Comparing Software Quality Models: An Overview

Abdul Rauf & Mohammad AlGhafees
Department of Computer Science
Imam Mohammad Ibn Saud Islamic University
(IMSIU), Riyadh, KSA
rauf.malik@ccis.imamu.edu.sa , maalghafees@ccis.imamu.edu.sa

***Abstract*— Software quality models are well-accepted means to support quality management of software systems. Over the last years many quality models are proposed and applied with varying aspects. But quality models are still criticized because of yet to be solved issue of little practical adoption of these models. It is defined and discussed in literature that "what are quality models?" but the key point is "how can quality models be adopted in practice?" Software quality models are used to determine the functionality, quality attributes, internal and external characteristics of the system. This paper aims to give the overview of software quality model by answering questions like what are these quality models?, how these can be implemented? and , how these software quality models can be compared? It also highlights the issues and proposed techniques, models and approaches of these models that are identified from literature survey.**

***Index Terms*— software quality, software quality models, software quality engineering.**

I. Introduction

Before exploring the vast area of software quality, I would like to start my discussion through a daily life example. In our daily routines, we usually come across a number of situations in which propositions like "I want a quality product…." As often said or heard. In daily routines, it can be observed that if we ask any person about what quality is, his/her answer varies. For example, if we take a digital watch as a product and want to perceive the ideas of quality of this watch in front of users, answers like **"**I want this to display correct time…**"**, "I want it to be a reliable product…." etc. So, it can be said that the concept of quality varies from person to person. Same is the case with software quality. The concept of quality varies from organization to organization i.e. how an organization perceives software quality. It can be measured from the perspectives of user view, manufacturer view, value based view, transcendental view [1]. However, it is usually referred to the conformance to user's expectations and needs [1, 2]. Quality is also defined as *"conformance to explicitly stated functional and performance requirements, explicitly documented development standards, and implicit characteristics that are expected of all professionally developed software."*[9] It is also defined as *"the totality of characteristics of an entity that bear on its ability to satisfy stated and implied needs"*. [10]

Quality model is defined as: "A quality model is a model with the objective to describe, assess and/or predict quality" [11]

Many tasks related to quality are not integrated in quality models. For example, specification of quality requirements and the assessment of software quality are usually not based on the same models. Another problem is that quality models miss different views on quality, which is very in software engineering. Existing quality models not define clear guideline for decomposition of quality attributes, difficult to locate elements and increase redundancy and ambiguity. The ambiguous decomposition overlaps different quality attributes.

Due to fewer resources, limited time and lack of data organizations prefer to use a limited concept of software quality. Models should evaluate high-level quality characteristics with great accuracy because high quality characteristics can assess programs and predict maintenance effort easily [12]. Considering model as tool can be used in an organization for software quality measurement at preparatory stage or for complimenting existing quality approach. Quality model should also measure the quality of changes that occur from production [14]. Section 2 provides a literature survey of software quality models. Section 3 provides comparative analysis of software quality models. Section 4 provides discussion and concluding remarks.

II. Literature Survey

Taghi M. Khoshgoftaar [15], discussed issue regarding the correctness of early detected faults in system. This work mainly focuses on "absence of fault recorded by a problem reporting system during the period of interest". The classification rules for software quality models are used to solve these types of issues.

Carina Alves [16], focused on the issue that selection process is one of the most important steps of the development of the Component Off The Shelf (COTS) based systems. Meeting the user requirements is one of the utmost important feature for COTS components. Author used the Goal-Based approach to make possible this matching between user requirements and the COTS products. Carine Alves et.al tried to focus on different aspects of this matching mechanism through a case study such as engineering of requirements, processes, components compatibility and available COTS with their properties inside market. Authors used "Mail Server System" as the case study and demonstrated the practical aspects of system by selecting different types of requirements.

IJSMS

Authors developed the requirements as can be shown in the table o1 describing the high level and operational goals. Evaluation of the goals along with the specification of goals, after which prioritization of requirements and goals has been developed. Goals are further divided into two categories, high level goals and operational ones. Constraints and objectives on the organizations are being defined by these high level goals while these high level goals are modifiable and they can be changed as change is expected. Operational goals on the other hand are providing the detailed refinement for the organizational goals; because of their nature, sometimes operational goals are also called as sub goals. To transform higher level organizational goals into operational goals, authors have suggested to use quality models. Quality models are used for specification and evaluation of the requirements based on the characteristics and relationships among them. These specific goals and quality models are used to build the COTS selection process and that is multiple process.

