2010 2nd International Conference on Software Technology and Engineering(ICSTE)

Quality Models: Role and Value in Software Engineering

Durgesh Samadhiya
Ph.D program of Technology Management
Chung Hua University, Taiwan
samadhiya.durgesh@gmail.com

Su-Hua Wang
Dept of Information Management
Chung Hua University, Taiwan
swang@chu.edu.tw

Dengjie Chen
Ph.D program of Technology Mgt
Chung Hua University, Taiwan
m9310001@chu.edu.tw

***Abstract*—Software quality is the totality of features and characteristics of a product or a service that bears on its ability to satisfy the given needs. Poor quality of the software product in sensitive systems may lead to loss of human life, permanent injury, mission failure, or financial loss. So the quality of the project should be maintained at appropriate label. To maintain the quality, there are different quality models. "A high quality product is one which has associated with it a number of quality factors. These could be described in the requirements specification; they could be cultured, in that they are normally associated with the artifact through familiarity of use and through the shared experience of users. In this paper, we will discuss all the quality models: McCall's quality model, Boehm's quality model, Dromey's quality model, and FURPS quality model and focus on a comparison between these models, and find the key differences between them.**

Keywords-Quality Model, Software Quality, Implementing, Quality factors, Software Engineering.

I. INTRODUCTION:

Error-free running software is an important quality issue for the user. But the past has shown that correctness of the code is not the only quality attribute. Software maintenance consists of failure correction, performance enhancement, and adoption to a new technical environment, whereas software enhancement consists of the introduction of new features and adoption to changes in the industry. All these activities require changes in the source code, which become larger and more complex and, as a direct consequence, more expensive with the progressing age of the software. From this point of view "good" software should be easy to change and to expand. Flexible software suits the needs of an agile development, as well. As we can see, there are different aspects which come together and form software quality.

There are several definitions for "Software Quality" term, for example, Quality is defined by International organizations as "Quality comprises all characteristics and significant features of a product or an activity which relate to the satisfying of given requirements" [4], and "Quality is the totality of features and characteristics of a product or a service that bears on its ability to satisfy the given needs"[5].

In software Engineering, there are so many types of software quality models, each one of them have different quality characteristics and features or factors. In this paper, we will discuss and compare the quality models and find the key features between them 1.McCall's Quality Model, 2.Boehm's Quality Model, 3.Dromey's Quality Model, 4.FURPS Quality Model, 5.ISO 9126 Quality Model.

II. SOFTWARE QUALITY MODELS:

We define quality assessment model as analytical models that provide quantitative assessment of selected quality characteristics or sub-characteristics based on measurement data from software projects. Such models can help us obtain an objective assessment of our current product quality, in contrast to the often unreliable subjective assessment based on personal judgment or imprecise qualitative assessment.

A. McCall's Quality Model

One of the more renowned predecessors of today's quality models is the quality model presented by Jim McCall et al (1977) [1-3] also known as the General Electrics Model of 1977 originates from the US military (it was developed for the US Air Force) and is primarily aimed towards the system developers and the system development process. In this quality model McCall attempts to bridge the gap between users and developers by focusing on a number of software quality factors that reflect both the user's views and the developers' priorities. McCall quality model has three major perspectives for defining and identifying the quality of a software product: product revision (ability to undergo changes), product transition (adaptability to new environments) and product operations (its operation characteristics). Table 1 illustrates the all 3 Major Perspective and 11 Quality Factors and 23 Quality criteria of McCall Quality Model.

TABLE I. THE CONTENTS OF MCCALL'S QUALITY MODEL - PRODUCT REVISION AND PRODUCT OPERATIONS

Major	Quality	Quality Criteria

978-1-4244-8666-3/10/$26.00 © 2010 IEEE

Perspective	Factors	
Product Revision	Maintainability	Simplicity
		Conciseness
		Self-descriptiveness
		Modularity
	Flexibility	Self-descriptiveness
		Expandability
		Generality
	Testability	Simplicity
		Instrumentation
		Self-descriptiveness
		Modularity
Product Operation	Correctness	Traceability
		Completeness
		Consistency
	Efficiency	Execution efficiency
		Storage efficiency
	Reliability	Consistency
		Accuracy
		Error Tolerance
	Integrity	Access Control
		Access Audit
	Usability	Operability
		Training
		Communicativeness
Product Transition	Portability	Self-descriptiveness
		Software-System Independence
		Machine Independence
	Reusability	Self-descriptiveness
		Generality
		Modularity
		Software-System Independence
		Machine Independence
	Interoperability	Modularity
		Communication Commonality
		Data Commonality

