

Comparative Study of Quality Models

Abdulhakim.Elmoawe Dreheeb^{1*}, Nurlida Basir² and Norasikin Fabil³

Abstract—Software is now used in almost every business application. Software quality has emerged as pivotal aspects of any software development process. Quality of software depends on the process followed during its development. Software quality is having multi-dimensional content which may be distinguished and measured easily. To be specific, with the idea of determining the multidimensional content in a more exact pattern various qualitative models have been presented by virtue of which different aspects of this matter have been attempted to be investigated properly. Quality is considered a crucial issue for education in general, and for e-learning in particular. Factors affecting the quality of software are identified from among the quality models, on the basis of the most common, crucial and critical ones. These quality characteristics could be used to reflect the quality of the software product from the view of the users. Selecting which one of the quality models to use is a real challenge; in this paper we will discuss the contents of the following quality models: include but are not limited to McCall, Boehm, FURPS, Dromey, QMOOD, ISO 9216 and etc. Which, this paper describes about software quality, affecting quality of software, different software quality models and its comparative study.

Keywords—Software Quality; Software Quality Models.

I. INTRODUCTION

SOFTWARE is critical in providing a competitive edge to many organizations, and is progressively becoming a key component of business systems, products and services. The quality of software products is now considered to be an essential element in business success. Additionally, quality is considered a crucial issue for education in general, and for e-learning in particular [1].

The poor quality may lead to financial loss, mission failure. There are a number of quality models in software engineering literature, each one of these quality models consists of a number of quality characteristics. 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 real challenge. In this paper we will discuss the contents of the following quality models: include but are not limited to McCall, Boehm, FURPS, Dromey, QMOOD, ISO 9216 and etc. and find the key differences between them. The rest of this paper is structured as follows: section 2 presents an overview of the twenty eight common quality models used in software engineering. Section 3 contains the analysis and comparison between the twenty eight quality models. Finally, section 4 concludes the paper with some comments.

II. AN OVERVIEW OF THE SOFTWARE QUALITY MODELS

This section presents several software quality models as the foundation for proposing an appropriate model for e-learning. Table 1 shows the models. A quality model is defined as ‘the set of characteristics and the relationship between them, which provide the basis for specifying quality requirements and evaluating product quality’ [7]. Among the most accepted models include but are not limited to McCall, Boehm, FURPS, Dromey, QMOOD, ISO 9216 and etc.

III. ANALYSIS OF THE QUALITY MODELS

In this section, the analysis of the quality models consists of a comparison of the availability of quality characteristics which are called factors or attributes. Thirty five characteristics within the twenty eight quality models will be presented. Table 2 presents this matrix comparison, at the in this table you will find the number of the characteristics that common in these models. Table 3 presents the characteristics definitions.

From the thirty five characteristic and twenty eight model, there are not characteristic is common to all quality models, but there a few of characteristics are common in most models, which the usability the higher one of all characteristics, where found in 27 model, while reliability in 25 model. Only one characteristic is common in 23 quality model, that is, efficiency Two characteristics is common only in 22 quality model, that is, the Maintainability and portability characteristic.. Also one characteristic is common in 18 quality model, that is, the Functionality. Moreover, one characteristic is common in 12 quality model, that is, the Reusability. Similarly, one characteristic is common in 11 quality model, that is, the Flexibility. Too, one characteristic is common in 10 quality model, that is, the Testability. furthermore , one characteristic is common in 9 quality model, that is, the Integrity .also one characteristic is common in 8 quality model, that is, the Interoperability .besides one characteristic is common to 7 quality model, that is, the Correctness . Also one characteristic is common in 6 quality model, that is, the Understandability. Likewise, one characteristic is common in 5 quality model, that is, the Expandability. Also one characteristic is common in 4 quality model, that is, the Modifiability. And 2 characteristics are common in 3 quality model, that is, the Verifiability and the Modularity. In addition, 7 characteristics is common in 2 quality model, that is , the Survivability , Supportability , Accuracy, Robustness, Scalability, Manageability ,and Learnability. Lastly, 11 characteristics that are Human Engineering, Intraoperability, Performance, Safety, security, Feasibility, Simplicity, Ambiguity, Evlovability, Compatibility, and Inheritability are defined in only one quality model.