In Adnan Rawashde's work [17], issue addressed is that there should not be a model that ensures quality characteristics of these systems and becomes a necessity. Among the most critical processes in COTS systems they are based on the evaluation and selection of COTS components.

McCall, Boehm, FURPS are not hierarchical models and may not be able to identify the secondary attributes. There is also a lack of measures that could assess and report quality characteristics. To resolve this; new model is proposed, the aim of the new model is to guide organizations are in the process of building based on COTS to evaluate and choose the right products systems, and is essential to the success of the entire system. Considering all these aspects, it is developing a new model that can overcome the identified constraints; as system functionality. The main features of the new model include processes, products and high-end features. An additional feature that is added is "interested parties". Stakeholders are the most important part of the system. This model has the disadvantage that you can not predict the internal quality characteristics, this can be overcome by providing a valuation technique.

Florian Deissenboeck [11], authors focused on the problem regarding quality models itself, quality models are always criticized, because its application in practice has several problems. This review is caused by a clear definition of what quality models are and what they are; there is a lack of explicitly stated requirements quality models in their mode of intended application. To resolve this problem, general set of requirements for the quality of models and meta-models are derived. The proposed model is the DAP (definition, evaluation, prediction) model is the most advanced form as a meta model and has been shown in Figure 01 below. You can define how quality models can be used in the development and evolution. It is a framework to define, predict, evaluate and improve quality. This meta-model can be used in two contexts; assessment of existing models in the specific context and the development and improvement of the quality attributes.

Table 1: Goals representation of Mail Server [16]

High level goals	Operational goals
g1 Ensure the delivery of messages	g1.1 Set the possible number of entries g1.2 The time span between each try should be determined g1.3 A notification has to be sent back with successful reception of the each message
g2 Ensure that messages should be retained in every situation	g2.1 If mailbox is out of space, message shouldn't be deleted. g2.2 In case any other failure happens, message should be retained and should have some backup. g2.3 Messages must never get lost if they cannot be delivered
g3 Make sure the speedy delivery of messages	g3.1 Maximum response time for a message to be delivered should have a limit g3.2 Message throughput should be less than 5 minutes per Mb g3.3 Message should have average delivery rate of 1 minute
g4 Work in cooperation should be encourages	g4.1 Integrated management should be provided. g4.2 Provide instant messaging g4.3 Voice and video communication should be provided
g5 Data should have a minimum level of security	g5.1 Only authorized users should be given access g5.2 Data must be integrated
g6 External attacks should be prevented	g6.1 Spam should be stopped and filters should be formed. g6.2 Virus scanning should be regular and on periodical basis.

IJSMS

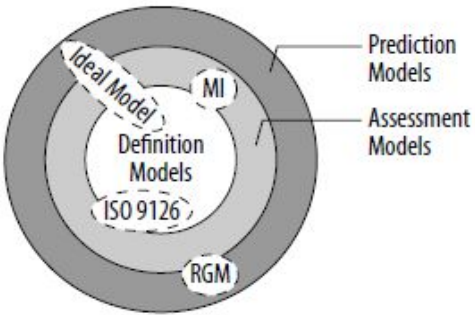


Fig 1: DAP Classification for Software Quality Models

Oleg V. Komogortsev [18], the authors questions utilization ratio. The research question raised in the case study was "to allow members of four different groups of actors agree largely on the priorities assigned to the different attributes of software quality?", He wondered "if the priorities of quality software differ depending on the type of software in the study? " to solve this problem is proposed and presented to software reuse based on effort model. Different strategies apply usability. But they can record the data manually for cheap usability evaluations. This research will allow us to show GUI (Graphical User Interface) design flaws. This is easier and achievable.

Hongyu Zhang [20], question addressed by the authors that there is a lack of an explicit model that can represent the impact of variations in quality attributes. To resolve this problem, the (Bayesian belief network) BBN proposed approach. A BBN represents domain knowledge and expert experience in developing similar projects. Metric, paper treated individually and establishing research hypothesis. However, further work is needed to determine if it is possible to reduce a metric or approximate coefficient of use.

Maria Haigh [19], question addressed in this paper is high quality "may seem an obvious requirement for any piece of software, but different groups of actors involved in the production and use of conceptualizing the requirement in the same way? To overcome this problem, survey is designed to explore interested to follow parts. two null hypotheses were also expressed. a pilot study is a measuring instrument used in this research. in exploring this survey the differences between the different economic actors are identified. this is achieved through a case study carried out with current MBA students and recent graduates. population size is 300 students. experimental design and research hypotheses also used in this model. Two null hypotheses of the study they can be expressed as follows:

H0A: Different stakeholders gets agree and there is no difference in their opinion with respect to the quality.

