Comparative Study of Software Quality Models

BOUKOUCHI Youness[1], MARZAK Abdelaziz[2], BENLAHMER Habib[3] and MOUTACHAOUIK Hicham [4]

[123] Department of mathematics and computer Science, Hassan II Ben M'Sik University, Faculty of Sciences, Casablanca, Morocco
[4]Laboratory IRF-SIG, IBN ZOHR University, Faculty of Science, Agadir, Morocco

Abstract
Software quality models play an important role in the measurement of software quality. A number of qualities models are used to build quality software. Different researchers have proposed different software quality models to help measure the quality of software products. In our research, we are discussing the different software quality models and comparing the software quality models with each other.
Keywords : *Software Quality, Quality Model, Quality*

1. Introduction

Software quality plays an important role in the success of the overall software system. So it is considered as a very important aspect for the developers, users and project managers. There is a number of quality models in software engineering literature, each one of these quality models consists of a number of quality characteristics (or factors, as called in some models).
In this paper, we will present the contents and comparison of the following quality models: McCall's Quality Model, Boehm's Quality Model, Evans & Marciniak Model, Deutsch & Willis Quality Model, ISO 9126's Quality Model, Dromey's Quality Model, FURPS+ Quality Model, SEI Model and ISO 25000 (SQuaRE) Model.
The rest of this paper is structured as follows: Section 2 presents an overview of the nine common quality models used in software engineering. Section 3 contains a comparison between the five quality models. Finally, Section 4 concludes the paper with some comments.

2. Software quality

2.1. Definitions of Software quality

Software quality has been a major challenge since the onset of computer programs. As a result, a large number of definitions of software quality have emerged, some of them have been standardized, but most of them are perceived too vague and abstract. There are several definitions of software quality: ISO defines it as a set of attributes of a software product by which its quality is described and evaluated. ANSI Standard defines it as the totality of features and characteristics of a product or a service that bears on its ability to satisfy the given needs. IEEE Standards defines it as the totality of features and characteristics of a software product that bears on its ability to satisfy given needs.

2.2. Software quality Model

The models of software quality are representations abstract and simplified which touch or affect the software quality. There are two different types of models of software quality, that is the general models and the specific models:

- The general models are developed to be used with all the classes of existing software applications. So, the attributes of these models are chosen to be applicable to any software.
- The specific models are developed to be exclusively used with a class of software application in particular. The attributes of these models are thus chosen to cover all the aspects of the quality which are relevant in the considered class.

These models of software quality are conceived to identify the quality requirements and the criteria of acceptance for software so they allow to estimate and to guide the progress of the software development with regard to the quality criteria. Finally, these models facilitate the communication of the aspects of the quality to the customers, the users and the various groups of the team of development.
Then, to measure the quality, these models propose quality factors (named also characteristics) which are directly observable by the users. These factors will be estimated by means of several criteria (named also sub-characteristics or attributes) which are observable that by developers, and which are measured and estimated by means of one or several metrics

3. Software Quality Models

3.1. Mc Call's Quality Model (1977)

The first quality model was proposed by Jim Mc Call et al. This quality model defines and identifies 11 factors of the quality associated with 23 criteria of a software product through three perspectives: Product Revision (is the ability to undergo changes, including error correction and system adaptation.), Product Operation (is the product's ability to be quickly understood, operated and capable of providing the results required by the user) and Product Transition (is the adaptability to new environments, distributed processing together with rapidly changing hardware).

