cybernetics, Part A: Systems and Human. 2009;39(5):1097-1107.

[11] Tomar AB, Thakare VM. A systematic study of software quality models, International Journal of software engineering & application. 2011;12(4):61-70.

[12] McCall, J. A., Richards, P. K., & Walters, G. F. (1977). *Factors in software quality*. Griffiths Air Force Base, N.Y.: Rome Air Development Center Air Force Systems Command.

[13] Hoyer, R.W. and Hoyer, B.B.Y, "What is quality?" Quality Progress, Volume 7, pp. 52-62, 2001

[14] Deepshikha Jamwal, “Analysis of Quality Models for Organizations”, International Journal of Latest Trends in Computing, Volume 1, Issue 2, December 2010.

[15] Divya Prasad Narayani, Poonam Uniyal, “Comparative Analysis of Software Quality Models”, International Journal of Computer Science and Management Research”, Volume 2, Issue 3, March 2013.