IMPACT: International Journal of Research in Engineering & Technology
ISSN (P): 2347-4599; ISSN (E): 2321-8843
Vol. 6, Issue 8, Aug 2018, 1-18

A COMPARATIVE STUDIES OF SOFTWARE QUALITY MODEL FOR THE SOFTWARE PRODUCT EVALUATION

Durgesh Kumar Singh[1] & Ajay Kumar Bharti[2]

[1]Research Scholar, Department of Computer Science & Engineering, Maharishi University, Main Campus, Lucknow, Uttar Pradesh, India

[2]Professor, Department of Computer Science & Engineering, Maharishi University, Main Campus, Lucknow, Uttar Pradesh, India

Received: 19 Jul 2018 **Accepted: 31 Jul 2018** **Published: 14 Aug 2018**

ABSTRACT

Actually, software products are increasing in a fast way and are used in almost all activities of human life. Consequently measuring and evaluating the quality of a software product has become a critical task for many companies. Several models have been proposed to help diverse types of users with quality issues. The development of techniques for building software has influenced the creation of models to assess the quality. Since 2000 the construction of software started to depend on generated or manufactured components and gave rise to new challenges in assessing quality. These components introduce new concepts such as configurability, reusability, availability, better quality and lower cost. Consequently, the models are classified into basic models which were developed until 2000, and those based on components called tailored quality models. The purpose of this article is to describe the main models with their strengths and point out some deficiencies. In this work, we conclude that in the present age, aspects of communications play an important factor in the quality of the software.

KEYWORDS: *Success Measures, Web Usability, Web Application Quality Model, Software Quality Model, Web Metrics, Quality Evaluation Framework, Attribute Weighting, Web Attribute*

INTRODUCTION

Many numbers of new websites have been launched every day. Ones with similar content will not have the same degree of quality. If the quality is poor, the user will simply leave the website and go elsewhere. Generally, there is no second chance to get a user back to the website. Therefore, in order to improve the quality of a website. The quality of a website makes a website profitable, user-friendly and accessible, and it also offers useful and reliable information, providing good design and visual appearance to meet the users' needs and expectations. This can be done by defining the measurable website criteria. Website quality is dependent on the quality of the software. Website Quality (or Quality of Websites) could be measured from two perspectives: Programmers, and End-users. The aspects of website quality from programmers focus on the degree of Maintainability, Security, Functionality, etc. Whilst the end-users are paying more attention to Usability, Efficiency, Creditability, etc. A website quality model shows an approach to the definition and measurement of website quality. It describes the trade-off between the user's needs to be well-established and flexible functions to permit the web application with diverse content.

METHODS

QCF provides the quality measurement in a simple quality compliance scale. The scale starts at 0% and ends at 100%, where 0% indicates poor quality compliance and 100% indicates excellent quality compliance. This is the QCF score of the web application. QCF works using bottom-up approach.

A final score is the quality measurement. The following formulas show how the quality measurement is calculated for different components of QCF:

Quality Measurement

Quality Measurement = Σ Children's QCF / No. of children

Characteristics and Sub-Characteristics QCF Score

Quality Characteristic Score = Σ Children's QCF / No. of children

Attribute QCF score

Quality indicator = (Earned Score/ Possible Score) ×100%

Here "Children" refers to the quality characteristics, quality sub-characteristics, or quality indicators in the hierarchy. It is worth remembering that some features of the website depending on the specific purpose, and perspective on the purpose of the page. Therefore, all the resulting values must be weighted.

Search Strategies

Quality models have been found using the search engine Google Scholar, databases Science Direct, Ebsco, Trove (repository of information of the National Library of Australia) and NDTLD (Networked Digital Library of Theses and dissertations).

The main keywords used were "quality of software", "models for quality of software", "Evaluation of the quality of software", "metrics for evaluation of software", "general quality software product models", "models for COTS components", "Models for free/open source quality", "Tailored quality models". The articles were classified according to the division established: Basic Quality, Tailored Models, and Open Source Models.

Inclusion and Exclusion Criteria

The articles were classified according to their relevance preferring those describing models. In the state of the art articles, we found several synonymous terms. Table 1 was constructed, using the literature review, to clarify the terminology and concepts related to quality. Regarding the exclusion criteria, the articles oriented to the evaluation of the software building process were set aside, since the purpose of the article is aimed at quality aspects of finished software products. The terminology mainly uses the international standards stated by the American Society for Quality [29] and in the ISO [5,11,12,13,14].