H0B: Different software types have same quality requirements and quality requirements remain same despite the change in nature of the software.

We have set the alternative hypothesis as well adopted from the above mentioned references.

H1A: Different stakeholders have different priorities with respect to the software quality.

H1B: Software quality varies from one software to another based on the type of the software. .

We want to discuss these hypothesis by asking the different respondents about their opinion by sending these hypothesis as attachment.

III. Discussion & Conclusion

Quality models software once were so impressive, but still have some common problems that exist. The Table 02 below list of quality models that were considered for comparison. The most common problem in almost all quality models is the "adoption in practice." Many models have been developed for quality improvement and measurement of quality, but it is unclear how it can easily pass or the circumstances in which they can use. One of the major shortcomings of current quality models do not conform to an explicit meta-model.

The responses we received as a result of this survey indicate that most of the respondents have agreed to our hypothesis H1A and H1B which was expected.

Table 03 lists some of the tools and techniques that are based models n. Therefore, the semantics of the elements of the model is not precisely defined and interpretation is left to the reader. Another common problem in quality models today do not take into account the different views on quality. The views are lacking in most current models. Table 04 shows a comparative analysis of the models shown in Table 02. In future work, we must develop a meta-model that can overcome all these problems are on the functionality, ease of use and adoption of models quality in practice. A new model should be able to solve these problems, and have additional features to determine the internal quality of the characteristic attributes that are most important. Different quality models described in the literature as McCall, Boehm, FURPS, Dromey of the ISO 9126 standard and so on, but all these models are theoretical concepts that have well-defined quality metrics by which we measure the quality product or software process. Several models are designed for different quality and therefore, not all points intentions are applicable to all models. Each quality model includes at least one of the following problems. They are: in general, the definition of models, pricing models, predictive models. These models focus on the factor and the environmental quality of the attributes that the environment in which the model appropriate quality and not to mention how to adopt this model. In the future, it is necessary to develop a well-defined meta-model that reduce the quality attributes of the problems and provide the framework and appropriate for the adoption of this model measures.

IJSMS

Sr no	Software quality models
1	McCall's Quality Model.
2	Boehm's Quality Model.
3	Dromey's Quality Model.
4	FURPS Quality Model.
5	ISO 9126 Quality Model.

Table 02: Software Quality Models

Sr #	List of approaches, tools and techniques
1	Classification rules for software Quality models.
2	Combined (ISO/IEC9126, DROMEY, ABAS).
3	Fuzzy-set decision trees.
4	White-box model (decision trees, rule classifiers).
5	Quality Model tool.
6	General model for COTS-Based systems.
7	Goal-based approach.
8	DAP (Definition, Assessment and Prediction)
9	Effort-based model.
10	BBN (Bayesian Belief Network).
11	Pilot study on the basis of survey design

Table 03: List of Approaches, Tools and Techniques

DISCLOSURE POLICY

"The authors declare that there is no conflict of interests regarding the publication of this article."

REFERENCES

[1] Barbra Kitchenham, Shai Lawrence Pfleeger, "Software Quality: The Elusive Target", IEEE Software, vol. 13, pp. 12 – 21, January 1996

[2] Patrik Berander, Lars-Ola Damm, Jeanette Eriksson, Tony Gorschek, Kennet Henningsson, Per Jönsson, Simon Kågström, Drazen Milicic, Frans Mårtensson, Kari Rönkkö, Piotr Tomaszewski, Software quality attributes and trade-offs, chapter 1, June 2005.

[3] Maryoly Ortega, Maria Perez, Teresita Rojas, "Construction of a Systemic Quality Model for Evaluating a Software Product", Software Quality Journal, vol.11,pp. 219–242, 2003.

[4] Adnan Rawashdeh, Bassem Matalkah, "A New Software Quality Model for Evaluating COTS Components", Journal of Computer Science, vol. 2, pp. 373-381, 2006.

[5] Luis Fernández Sanz, "Software quality: concepts and evidences"

[6] Rafa E. Al-Qutaish, "Quality Models in Software Engineering Literature: An Analytical and Comparative Study", Journal of American Science, vol. 6, pp. 166-175, 2010

[7] Xavier Franch, Juan Pablo Carvallo, "Using Quality Models in Software Package Selection", IEEE Software, vol. 20, pp. 34-41, January 2003.

