Software Quality Models: A Comparative Study

Gauri Shankar[1], Er. Laxmi Shanker Maurya[2]

[1]*Research Scholar, SRMSCET, Bareilly*
[2]*Associate Professor, Department of Information Technology, SRMSCET, Bareilly*

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. Software quality model is the representation of the characteristics of the software that describe the quality. The benefit of quality models is that they are simpler to use and the benefit of quality management is that they probably more to the point capture the idea of quality. 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 compare all the quality models: McCall's quality model, Boehm's quality model, Dromey's quality model, and ISO 9126 quality model and focus on a comparison between these models, and find the key differences between them.

Keywords: Software Quality, Quality factors, Software Engineering.

1. INTRODUCTION

In software, the narrowest sense of quality is commonly recognized as lack of "bugs" in the product. It is also the most basic meaning of conformance to requirements, because if the software contains too many functional defects, the basic requirement of providing the desired function is not met[18]. Error free running software is an important quality issue for the user. But the past has shown that correctness. But in previous years correctness of code is not only the quality attribute. Software maintenance consist of failure corrections, performance improvements and enhancement according to new technology or technical environment, introduction to new features and adoption of changes in new environment in industry. These changes takes place when we make changes in source code, this process becomes expensive and complex when source code is comparatively longer. Longer source code are not easy to maintain because they become more complex in structure more expensive with the progressive age of software. For these point of view "good" or " quality" software should be easy to change and easy to expand. There are different aspect which comes together and forms software quality.

There are many 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].Software quality IEEE definitions:[19]

1. The degree to which a system, component, or process meets specified requirements.

2. The degree to which a system, component, or process meets customer or user needs or expectations.

In software engineering there are so many quality model defined each of 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.ISO 9126 Quality Model.

2. QUALITY M ODELS IN SOFTWARE ENGINEERING

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 Electrical 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

Table 1..
THE CONTENTS OF Mc CALL'S QUALITY MODEL

Major Perspective	***Quality factors***	***Quality Criteria***
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	Portability	Self-descriptiveness
		Software - System Independence
		Machine Independence
		Self-descriptiveness

Transition	Reusability	Generality
		Modularity
		Software - System
		Independence
		Machine Independence
	Interoperability	Modularity
		Communication
		Commonality
		Data Commonality

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.

B. Boehm' s Quality Model:

The second after Mc Call's quality models 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:-

I. **As-is utility**, the extent to which the as-is software can be used (i.e. ease of use, reliability and efficiency).

a) Reliability: Code possesses the characteristic reliability to the extent that it can be expected to perform its intended functions satisfactorily.

b) Efficiency: Code possesses the characteristic efficiency to the extent that it fulfils its purpose without waste of resources.

c) Usability: Code possesses the characteristic usability to the extent that it is reliable, efficient and human-engineered.

II. **Maintainability**, ease of identifying what needs to be changed as well as ease of modification and retesting.

a) Testability: Code possesses the characteristic testability to the extent that it facilitates the establishment of verification criteria and supports evaluation of its performance.

b) Understandability: Code possesses the characteristic understandability to the extent that its purpose is clear to the inspector.

III. **Portability**, ease of changing software to accommodate a new environment. Code possesses the characteristic portability to the extent that it can be operated easily and well on computer configurations other than its current one.

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.

D. ISO 9126 QUALITY MODEL:

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].

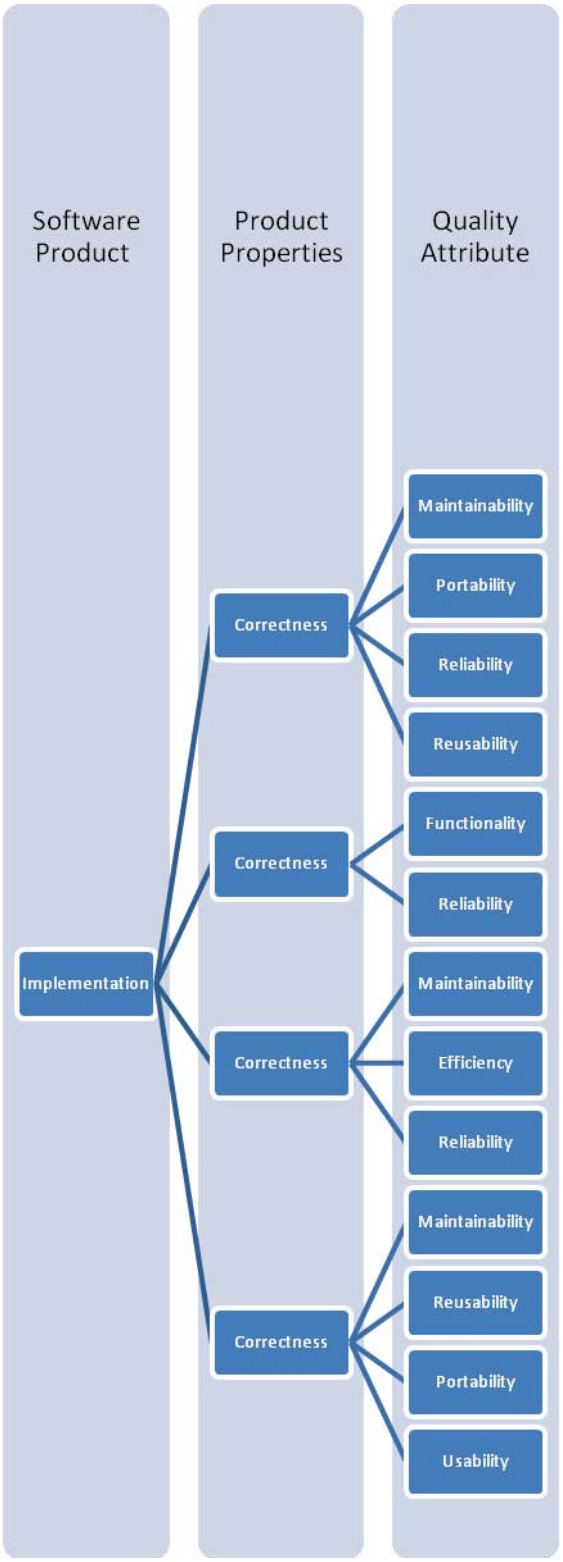

Figure 1. Principles o f Dromey's Quality Model

ISO 9126 identifies the external quality characteristics of a software product. Therefore it represents product-effectiveness. Figure 2 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 3 explains the factor, and sub factors.