Fakulti Sains dan Teknologi, Universiti Sains Islam Malaysia (USIM), 71800 Nilai, Negeri Sembilan Email:asid1433@yahoo.com¹, nurlida@usim.edu.my², norasikin.fabil@usim.edu.my³

TABLE I
QUALITY MODELS

No	Name of model	year	Owners	Comments	Referen ces
1	McCall's	1977	McCall	The attributes are correctness, efficiency, integrity, reliability, usability, maintainability, flexibility, testability, portability, reusability, and interoperability	[12]
2	Boehm's	1978	Boehm	The attributes are usability, human engineering, portability, testability, modifiability, understandability and efficiency.	[3,8]
3	ISO 9126	1991	ISO	The attributes are functionality, reliability, usability, efficiency, maintainability, and portability.	[2]
4	IEEE	1993	IEEE	The attributes are efficiency, functionality, maintainability, portability, reliability and usability	[2,12]
5	ISO 25010	2011	ISO	The attributes are efficiency, functionality, maintainability, portability, usability, reliability, security, and Compatibility.	[5]
6	Ghezzi	1991	Ghezzi	The attributes are accuracy, flexibility, integrity, maintainability, portability, reliability, reusability and usability	[2,12]
7	Software Assurance Technology Center (SATC)	1996	Hyatt	The attributes are efficiency, functionality, maintainability, portability, reliability, usability, supportability and feasibility.	[2,6,12]
8	AOSQUAMO	2009	Kumar	The attributes are efficiency, functionality, maintainability, portability, reliability and usability.	[8,12]
9	Perry	1987	Perry	The attributes are correctness, efficiency, flexibility, integrity, maintainability, portability, testability, usability, reliability, interoperability, and reusability.	[3]
10	Khosravi	2004	Khosravi	The attributes are namely reusability, understandability, flexibility, modularity, robustness, scalability and usability.	[2]
11	software development quality model	2008	Sharma	The attributes are efficiency, functionality, Maintainability, portability, reliability and usability.	[8,12]
12	Detusch	1988	Detusch & Willis	The attributes are correctness, efficiency, expandability, flexibility, integrity, interoperability, maintainability, manageability, portability, reliability, reusability, safety, survivability, verifiability and usability.	[11]
13	quality model for object - oriented Design (QMOOD)	2002	Bansiya	The attributes are flexibility, functionality, modifiability, reusability, understandability, usability, expandability, and modularity.	[8,12]
14	Aspect – oriented software Quality (AOSQ)	2012	kumar	The attributes are usability, maintainability, functionality, efficiency, reliability, portability and evolvability.	[8]
15	Design Enhanced Quality Evaluation (DEQUAITE)	2009	Khomph	The attributes are functionality, modifiability, robustness, scalability, learnability, reusability, understandability, testability, usability, simplicity, expandability, and modularity	[2,12]
16	UML	2010	Castillo	The attributes are usability, functionality, portability, efficiency, reliability, integrity, interoperability, accuracy, learnability, understability, testability, and modifiability	[2,12]
17	Evans	1987	Evans	The attributes are correctness, efficiency, flexibility, integrity, maintainability, portability, testability, usability, verifiability, expandability, reliability, interoperability, and reusability.	[3]
18	System and software Quality Requirements and Evaluation	2005	ISO	The attributes are efficiency, functionality, Understandability, maintainability, portability, testability, usability, reliability, and ambiguity.	[4,12]
19	Gilles	1987	Gilles	The attributes are correctness, efficiency, flexibility, integrity, maintainability, portability, testability, usability, reliability, interoperability, and reusability	[3]
20	object oriented quality model (OOQM)	2001	ISO	The attributes are efficiency, maintainability, portability, usability, reliability, and maintainability.	[8]
21	Bertoa	2002	Bertoa	The attributes are efficiency, maintainability, portability, usability, reliability, and maintainability.	[9]
22	Alvaro	2005	Alvaro	The attributes are efficiency, maintainability, portability, usability, reliability, and maintainability.	[9]
23	Rawashdeh	2006	Rawashdeh	The attributes are efficiency, maintainability, portability, usability, reliability, and maintainability.	[9]
24	Dromey's	1995	Dromey's	The attributes are functionality, reliability, efficiency, usability, maintainability, portability, and reusability.	[2,8,12]
25	FURP	1987	Grady and Packard	The attributes are functionality, usability, reliability, performance and supportability.	[2,6,8]
26	Bowen	1985	Bowen	The attributes are correctness, efficiency, usability, maintainability, portability, flexibility, reliability, reusability, integrity, interoperability, verifiability, survivability, expandability.	[13]