B. *Boehm's Quality Model:*

The second of the basic and founding predecessors of today's quality models is the quality model presented by Barry W. Boehm (1978) [6, 7]. Barry W. Boehm also defined a hierarchical model of software quality characteristics, in trying to qualitatively define software quality as a set of attributes and metrics (measurements).

At the highest level of his model, Boehm defined three primary uses (or basic software requirements), these three primary uses are:-

- **As-is utility**, the extent to which the as-is software can be used (i.e. ease of use, reliability and efficiency).
- **Maintainability**, ease of identifying what needs to be changed as well as ease of modification and retesting.

Portability, ease of changing software to accommodate a new environment.

These three primary uses and associated quality factors and these quality factors are further broken down into Primitive constructs that can be measured, measure the lowest level of the model [8]

- Portability (General utility characteristics): Code possesses the characteristic portability to the extent that it can be operated easily and well on computer configurations other than its current one.
- Reliability (As-is utility characteristics): Code possesses the characteristic reliability to the extent that it can be expected to perform its intended functions satisfactorily.
- Efficiency (As-is utility characteristics): Code possesses the characteristic efficiency to the extent that it fulfills its purpose without waste of resources.
- Usability (As-is utility characteristics, Human Engineering): Code possesses the characteristic usability to the extent that it is reliable, efficient and human-engineered.
- Testability (Maintainability characteristics): Code possesses the characteristic testability to the extent that it facilitates the establishment of verification criteria and supports evaluation of its performance.
- Understandability (Maintainability characteristics): Code possesses the characteristic understandability to the extent that its purpose is clear to the inspector.

C. *Dromey's Quality Model:*

Dromey states that quality characteristics or high-level attributes cannot be built directly into software (1995) [9]. This product base quality model presented by R. Geoff Dromey [10, 11] recognizes that the quality evaluation differs for each product and that a more dynamic idea for modeling the process is needed to be wide enough to apply for different systems. Dromey is focusing on the relationship between the quality attributes and the sub-attributes, as well as attempting to connect software product properties with software quality attributes. Fig. 1 shows the all three principal elements to Dromey's generic quality model:

Product properties that influence quality

1. High level quality attributes
2. Means of linking the product properties with the quality attributes.

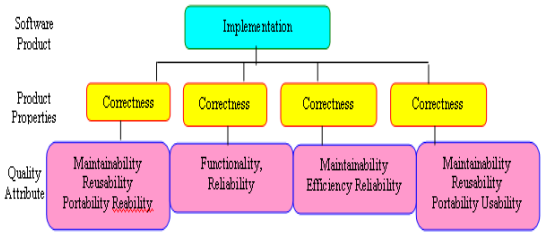

Figure 1. Principles of Dromey's Quality Model

D. FURPS Quality Model:

Robert Grady and Hewlett Packard proposed this model in 1987. FURP model decomposes characteristics in two categories of requirement namely functional requirement and non-functional requirement. Functional requirement are defined by input and expected output, while non functional requirement (also known as URPS) include usability, reliability, performance and superoperatibility. One disadvantage of the FURPS model is that it fails to take into account the software product's Portability [12]. Fig 2 shows all the characteristics of FURPS model.

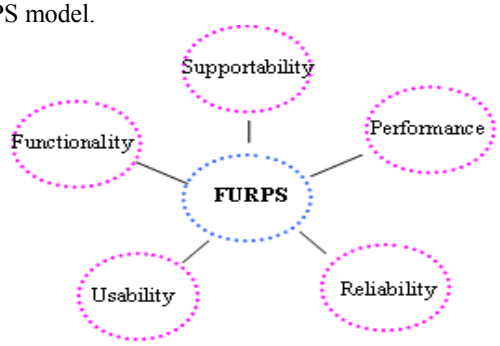

Figure 2. FURPS model

When the FURPS model is used, two steps are considered: Setting priority and defining quality attributes that can be measured. Grady and Caswell note that setting priorities is important given the implicit trade-off, i.e. one quality characteristics can be obtained at the expense of another. [table 2].