Basic Quality Models

According to their importance and following the timeline of figure 1, the main Basic models are described in this section. They are characterized because they make global assessments of a software product.

Mc Call Model

The Mc Call model established product quality through several features. These were grouped into three perspectives: Product Review (maintenance, flexibility, and testing), Product Operation (correct, reliable, efficient, integrity and usability) and Product Transition (portability, reusability, and interoperability). Figure 2 shows the model.

The major contribution of the McCall method was to considered the relationships between quality characteristics and metrics. This model was used as a base for the creation of others quality models [25].

The main drawback of the Call Mac model is the accuracy in the measurement of quality, as it is based on responses of Yes or No. Furthermore, the model does not consider the functionality so that the user's vision is diminished.

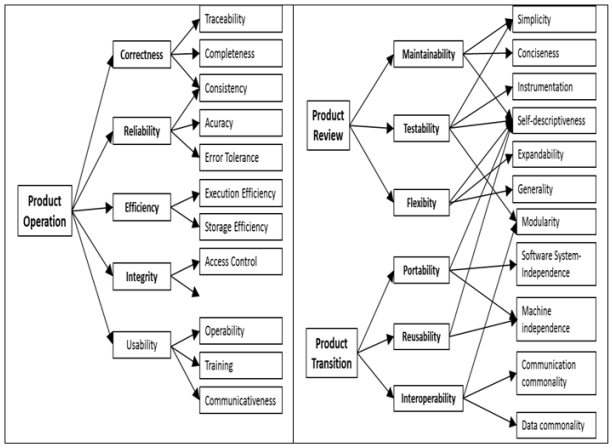

Figure 2: Mc Call Quality Model – 1977

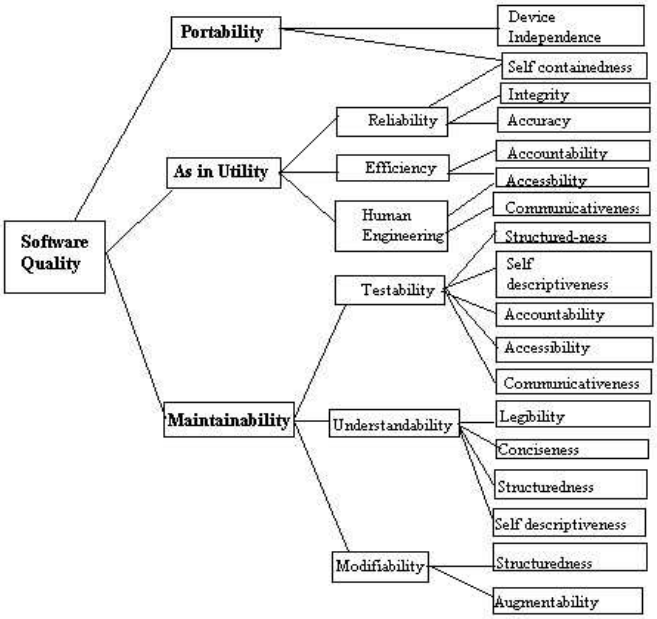

Figure 3

Boehm Model

Boehm [8] establishes large-scale characteristics that constitute an improvement over the Mc Call model because add factors at different levels. The high-level factors are: a) Utility indicating the easiness, reliability, and efficiency of use

of a software product; b) maintainability that describes the facilities to modify, the testability and the aspects of understanding; c) portability in the sense of being able to continue being used with a change of environment. Figure 3 [25] shows the model.

Dromey Model

The Dromey model [10] is based on the perspective of product quality. In this way, the quality evaluation for each product is different and a more dynamic evaluation is established. The model states that for a good quality product, all the elements that constitute it should be so. However, there is no discussion of how this can be done in practice, and this theoretical model is used to design others more specific models. Figure 4 shows the model.