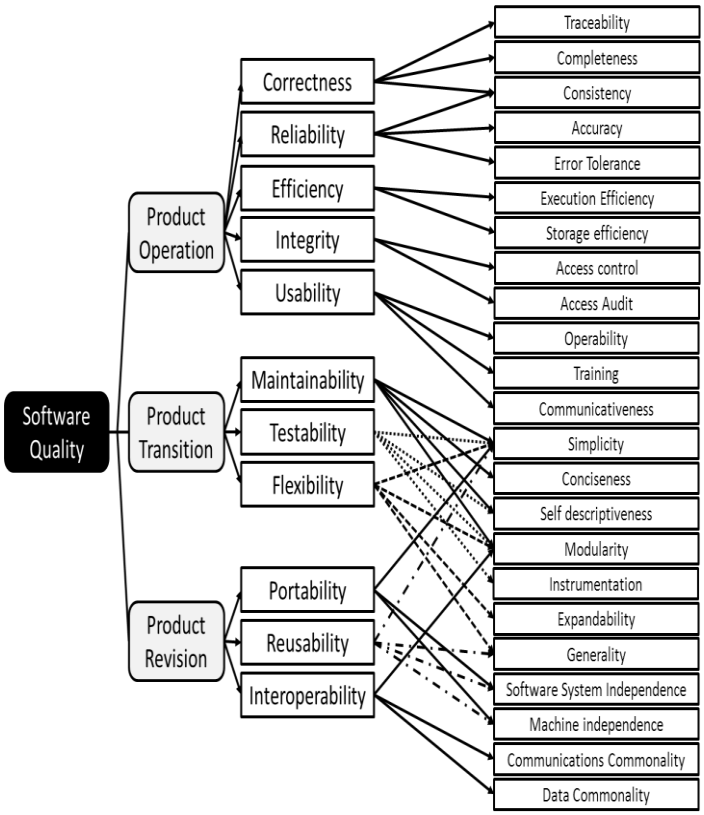

Figure 1. Mc Call's Model

3.2. BOEHM's Quality Model (1978)

Boehm and et al. defined the prime characteristic of quality as "general utility". Their quality model represents a hierarchical structure of characteristics, each of which contributes to the total quality. The high-level characteristics represent basic high-level requirements of actual use to which evaluation of software quality could be put. It includes as-is utility, maintainability and portability. The intermediate level characteristic, there are seven quality characteristics that together represent the qualities expected from a software system: Portability, Reliability, Efficiency, Usability, Testability, Understandability, and Flexibility.