TABLE II. THE CONTENT OF FURPS MODEL

Characteristics	Description
Functionality	Include feature sets, capabilities, and security.
Usability	Human Factors, overall aesthetics, consistency, and documentation
Reliability	Frequency and severity of failure, recoverability, predictability, accuracy, and mean time between failures (MTBF).
Performance	Processing speed, response time, resource consumption, throughput and efficiency.
Supportability	Testability, extensibility, adaptability, maintainability, compatibility, configurability, serviceability, installability, and localizability.

E. ISO: The International Organization for Standardization is an international-standard-setting body composed of representatives from various national standards organizations. Founded on 23 February 1947, the organization promulgates worldwide proprietary industrial and commercial standards [13]. ISO also publishes Technical Reports, Technical Specifications, Publicly Available Specifications, Technical Corrigenda, and Guides [14, 15].

ISO 9126 Quality Model.: ISO 9126 identifies the external quality characteristics of a software product. Therefore it represents product-effectiveness. Fig 3 show the ISO model.

This standard was based on the McCall and Boehm models. Besides being structured in basically the same manner as these models, ISO 9126 also includes functionality as a parameter, as well as identifying both internal and external quality characteristics of software products. Fig 4 explains the factor, and sub factors.

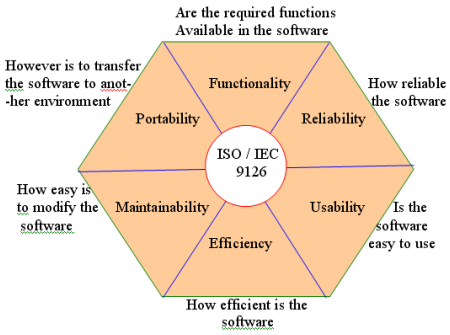

Figure 3. The ISO 9126 quality model

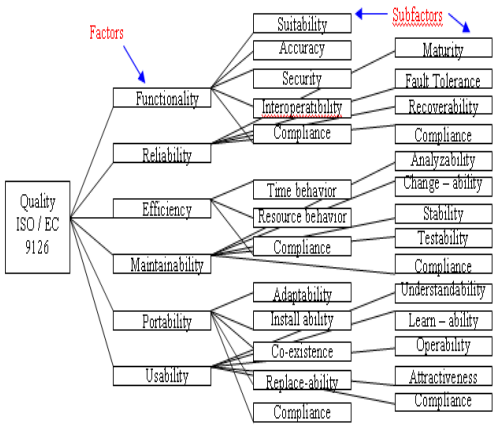

Figure 4. ISO 9126: Software Product Evaluation: Quality Characteristics and Guidelines for their Use.

III. COMPARISON:

The quality characteristics found in the majority of the models are: Efficiency, Reliability, Mainability, Portability, Usability and Functionality, these have been present in more recent models, as described above. In order to examine, compare and come to a conclusive result, we have prepared Table below, each associated with its software quality characteristics. With this tabular illustration, it becomes easy to observe the models that support a wider range of

characteristics in comparison to the ones that support fewer features. For example, it can be seen clearly that ISO 9126 and McCall support more characteristics than Boehm, FURPS and Dromey. Furthermore we can observe the common characteristics that are supported by almost all of the models, namely: Efficiency, Reliability, Functionality, Mainability, Portability, and Usability.[Fig 5, table 3]