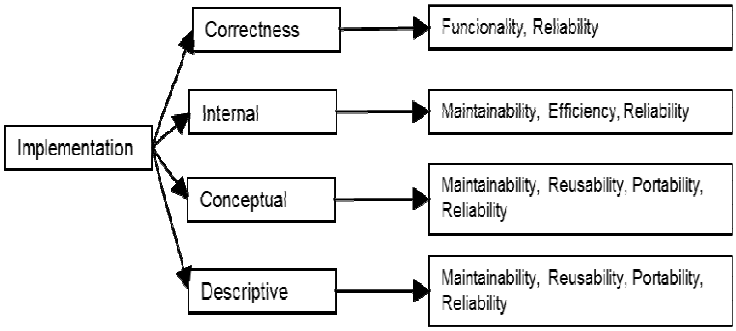

Figure 4: Dromey Model

FURPS Model

The model categorizes the characteristics as Functional Requirements (RF) and non-functional (NF). The RF is defined by the inputs and outputs expected or Functionality(F) while the NF are grouped as Usability (U), Reliability (R), Performance (P) and product support (S) [9]. Figure 5 shows these characteristics. Its main problem is that some main features, like portability, are not considered.

ISO 9126 Model

The ISO 9126 model [5] was based on the McCall and Boehm models. The model has two main parts consisting of: 1) the attributes of internal and external quality and 2) the quality in use attributes.

Internal quality attributes are referred to as the system properties that can be evaluated without executing, while external refers to the system properties that can be assessed by observing during its execution. These properties are experienced by users when the system is in operation and also during maintenance.

The quality in use aspects are referred to the effectiveness of the product, productivity, security offered to the applications and satisfaction of users. Figure 6 [11,12] shows a view of the relationship between internal, external and quality in use attributes. Figure 7 and 8 illustrates the model [5,11,13].

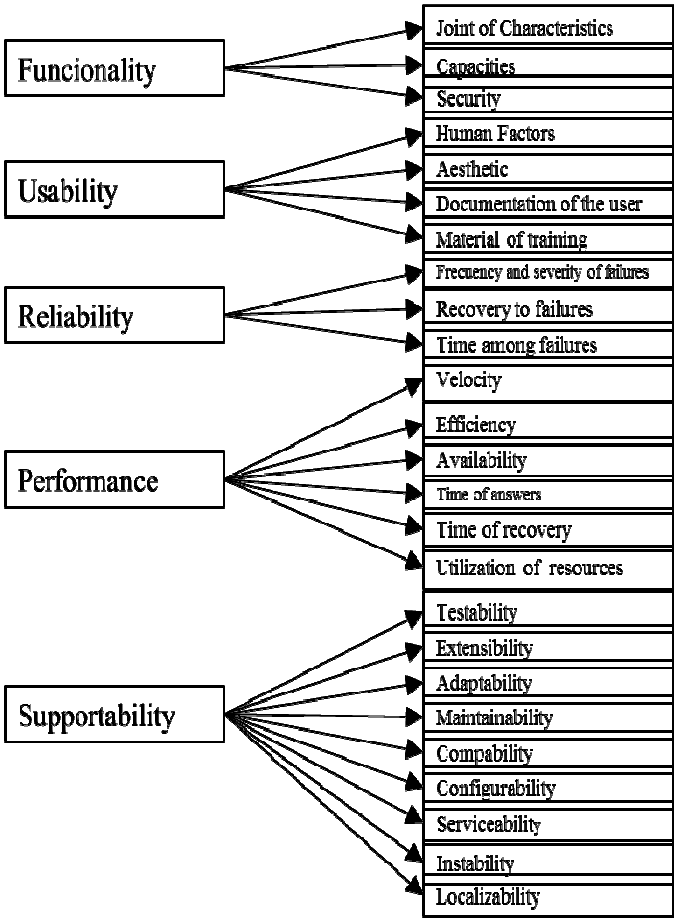

Figure 5: FURPS Model

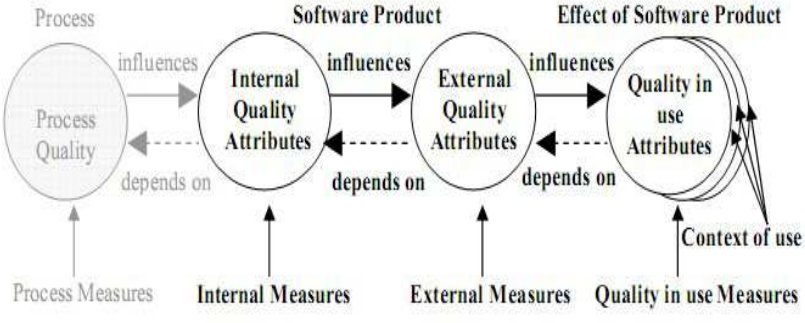

Figure 6: Quality in the Lifecycle ISO 9126

The ISO-9126 model has been used as the basis for Tailored Quality Models. One of its features was to standardize the terminology regarding the quality of software.