27	Marine	1983	Marine	The attributes are correctness, reliability, efficiency, usability, integrity, maintainability, portability, flexibility, testability, reusability, interoperability, interoperability	[13]
28	Kazman	2003	Kazman	The attributes are efficiency, security, availability, Functionality Modifiability, portability, reusability, inheritability and testability	[2,12]

IV. CONCLUSIONS

Software quality is considered as a major issue for every users of software. Through comparative study of software quality models for different characteristics. It has found that, software should have different characteristics to achieving high quality, also found many characteristics that have common in most quality models which show the importance. This comprehensive study will help the users to understand the quality factors properly, as well as, known the quality criteria in desired manner. Selecting which one of the quality models as well as the characteristic to use is a real challenge.

REFERENCES

- [1] Alkhatabi, M., Neagu, D., & Cullen, A. (2010). Information quality framework for e-learning systems. *Knowledge Management & E-Learning: An International Journal (KM&EL)*, 2(4).
- [2] Dubey, S. K., Ghosh, S., & Rana, A. (2012). Comparison of software Quality models: An analytical approach. *International Journal of Emerging Technology and Advanced Engineering*, 2(2)
- [3] Elmir, A., Elmir, B., & Bounabat, B. (2013). Towards an Assessment-oriented Model for External Information System Quality Characterization. *arXiv preprint arXiv:1310.8111*
- [4] Esaki, K. (2013). Verification of Quality Requirement Method Based on the SQuaRE System Quality Model. *American Journal of Operations Research*, 3(01), 70.
- [5] Jean-Marc Desharnais (2012) . Software Measurement Analysis of ISO/IEC 9126 and 25010.
- [6] Khaliq, M., Khan, R. A., & Khan, M. H. (2010). Software quality measurement: A Revisit. *Oriental Journal of Computer Science & Technology*, 3(1), 05-11.
- [7] Khosravi, K., & Guéhéneuc, Y. G. (2004). A quality model for design patterns. *German Industry Standard*.
- [8] Kumar, P. (2012). Aspect-oriented software quality Model: The AOSQ Model. *Advanced Computing: An International Journal (ACIJ)*, 3(2).
- [9] Miguel, J. P., Mauricio, D., & Rodríguez, G. (2014). A review of software quality models for the evaluation of software products. *arXiv preprint arXiv:1412.2977*.
- [10] R. E. Al-Qutaish, "Quality Models In Software Engineering Literature: An Analytical And Comparative Study", of *American Science*, Marsland Press, Michigan, USA, vol. 6, no. 3, (2010).
- [11] Sörumgård, S., & Sindre, G. (1995, July). Aspects of process quality. In *Proc. 4th Software Quality Conference, Dundee, Scotland* (pp. 4-5).
- [12] Suman, M. W. A. (2014). Comparative Study of Software Quality Models.
- [13] Waghmode, M. M. L., & Jamsandekar, P. P. (2013). SOFTWARE QUALITY MODELS: A COMPARATIVE STUDY.