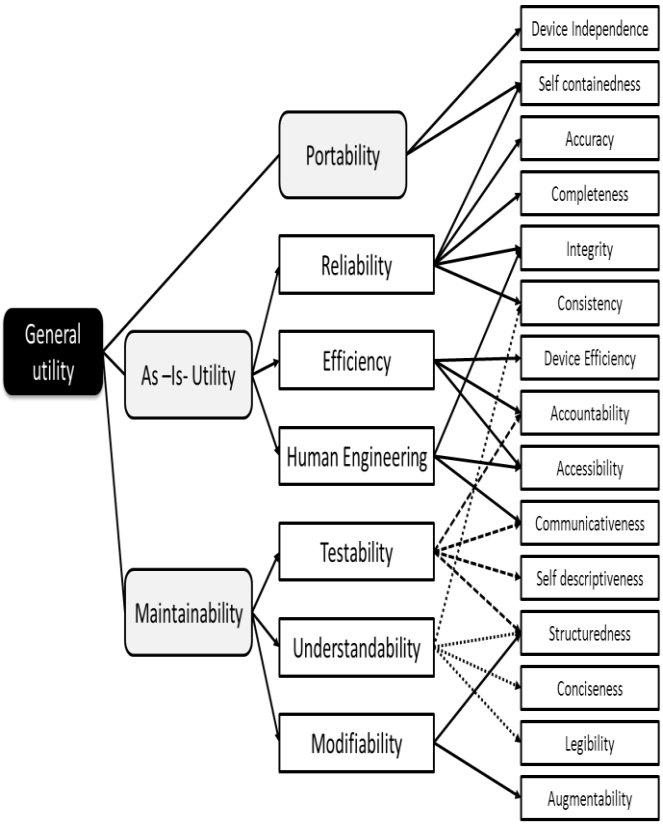

Figure 2. Boehm's Model

3.3. Evans & Marciniak Quality Model (1987)

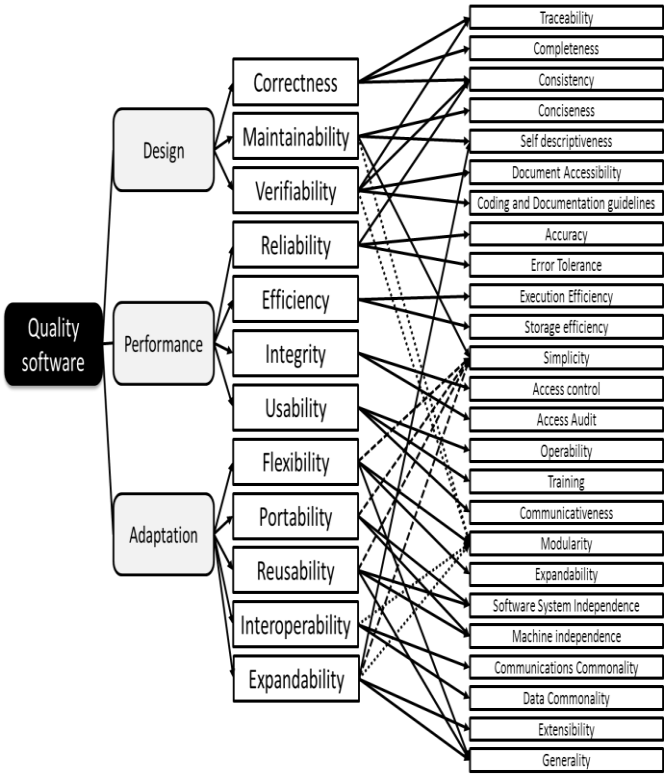

Figure 3. Evans & Marciniak's Model

The Evans and Marciniak model is an alternative model have emerged after the McCall model, it defines twelve factors that are grouped into three categories: design, performance and adaptation.

3.4. Deutsch & Willis Quality Model (1988)

The Deutsch and Willis model is an alternative model have emerged after the McCall model, it has fifteen factors grouped into four categories: operating, performance, change and management.

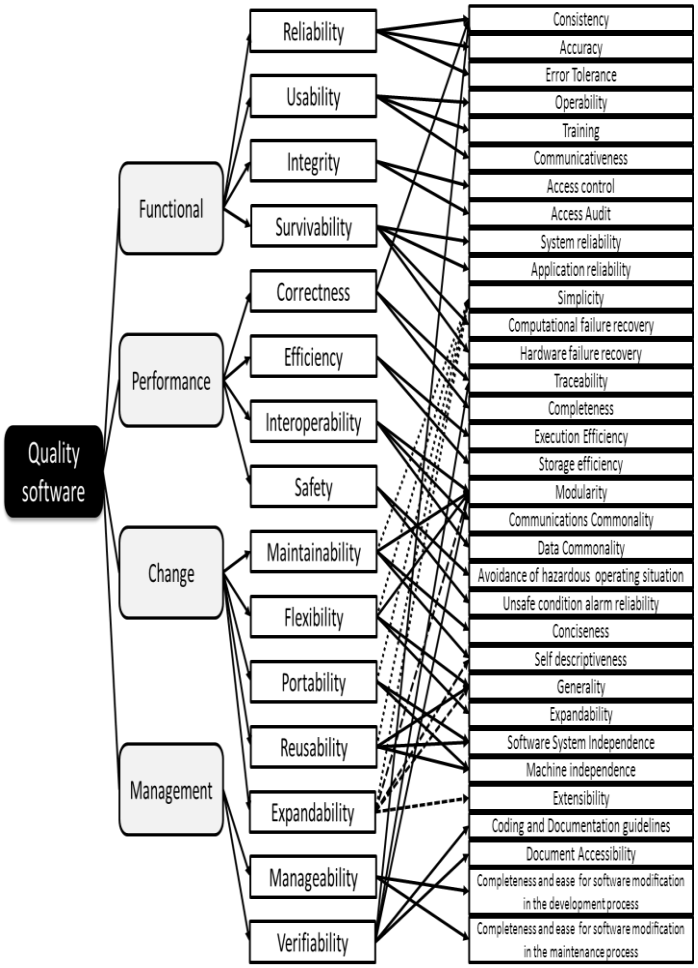

Figure 4. Deutsch & Willis's Model

3.5. ISO 9126's Quality Model (1991)

The ISO 9126 quality model was proposed as an international standard for software quality measurement. It was derived using the McCall model. ISO 9126-1 quality model has two main parts consisting of Internal and External Quality Attributes and Quality in Use Attributes. The Internal quality attributes refers to the properties of the system that can be evaluated without executing it while External quality attributes refers to the system properties that may be evaluated by observing the system during its execution. The quality in use attributes refers to the properties of the system that are experienced by the users of the system when the system is in operable condition and also during its maintenance. The characteristics of this model (Internal and external quality attributes) are Efficiency, Functionality, Maintainability, Portability, Reliability and Usability.