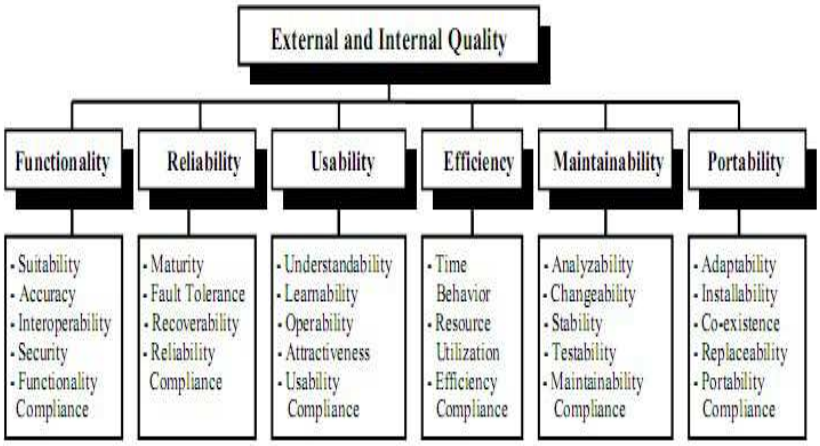

Figure 7: ISO 9126 Quality Model for External and Internal Quality

Figure 8: ISO 9126 Quality in Use

ISO 25010Model

This standard emerged in 2007 updating the ISO 9126 model. It is subdivided into 8 subkey features and characteristics. Constitute a set of standards based on ISO 9126 and one of its main objectives is to guide in the development of software products with the specification and evaluation of quality requirements. Figure 9 illustrates the model

This model considers as new characteristics the security and compatibility that groups some of the former characteristics of portability and those that were not logically part of the transfer from one environment to another. It uses the term transferability as an extension of portability.

As with the ISO / IEC 9126, this standard maintains the three different views in the study of the quality of a product, as they were illustrated in Figure 6 [14].

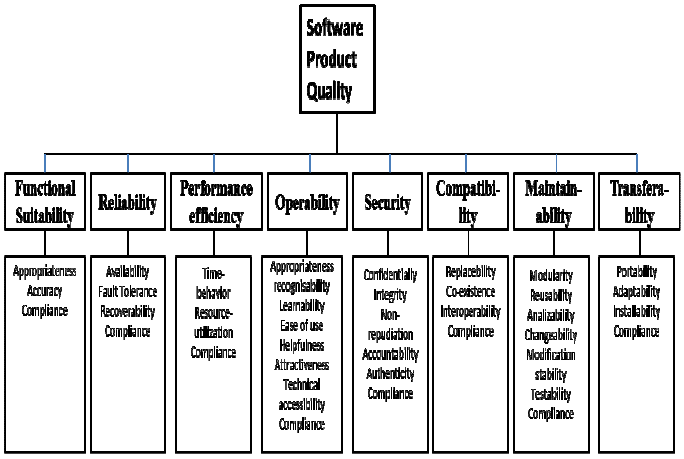

Figure 9: ISO 25010 Models (ISO/ IEC CD 25010 2007)

Tailored Quality Models

From 2001 the development of software was based on components (CBSD). The Non-Basic models Software development concentrated on the use of Commercial Off-The-Shelf Components (COTS). Figure 10 illustrates the activities of the development of a product based on COTS available in the market

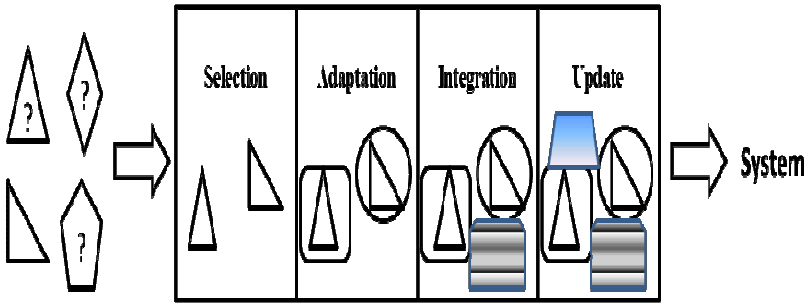

Figure 10: Activities for the Construction of a System Using Components

Bertoa Model

The Quality Model Bertoa [15] is based on the ISO 9126 Model [5]. It defines a set of quality attributes for the effective evaluation of COTS. The COTS are used by software development companies to build more complex software. The model discriminates those features that make sense for individual components and is shown in figure 11.