[8] Taz Daughtrey, Fundamental Concepts for the Software Quality Engineer, chapter 1.2, pp.17- 24, 2007

[9] Roger S. Pressman, Software Engineering: A Practitioner's Approach, 6th Edition, chap.8, pp. 462-497, 2005

[10] Behshid Behkamal, Mohsin Kahani, Mohammad Kazem Akbari, "Customizing ISO 9126 quality model for evaluation of B2B applications", Information and Software Technology, vol.51, pp. 599-609, March 2009

[11] F. Deissenboeck, E. Juergens, K. Lochmann, and S. Wagner, "Software quality models: Purposes, usage scenarios and requirements", WoSQ'09, IEEE CS Press, 2009.

[12] Khashayar Khosravi, Yann-Ga¨el Gu´eh´eneuc, "On Issues with Software Quality Models", chapter 11, pp.218-235, ICFAI University Press, January 2008.

[13] Juan P. Carvallo, Xavier Franch, Gemma Grau, Carme Quer, "A Tool for Building Software Quality Models", Proceedings of the 12th IEEE Requirements Engineering International Conference pp. 358-359, 2004.

[14] Gerard Horgan, Souheil Khaddaj , "Use of an adaptable quality model approach in a production support environment", Journal of Systems and Software, vol. 82, pp.730-738, 2009

[15] Taghi M. Khoshgoftaar, Member, IEEE, and Edward B. Allen,. 49, NO.. "A Practical Classification-Rule for Software-Quality Models", IEEE TRANSACTIONS ON RELIALIBITY, 2, JUNE 2000.

[16] Carina Alves1, Xavier Franch2, Juan P. Carvallo2, and Anthony Finkelstein1, "Using Goals and Quality Models to Support the Matching Analysis During COTS Selection", Department of Computer Science, University College London, London, UK, X. Franch and D. Port (Eds.): ICCBSS 2005, LNCS 3412, pp. 146 – 156, 2005.

[17] Adnan Rawashdeh and Bassem Matalkah, "A New Software Quality Model for Evaluating COTS Components", IT Department, Faculty of Science, Mutah University, P.O. Box 7, Mutah 61710, Karak, Jordan, Journal of Computer Science 2 (4): 373-381, 2006.

[18] Oleg V. Komogortsev , Carl J. Mueller , Dan Tamir , Liam Feldman, "An Effort Based Model of Software Usability", Texas State University San Marcos, TX 78666, Proceedings of the International Conference on Software Engineering Theory and Practice (SETP-08), July 2009.

[19] Maria Haigh, "Software quality, non-functional software requirements and IT-business alignment", School of Information Studies, University of Wisconsin-Milwaukee, 3210 N. Maryland Ave, Bolton Hall, Rm. 568, Milwaukee, WI 53211, USA, Published online: 1 May 2010.

[20] Hongyu Zhang, Stan Jarzabek, and Bo Yang, "Quality Prediction and Assessment for Product Lines", Department of Computer Science School of Computing National University of Singapore 3 Science Drive 2, Singapore 117543, J. Eder and M. Missikoff (Eds.): CAiSE 2003, LNCS 2681, pp. 681–695,2003.

IJSMS

Table 04: Comparative Analysis of Quality Models

Sr. no	Attributes/characteristics/factors	McCall's	Boehm's	Dromey's	FRUPS	ISO 9126
1	Maintainability	YES	NO	NO	NO	YES
2	Flexibility	YES	NO	NO	NO	NO
3	Testability	YES	YES	NO	NO	NO
4	Correctness	YES	NO	YES	NO	NO
5	Efficiency	YES	YES	NO	NO	YES
6	Reliability	YES	YES	NO	YES	YES
7	Integrity	YES	NO	NO	NO	NO
8	Usability	YES	NO	NO	YES	YES
9	Portability	YES	YES	NO	NO	YES
10	Reusability	YES	NO	NO	NO	NO
11	Interoperability	YES	NO	NO	NO	NO
12	Human Engineering	NO	YES	NO	NO	NO
13	Understandability	NO	YES	NO	NO	NO
14	Modifiability	NO	YES	NO	NO	NO
15	Functionality	NO	NO	NO	YES	YES
16	Performance	NO	NO	NO	YES	NO
17	Supportability	NO	NO	NO	YES	NO

IJSMS