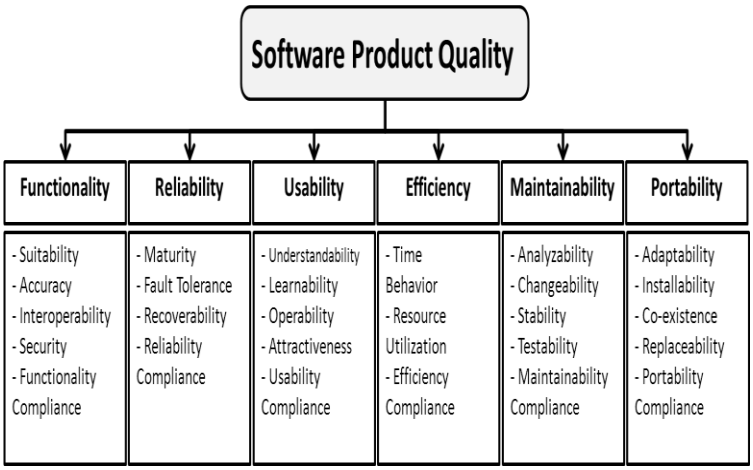

Figure 5. ISO 9126's Model

3.6. DROMY's Quality Model (1992)

Dromey's proposes a working framework for evaluating Requirement determination, design and implementation phases. The framework consists of three models: Requirement quality model, Design quality model and Implementation quality model. The high-level product properties for the implementation quality model include: (i) Correctness evaluates if some basic principles are violated, with functionality and reliability as software quality attributes; (ii) Internal measures how well a component has been deployed according to its intended use, with maintainability, efficiency and reliability as software quality attributes; (iii) Contextual deals with the external influences on the use of a component, with software quality attributes in maintainability, reusability, portability and reliability; (iv) Descriptive measures the descriptiveness of a component, with software quality attributes in maintainability, reusability, portability and usability.

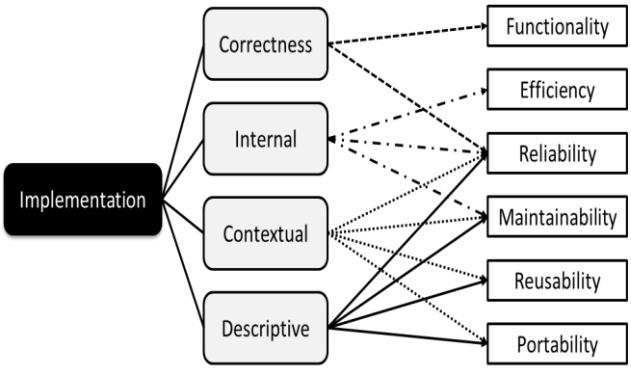

Figure 6. Dromy's Model

3.7. FURPS + 's Quality Model (1992)

The FURPS model was originally presented by Robert Grady at Hewlett Packard (and extended by Rational Software – now IBM Rational Software – into FURPS+). This model categorized characteristics into two different requirements such as Functional Requirements (F) which is defined by expected input & output and Non Functional Requirements in which U stands for Usability (includes human factors, aesthetic, documentation of user and material of training), R stands for Reliability (includes frequency and severity of failure, recovery to failure, time among failure), P stands for Performance (includes functional requirements) and S stands for Supportability (includes backup, requisite of design, implementation, interface and physiosts).
The "+" in FURPS+ stands for: Design requirements, Implementation requirements, Interface requirements and Physical requirements

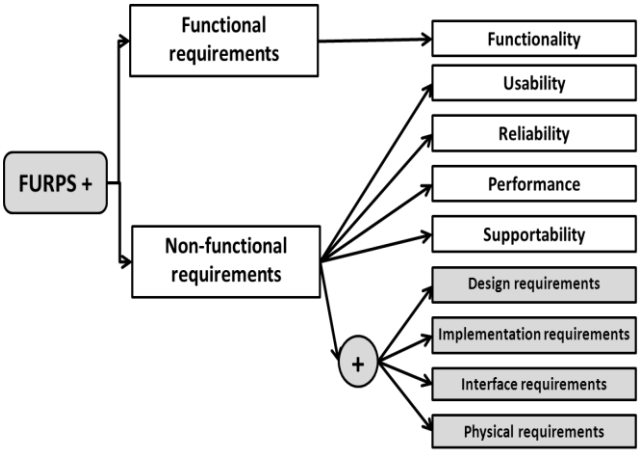

Figure 7. FURPS+'s Model

3.8. SEI Quality Model (1995)

Software Engineering Institute (SEI) published a report of Quality Attributes (Technical Report CMU/SEI-95-TR-021) in december 1995, The purpose of this report is to take a small step in the direction of developing a unifying approach for reasoning about multiple software quality attributes. This report examines the following four software quality attributes: performance, dependability, security, and safety.