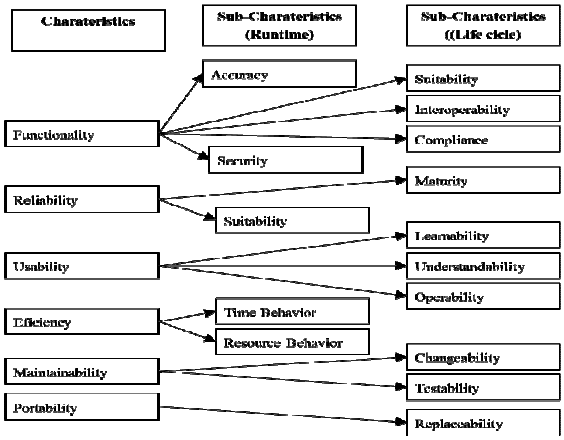

Figure 11: Bertoa Model

GEQUAMO

This model called GEQUAMO (Generic, Multilayered and Customizable Model), was created by E. Georgiadou [16] and consists of the gradual breakdown into sublayers of features and characteristics and is intended to encapsulate the various user requirements in a dynamic and flexible way. In this form, the user (end user, developer, and manager) can build their own model reflecting the emphasis (weight) for each attribute and/or requirement. Figure 12 shows the decomposition of a CASE tool [16].

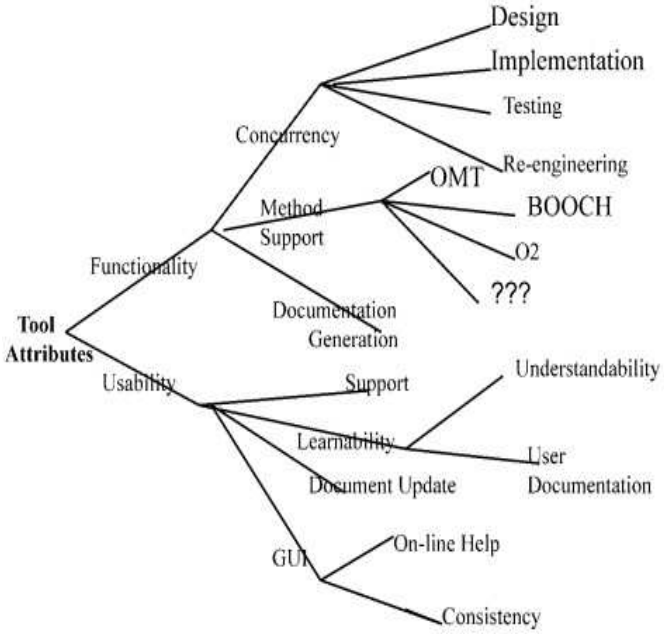

Figure 12: Layer of Characteristics Applied to a Tool CASE

Alvaro Model

Alvaro method considers a framework for the certification of software components) in order to establish the elements of quality components [17,30]. This framework considers four modules:

Model quality components for the purpose of determining the characteristics to be considered, 2) Framework for technical certification, which determines the techniques that will be used to evaluate the features provided by the model 3) the certification process that defines a set of techniques that evaluates and certifies the software components with the aim of establishing a well-defined component certification standard and 4) the frame containing the metric, which is responsible for defining a set of metrics evaluating the properties of the components in a controlled manner. In this article, we refer to the quality components model.

Figure 13 describes the model where the introduced sub-features are highlighted in bold.