TABLE II
COMPARISON MATRIX

No	Characteristics Models	McC all's	Bo eh m's	AO SQ UA MO	D R O m e y's	F U R P S	P E R R Y	IS O 9 1 2 6	B o w e n	Mu r i e 19 83	Ka z m a n 20 03	K h o s r a v i k	I E E	D e u t s c h	G h e z i	S A T C s	Q M O O D	A O S Q	C S D Q	D E Q U L A I	U M L	E v a n s	SQ u a R E s	Gil l i e s 19 87	O O Q M	IS O 2 50 10	B e r t o a	A l v a r o	R a w a s h d e h	Account		
1	Correctness	✓					✓		✓	✓				✓								✓		✓							7	
2	Efficiency	✓	✓	✓	✓		✓	✓	✓	✓	✓		✓	✓		✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	23	
3	Expandability								✓					✓			✓			✓			✓								5	
4	Flexibility	✓					✓		✓	✓	✓	✓		✓	✓								✓		✓						11	
5	Functionality			✓	✓	✓		✓		✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	18	
6	Human En		✓																												1	
7	Integrity	✓					✓		✓	✓				✓	✓							✓	✓	✓							9	
8	Interoperability	✓					✓		✓	✓				✓								✓	✓	✓							8	
9	Intraoperability								✓	✓																					1	
10	Maintainability	✓		✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	22	
11	Modifiability		✓														✓			✓	✓										4	
12	Performance					✓																									1	
13	Portability	✓	✓	✓	✓		✓	✓	✓	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	22	
14	Supportability						✓									✓															2	
15	Survivability								✓					✓																	2	
16	Testability	✓	✓				✓		✓											✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	10	
17	Understandability		✓										✓					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	6	
18	Usability	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	27
19	Reliability	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	25
20	Accuracy																					✓									2	
21	Verifiability							✓						✓									✓								3	
22	Safety													✓																	1	
23	Reusability	✓		✓			✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	12	
24	Manageability													✓																	✓	2
25	Feasibility															✓															1	
26	Robustness																														1	
27	Scalability																					✓									2	
28	Modularity																	✓													3	
29	Learnability																				✓	✓									2	
30	Simplicity																				✓										1	
31	Ambiguity																						✓								1	
32	Extensibility																														1	
33	Inheritability										✓																				1	
34	Security																										✓				1	
35	Compatibility																										✓				1	

TABLE III
CHARACTERISTICS DEFINITIONS

No	Characteristics	Definitions
1	Accuracy	The capability to the software product to provide the right or agreed results or effects with the needed degree of precision.
2	Ambiguity	Attributes of software related with requirements with potential multiple meanings.
3	Availability	The degree to which a work is operational and available for use as a product or to uses.
4	Correctness	The ease with which minor defects can be corrected between major releases while the application or component is in use by its users.
5	Compatibility	The degree to which a system or a component can be used and functions correctly under specified conditions of physical environments.
6	Efficiency	The capability of the software product to provide appropriate performance, relative to the amount of resources used understated conditions.
7	Expandability	The degree to which architectural, data or procedural design can be extended.
8	Evolvability	The evolution is the evaluate any software after a certain period of time to fulfill the current trends, technology, changes in user requirement and operational environment.
9	Flexibility	The effort required modifying an operational program.
10	Functionality	The capability of the software product to provide functions meet stated and implied needs when the software is under specified conditions.
11	Integrity	The extent to which access to software or data by unauthorized persons can be controlled.
12	Interoperability	The capability of the software product to interact with one or more specified systems.
13	Learnability	The capability of the software product to enable the user to learn its application.
14	Maintainability	The capability of the software product to be modified.
15	Manageability	Is concerned with how easy it is to manage changes to the process.
16	Modifiability	Corrections, improvements or adaptations of the software to changes in environment and in requirement and functional specifications.
17	Modularity	The functional independence of program components.
18	Performance	The degree to which timing characteristics are adequate.
19	Portability	The capability of the software product to be transferred from one environment to another.
20	Reliability	The capability of the software product to maintain a specified level of performance when used specified conditions.
21	Reusability	The ease with which an existing applications or component can be reused.
22	Robustness	The degree to which an executable work product continues to function properly under abnormal conditions or circumstances.
23	Safety	The capability of the software product to achieve acceptable levels of risk of harm to people business, software, property or the environment in specified context of use.
24	Scalability	The ease with which an application or component can be modified to expand its existing capacities.
25	Security	The availability of mechanisms that control of protect programs and data.
26	Simplicity	The degree to which a program can be understood without difficulty.
27	Supportability	The ability to extend the program , adaptability and serviceability , in addition to testability, computability , configurability, the ease with which a system can be installed and the ease with which problems can be localized.
28	Survivability	The degree to which essential, mission critical services continue to be provided in spite of either accidental or malicious harm.
29	Testability	The capability of the software product to enable modified software to be validated.
30	Understandability	The capability of the software product to enable the user to understand whether the software is suitable and how it can be used for particular tasks and conditions of use.
31	Usability	The capability of the software product to be understood learned, used and attractive to the user, when used specified conditions.
32	Verifiability	The ease with which an application or component can be verified to meet its associated requirements and standards.