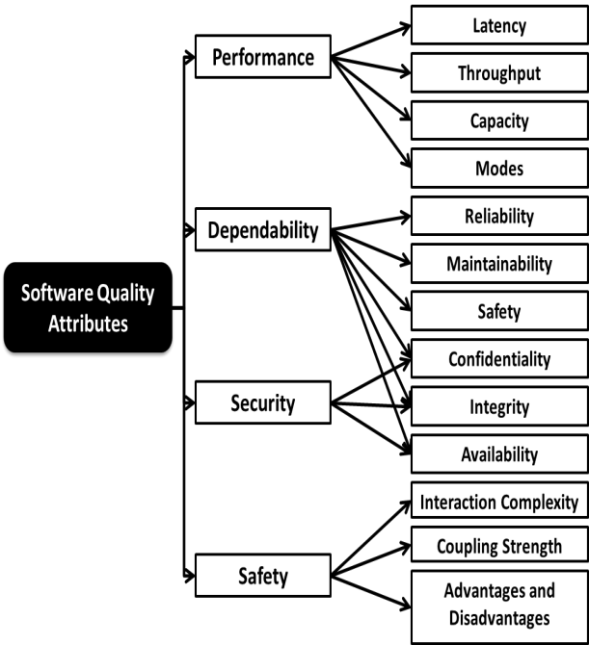

Figure 8. SEI's Model

3.9. ISO 25000 (SQuaRE) (2011)

The Systems and software Quality Requirements and Evaluation (SQuaRE) This International Standard is derived from ISO/IEC 9126:1991, it defines:

- A quality in use model composed of five characteristics (some of which are further subdivided into sub-characteristics) that relate to the outcome of interaction when a product is used in a particular context of use. This system model is applicable to the complete human-computer system, including both computer systems in use and software products in use.
- A product quality model composed of eight characteristics (which are further subdivided into sub-characteristics) that relate to static properties of software and dynamic properties of the computer system. The model is applicable to both computer systems and software products.