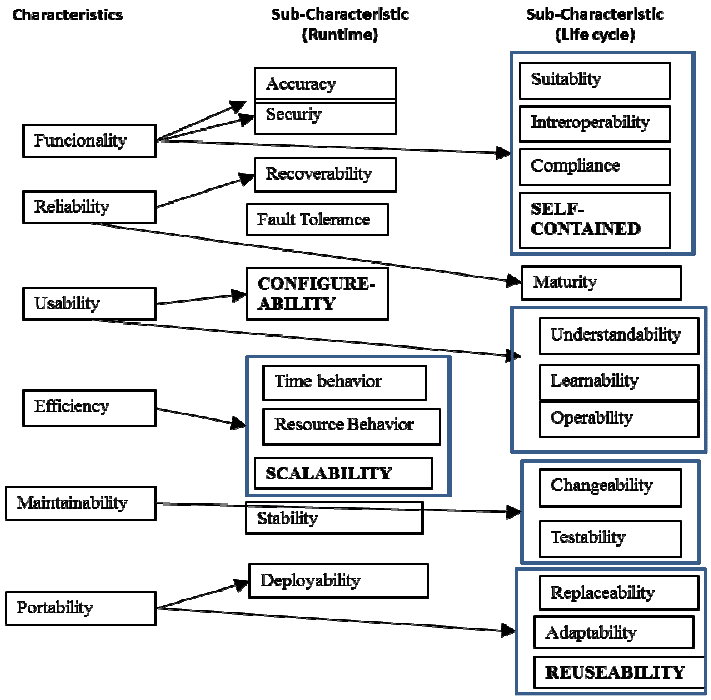

Figure 13: Alvaro Model

Rawashdeh Model

The Rawashdeh Model [18] has as main objective the needs of different types of users. The model focuses on using components COTS and has been influenced by the ISO 9126 and Dromey models. The model sets out four steps to create a product quality model [18] that are:

Identify a small group of high-level quality attributes, then using a top-down technique each attribute is decomposed into a set of subordinate attributes.

Distinguish between internal and external metrics. Internal measure internal attributes such as specifications or source code, and external system behavior during testing operations and components. Identification of users for each quality attributes. Built the new model is with ideas of ISO 9126, and Dromey Model Figure 14 shows the model.

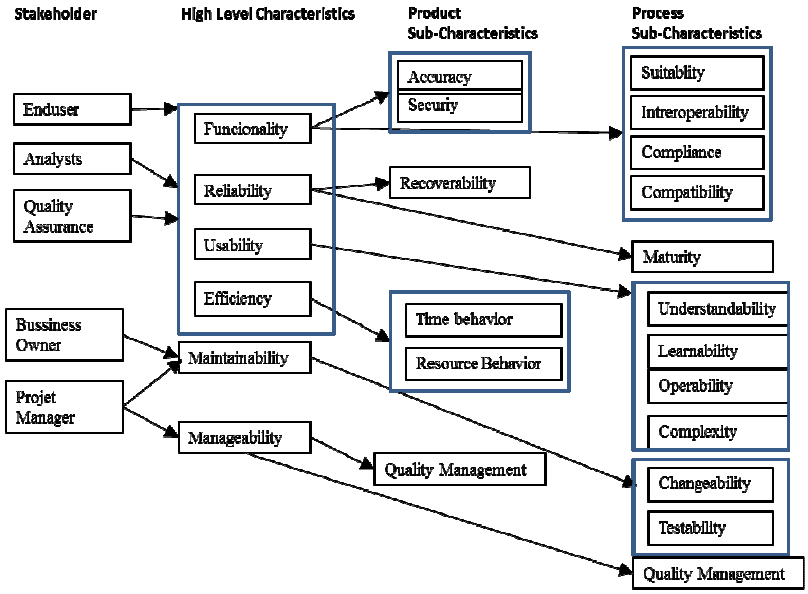

Figure 14: Rawashdeh Model

Open Source Models

Actually, free Software products have much popularity for the diverse characteristics and freedoms they offer and because they are used in different contexts. Many of them are directed to perform the same or similar applications than traditional products. For example, they can be Free Software Operating Systems (such as Linux, Solaris, FreeBSD), middleware technologies/databases (Apache Web Server, MySQL) and products for the end user (Mozilla Firefox, Open Office).

Models for assessing the quality of Free Software products adapt models like ISO-9126, adding some particular aspects of Free Software. It is noteworthy that although there is a distinction between models of first and second generation, an ideal model that captures all aspects of quality in a free software product has not been defined yet [31].