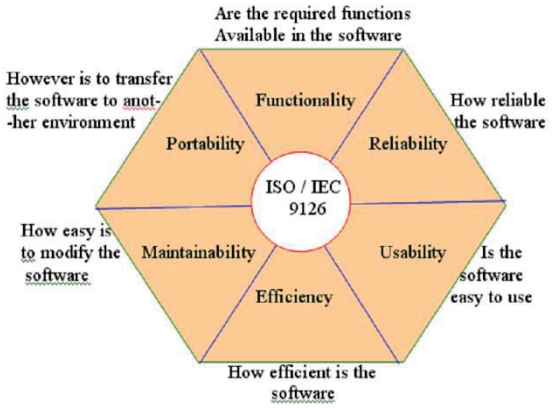

Figure 2 The ISO 9126 quality model

3. COMPARISON

There are total 17 characteristics in all the models. Efficiency, Reliability, Maintainability, Portability, Usability and Functionality, these have been present in more recent models, as described above. From the table it can be seen that ISO 9126 and McCall support more characteristics than Boehm and Dromey. Furthermore we can observe the common characteristics that are supported by almost all of the models, namely: Efficiency, Reliability, Functionality, Maintainability, Portability, and Usability.

4. ANALYSIS

From the table we can see that 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, 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.

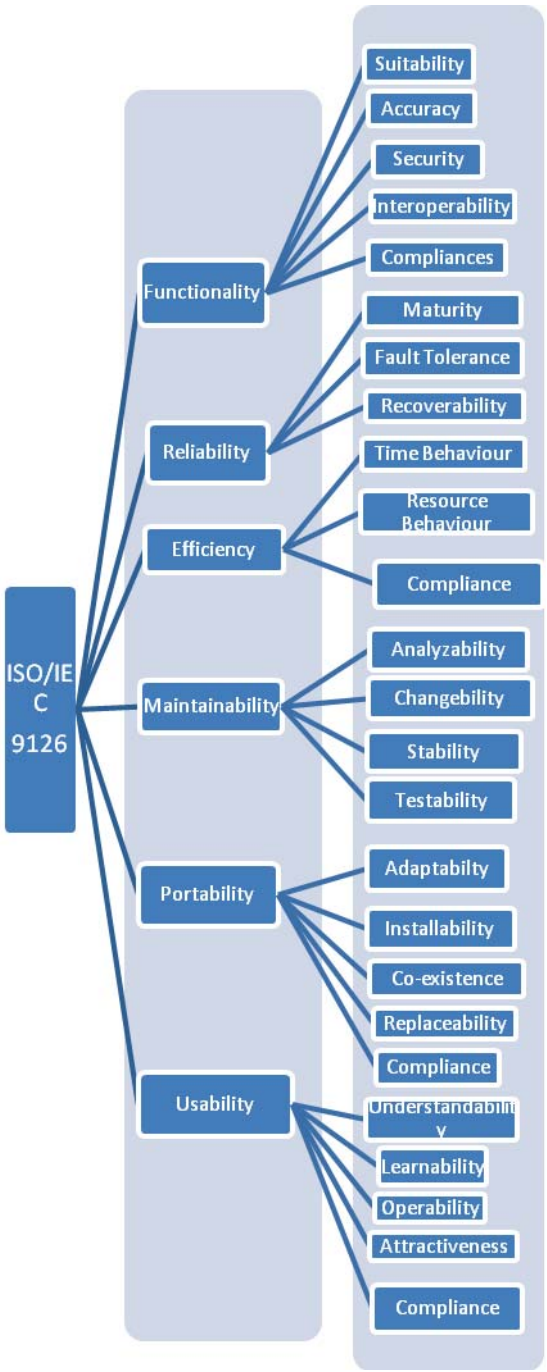

Figure3. ISO 912 6: Software Product Evaluation: Quality Characteristics and Guidelines for their Use.

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,ISO 9126 AND DROMEY MODELS

factor/ attributes/ characteristics	McCall	Boehm	Dromey	ISO/IEC 9126
Maintainability	√		√	√
Flexibility	√			
Testability	√	√		
Correctness	√			
Efficiency	√	√	√	√
Reliability	√	√	√	√
Integrity	√			
Usability	√		√	√
Portability	√	√	√	√
Reusability	√		√	
Interoperability	√			
Human – Engineering		√		
Understand-ability		√ √		
Modifiability			√	√
Functionality				
Performance				
Supportability				
17	11	7	7	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 means software maintenance cost effectiveness is the primary reward of an increased capability with software quality considerations.

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.

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.

5. CONCLUSION

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/IEC9126 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.

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 Information, 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.

[18]. Metrics and Models in Software Quality Engineering, Second Edition By Stephen H. Kan, Addison Wesley, September 2002

[19]. Software Quality Assurance , From theory to implementation By Daniel Galin, Addison Wesley,2004