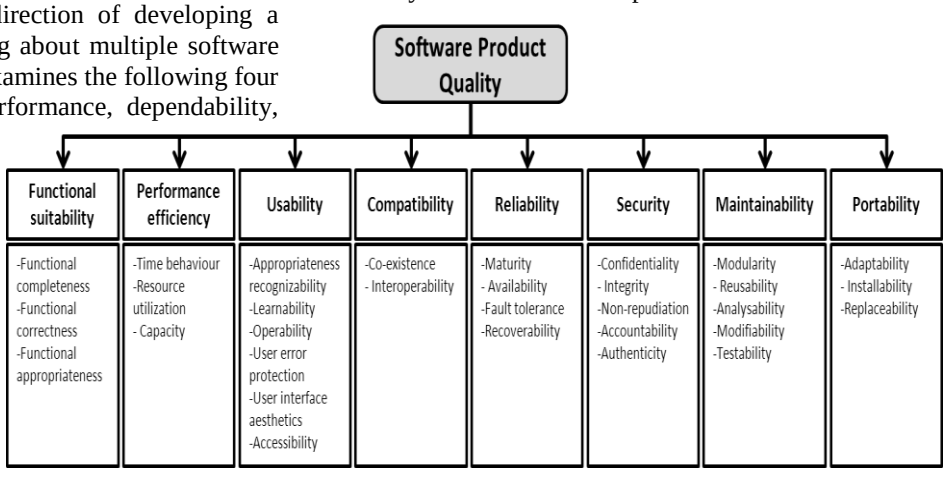

Figure 9. ISO SQuaRE's Model

4. Comparison

In this section, a comparison between the availability of the characteristics (called factors or attributes in some quality models) within the nine quality models will be presented: Table 1 presents this comparison, at the end this table you will find the number of the corresponding characteristics for each quality model and Figure 1 presents the frequency of each characteristic in this table. From the 28 characteristics, only one characteristic is common to eight quality models except ESI model that is the reliability. Also, there are only three characteristics (efficiency, usability and portability) which are belonging to seven models. One characteristic is common only to six quality models that is the Maintainability characteristic. Two characteristic belong to four quality models that is the Functionality and Reusability characteristics. Five characteristic belong to three quality models that is, the Performance, Interoperability, Integrity, Correctness and Flexibility characteristics. Four characteristic belong to two quality models (Safety, Expandability, Verifiability, and Testability). And, twelve characteristics (Manageability, Dependability, Security, Survivability, Supportability, Design Requirements, Implementation Requirements, Interface Requirements and Physical Requirements) are defined in only one quality model.

TABLE I. COMPARISON OF QUALITY MODELS

	Mac Call 1977	BOEHM (1978)	Evans & Marciniak (1987)	Deutsch & Willis (1988)	ISO 9126 (1991)	FURPS + (1992)	Dromey (1992)	SEI (1995)	SQuaRE (2011)
Maintainability	x		x	x	x		x		x
Flexibility	x		x	x					
Testability	x	x							
Correctness	x		x	x					
Efficiency	x	x	x	x	x		x		x
Reliability	x	x	x	x	x	x	x		x
Integrity	x		x	x					
Usability	x		x	x	x	x	x		x
Portability	x	x	x	x	x		x		x
Reusability	x		x	x			x		
Interoperability	x		x	x					
Human Engineering		x							
Understandability		x							
Modifiability		x							
Functionality					x	x	x		x[1]
Performance						x		x	x[1]
Supportability						x			
Design Requirements						x			
Implementation Requirements						x			
Interface Requirements						x			
Physical Requirements						x			
Verifiability			x	x					
Expandability			x	x					
Survivability				x					
Safety				x				x	
Manageability				x					
Dependability								x	
Security								x	
28	**11**	**7**	**12**	**15**	**6**	**9**	**7**	**4**	**8**

(1) For SQuaRE: Functionality is Functional suitability and Performance is Performance efficiency