According to [32,33] these models started in 2003 and all of them emphasizes about the open source. In the next section, we describe four models.

Cap Gemini Open Source Maturity Model

The model is based on the maturity of the product and is set according to maturity indicators. These indicators are grouped into product and application indicators [34]. For the final evaluation, each of the sub-indicators is given a value between 1 and 5 giving a total score. Figure 15 shows the model.

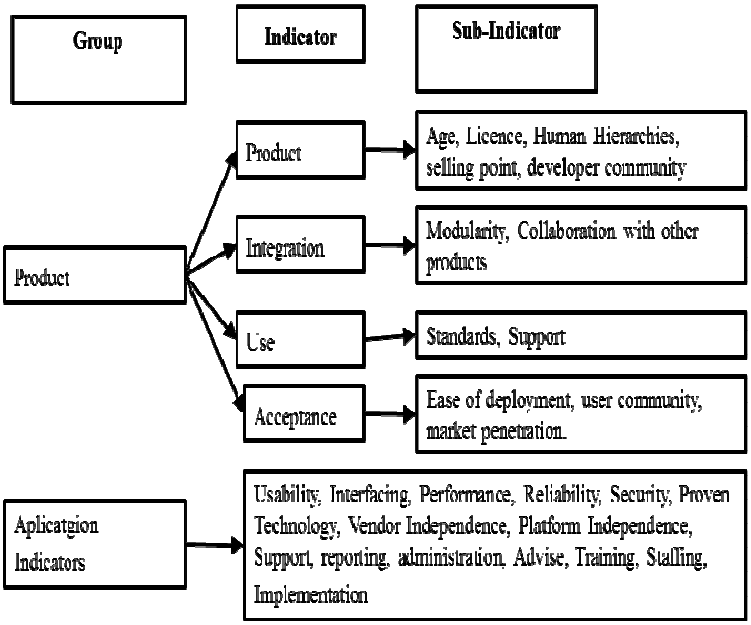

Figure 15

Open BRR Model

The model is called the Business Readiness Rating framework and was influenced by the Cap Gemini and ISO 9126 Models. In this context identifies categories that are important for evaluating open software. The model has seven categories and thereby accelerates the evaluation process, ensuring better choices with a small set [32]. The seven categories can be refined for greater granularity and cover aspects that have not been considered at the highest level.The objective is to keep always at a very simple level [35]. Figure 16 shows the model.

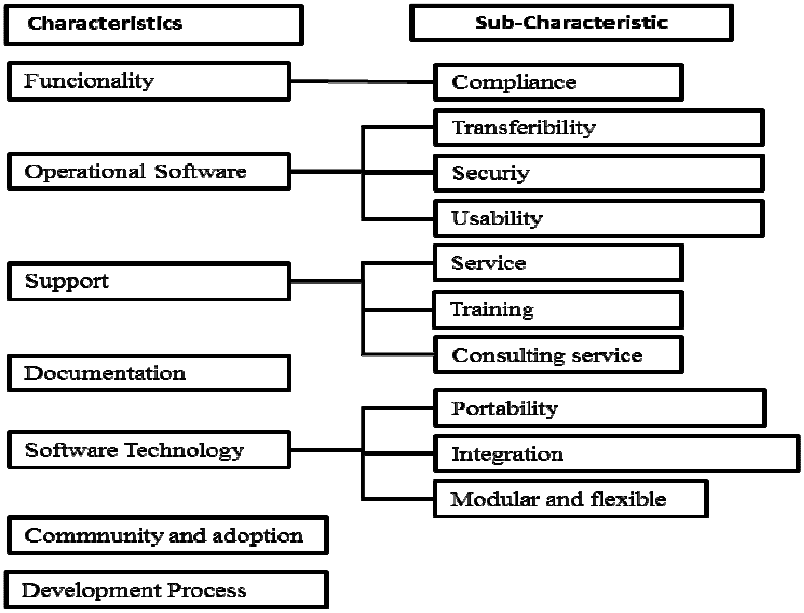

Figure 16: Open BRR Model

SQO-OSS Model

This is a hierarchical model that evaluates the source code and the community process allowing automatic calculation of metrics [32]. The model is shown in figure 17 and according to [36], the model differs from others in the following aspects:

Focus on the automation in contrast of other models that require heavy user interference.