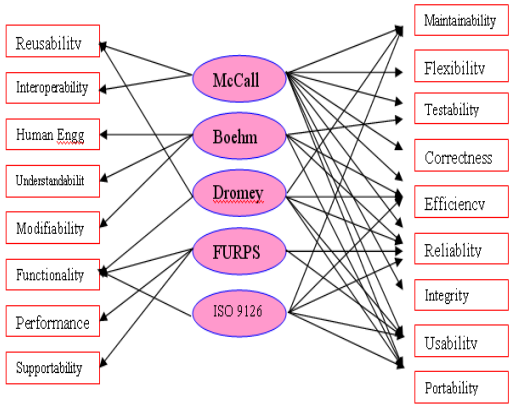

Figure 5. Comparative model

IV. ANALYSIS:

There are total 17 characteristics in all the models. Only one characteristic "Reliability" is common in all quality models. Three characteristics "Efficiency, Usability, Portability" belonging to four quality models. Two characteristics "Testability and Reusability" are belong only to two models. And in rest of the 9 characteristics "Flexibility, Correctness, Integrity, and Interoperability" in McCall's quality models, "Human engineering, Understandability, and Modifiability" in Boehm's quality model, "Performance and Supportability" in FURPS quality model. Testability, Interpretability, Understandability are used as factor/attributes/characteristics in some quality models. However in ISO 9126, these factor/attributes/characteristics are defined as sub characteristics. Specifically the testability is belonging to the maintainability characteristic, the understandability is belonging to the usability and the interoperability is belonging to the functional characteristic.

McCall model is mostly useful for a bottom to top approach to software quality and it can be effectively used to define measures of software quality, but is more difficult to use to specify quality requirement. Boehm's model is a step forward in the sense that it provides basic support for a top to bottom approach to software quality, this support is too ephemeral to be considered as a solid foundation for quality engineering. Boehm's model is decomposed in a hierarchy that at the top addresses the concerns of end-users while the bottom is of interest to technically inclined personnel. It is in effect the emergence of the user perspective of quality. In McCall's Quality Model, the quality factors synthesized provide a complete software quality picture [16]

TABLE III. QUALITY CHARACTERISTICS IN BOEHM, MCCALL, FURPS, ISO 9126 AND DROMEY MODELS

Factor/Attributes/Characteristics	McCall	Boehm	Dromy	FURPS	ISO 9126
Maintainability	Y		Y		Y
Flexibility	Y				
Testability	Y	Y			
Correctness	Y				
Efficiency	Y	Y	Y		Y
Reliability	Y	Y	Y	Y	Y
Integrity	Y				
Usability	Y		Y	Y	Y
Portability	Y	Y	Y		Y
Reusability	Y		Y		
Interoperatability	Y				
Human Engineering		Y			
Understandability		Y			
Modifiability		Y			
Functionality			Y	Y	Y
Performance				Y	
Supportability				Y	
17	11	7	7	5	6

Though Boehm's and McCall's models might appear very similar, the difference is that McCall's model primarily focuses on the precise measurement of the high-level characteristics, whereas Boehm's quality model is based on a wider range of characteristics with an extended and detailed focus on primarily maintainability. Boehm focuses a lot on the models effort on software maintenance cost effectiveness – which, he states, is the primary payoff of an increased capability with software quality considerations.

The FURPS-categories are of two different types: Functional (F) and Non-functional (URPS). These categories can be used as both product requirements as well as in the assessment of product quality. [17] The disadvantage of the FURPS model is that it fails to take into account the software product's Portability.

As ISO 9126 model has 6 important areas of importance of software evaluation so ISO 9126 quality model is the good model for software process. The first document of the ISO 9126 series –Quality Model – contains two-part quality model for software product quality [ISO, 2001]:

1. Internal and external quality model.
2. Quality in use model.

The first part of the two-part quality model determines six characteristics in which they are subdivided into twenty-seven sub-characteristics for internal and external quality, as in table [ISO, 2001].These sub-characteristics are a result of internal software attributes and are noticeable externally when the software is used as a part of a computer system. ISO defines the important product quality as a set of product characteristics, these are:

- External characteristics: How the product works in its environment e.g. Usability, Reliability.
- Internal characteristics: How the product was developed e.g. size, test and failure rate

Dromey model is a product based quality model that recognizes that quality evaluation differs for each product and that a more dynamic idea for modeling the process is needed to be wide enough to apply for different systems. Dromey is focusing on the relationship between the quality attributes and the sub-attributes, as well as attempting to connect software product properties with software quality attributes.