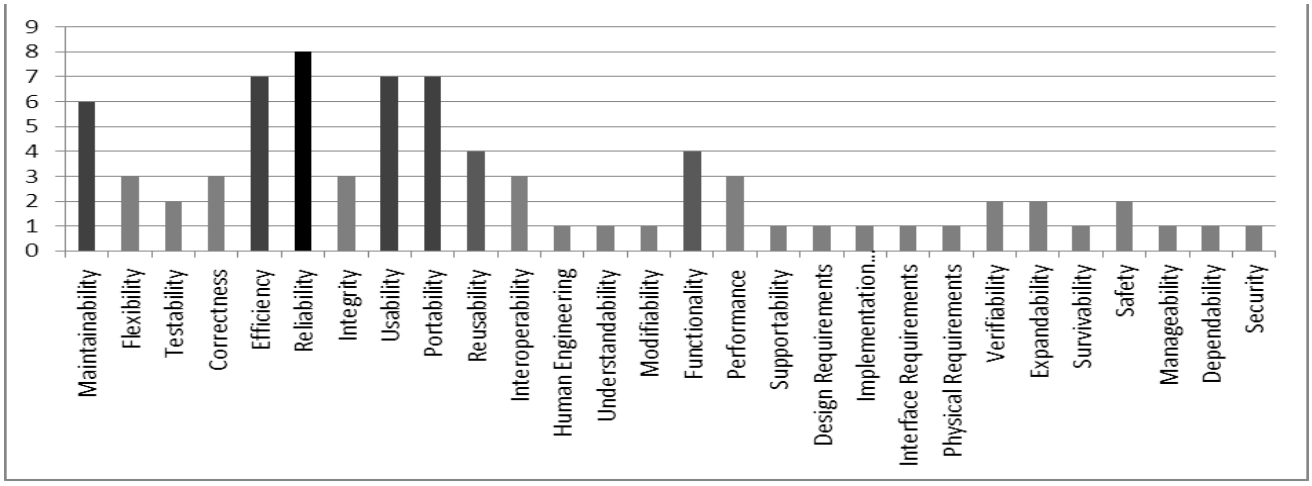

Figure 10. Frequency of characteristics appear in the nine quality models

5. Conclusion

We have studied different types of software quality models in software engineering; each of these quality models consists of numbers of characteristics to measure software quality. Selecting which one of the quality models to use is a real challenge. However Users have a direct impact on the Software Quality, and thus play a pivotal role in the measurement of Software Quality. So their needs of quality will decide which of the quality models be perfect to measure software quality, if not we have to define à new quality model through this characteristics. This is the objective of this study, it is to enumerate the models of quality as well as their characteristics and to present to the users and developers several solutions of quality measure so as to choose their models as well as factors appropriate to their needs

6. References

[1] Daniel Galin, 2004. Software Quality Assurance From theory to implementation. Pearson Education Limited

[2] ISO/IEC 25010:2011. Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models

[3] Khashayar Khosravi and Yann-Gaël Guéhéneuc, 2004, A Quality Model for Design Patterns

[4] Namita Malhotra, Shefali Pruthi, 2012. An Efficient Software Quality Models for Safety and Resilience. International Journal of Recent Technology and Engineering (IJRTE), ISSN: 2277-3878, Volume-1, Issue-3, August 2012

[5] Deepshikha Jamwal,Analysis of Software Quality Models for Organizations. International Journal of Latest Trends in Computing (E-ISSN: 2045-5364) Volume 1, Issue 2, December 2010

[6] Divya Prasad Naragani, Poonam Uniyal, Comparative Analysis of Software Quality Models. International Journal of Computer Science and Management Research, Vol 2 Issue 3 March 2013, ISSN 2278-733X

[7] Sanjay Kumar Dubey1, Soumi Ghosh, Rana ,Comparison of Software Quality Models: An Analytical Approach. International Journal of Emerging Technology and Advanced Engineering Website: www.ijetae.com (ISSN 2250-2459, Volume 2, Issue 2, February 2012)

[8] IEEE Std 830-1998, (Revision ofIEEE Std 830-1993). IEEE Recommended Practice for Software Requirements SpeciÞcations

[9] IEEE Std 1061-1998 (R2009) (Revision of IEEE Std 1061-1992). IEEE Standard for a Software Quality Metrics Methodology

[10] Alain Abran1, Rafa E. Al-Qutaish1, Jean-Marc Desharnais1 and Naji Habra2, ISO-BASED MODELS TO MEASURE SOFTWARE PRODUCT QUALITY

[11] Cem Kaner and Walter P. Bond, Software Engineering Metrics: What Do They Measure and How Do We Know? 10TH INTERNATIONAL SOFTWARE METRICS SYMPOSIUM, METRICS 2004

[12] Bogdan VINTILA, Quality Standards in Open Source Lifecycle, Open Source Science Journal Vol. 2, No. 1, 2010

[13] Mario Barbacci, Mark H. Klein, Thomas A. Longstaff, Charles B. Weinstock. Quality Attributes, Technical Report CMU/SEI-95-TR-021 ESC-TR-95-021

[14] Jeff Offutt, Quality Attributes of Web Software Applications, Focus Engineering Internet software, March/April 2002 IEEE SOFTWARE.

[15] Rafa E.Al-Qutaish, Quality Models in Software Engineering Literature: An Analytical and Comparative Study. Journal of American Science, 2010

[16] BARBARA KITCHENHAM and SHARI LAWRENCE PFLEEGER,SOFTWARE QUALITY: THE ELUSIVE TARGET. 1996 IEEE. Vol. 13, No. 1: JANUARY 1996, pp. 12-21

[17] Spector, A. Z. 1989. Achieving application requirements. In Distributed Systems, S. Mullender

[18] Patrik Berander, Lars-Ola Damm, Jeanette Eriksson and al., Software quality attributes and trade-offs. Blekinge Institute of Technology June 2005.

[19] Su-Hua Wang, Durgesh Samadhiya, Dengjie Chen. Software Quality: Role and Value of Quality Models.

[20] International Journal of Advancements in Computing Technology Volume 3, Number 6, July 2011

[21] Witold Suryn & Alain Abran, ISO/IEC SQuaRE. The second generation of standards for software product quality. November 3-5, 2003 Marina del Rey, CA, USA