V. CONCLUSION

The paper has shown that quality can be a very elusive concept that can be approached from a number of perspective dependent on once take and interest. Most of the quality models presented within this technical paper probably could be fitted within the user view, manufacturing view or product view. The models presented herein are focused around either processes or capability level where quality is measured in terms of adherence to the process or capability level, or a set of attributed/metrics used to distinctively assess quality (McCall, Boehm etc.) by making quality a quantifiable concept. This structure of quality is in great contrast to the dynamic, moving target, fulfilling the customers' ever changing expectations perspective presented by some of the quality management experts. These quality characteristics could be used to reflect the quality of the software product from the view of that characteristic. Selecting which one of the quality models to use is a real challenge.

In this paper, we conclude that McCall's and Dromey quality model focus on the product perspective of quality to the detriment of other perspectives. Furthermore, they are primarily useful in a bottom up approach to quality that is not suitable for Software Quality Engineering. ISO/IEC 9126 is the only model that supports all the perspectives of quality (with the exception of the transcendental perspective as noted). Furthermore, its predictive framework clearly supports both the top down and bottom up approaches. ISO 9126-1 quality model as sub-characteristics from other characteristics, FURPS quality model is built and extended to be used in the IBM Rational Software Company. Therefore, it is a special-purpose quality model, that is, for the benefits of that company,

The metrics in the lower level of the McCall's, Boehm's, Doromey's and FURPS quality models are neither clearly nor completely defined and connected to the upper level of the quality models. ISO/IEC 9126 seems well suited for Software Quality Engineering. Further research is needed to see if the measures associated with ISO/IEC 9126 make this model usable for Software Quality Engineering in practice.

REFERENCES:

[1] McCall, J. A., Richards, P. K., and Walters, G. F., Factors in Software Quality", Nat'l Tech.Information Service, no. Vol. 1, 2 and 3, 1977.

[2] Marciniak, J. J., Encyclopedia of software engineering, 2vol, 2nd ed., Chichester : Wiley, 2002.

[3] Kitchenham, B. and Pfleeger, S. L., "Software quality: the elusive target [special issues section]", IEEE Software, no. 1, pp. 12-21, 1996.

[4] German Industry Standard DIN 55350 Part 11

[5] ANSI Standard (ANSI/ASQC A3/1978)

[6] Boehm, B. W., Brown, J. R., Kaspar, H., Lipow, M., McLeod, G., and Merritt, M., Characteristics of Software Quality, North Holland, 1978, Boehm, Barry W

[7] Brown, J. R, and Lipow, M.: Quantitative evaluation of software quality, International Conference on Software Engineering, Proceedings of the 2nd international conference on Software engineering, 1976.

[8] Adapted from Pfleeger (2003), Boehm et al. (1976; 1978)

[9] Dromey, R., "Comering the Chimera", 13 (1): 33-43, IEEE Software, January 1996.

[10] Dromey, R. G., "Concerning the Chimera [software quality]", IEEE Software, no. 1, pp. 33-43, 1996.

[11] Dromey, R. G., "A model for software product quality", IEEE Transactions on Software Engineering, no. 2,pp. 146-163, 1995.

[12] .Maryoly, O., M.A. Perez and T. Rojas, 2002. A systemic quality model for evaluating software products. Laboratorio de Investigcin en Sistemas de Informacin, 2002, http:/www.lisi.usb.ve/publicaciones.

[13] Discover ISO – Meet ISO". ISO. © 2007. http://www.iso.org/iso/about/discover-iso_meet-iso.htm. Retrieved 7 September 2007.

[14] The ISO directives are published in two distinct parts: "ISO Directives, Part 2: Rules for the structure and drafting of International Standards. 5th Edition" (pdf). ISO/IEC. 2004. http://www.iec.ch/tiss/iec/Directives-Part2-Ed5.pdf Retrieved 7 September 2007 .

[15] ISO. "ISO/IEC Directives and ISO supplement". http://www.iso.org/directives. Retrieved 1 January 2010.

[16] .Kitchenham, B. and Pfleeger, S. L., "Software quality: the elusive target [special issues section]", IEEE Software, no. 1, pp. 12-21, 1996.

[17] Grady, R. B., Practical software metrics for project management and process improvement, Prentice Hall, 1992.