Is the core of a continuous quality monitoring system and allows automatic metrics collection.

It does not evaluate functionality.

It focuses on the source code. The source code is the most important part of a software project.

Considers only the community factors that can be automatically measured.

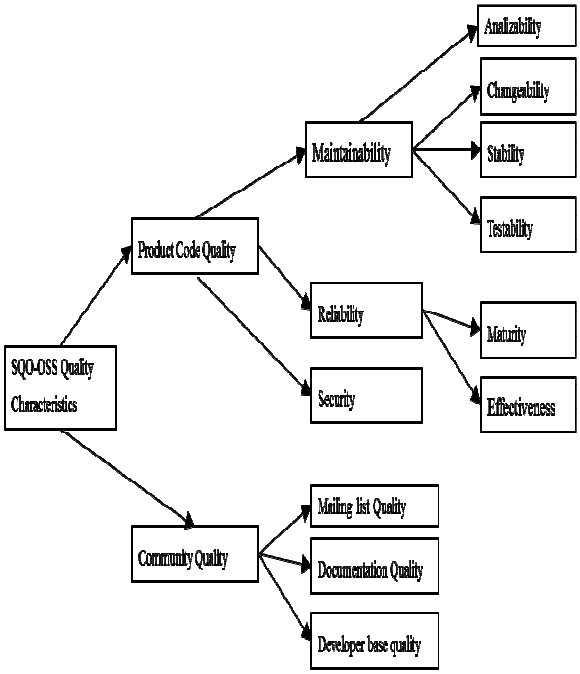

Figure 17

QualOSS Model

It is a model that emphasizes three aspects: 1) Product characteristics, community characteristics and 3) Software process characteristics are equally important for the quality of a Free/ Open source product [33]. The model is shown in figure 18 [31].

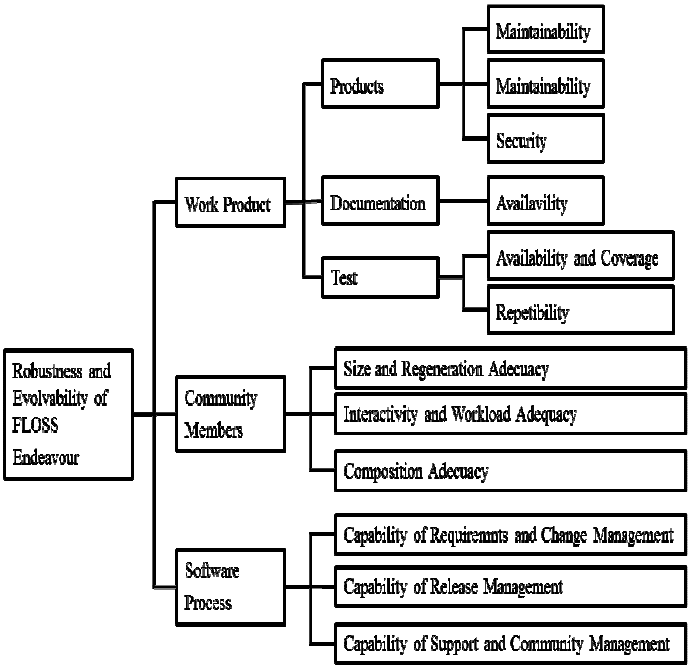

Figure 18: QualOSS Model

The QualOSS model states that quality is highly depending on the context in which it is used an the purposes that a company or person pursues with it.

This model corresponds to the second generation of Free/Open source models and where most of the assessment is highly automated.

Model Comparison

Al-Baradeen [24, 37], Al-Qutaish [25], Samarthyam [21] and Ghayathri [27] conducted comparative studies of Basics Quality Models, reaching different conclusions depending on the as they consider more important.

Table 2 shows a comparison of the basic models regarding the main characteristics according to Table 1. We include the ISO 25010 in this evaluation because it contains the last standardized terminology.

From table 2 we conclude that Model ISO 25010 is the most complete among the Basic Models because it covers 26 of the 28 features. Flexibility is related to the manufacturing process [27] and is considered as an aspect of maintainability. Regarding Human Engineering this is a particular feature considered only in the Boehm model and has the close relation with operability, but this last concept is wider. From the table, we conclude that reliability is a common feature of all models. The reason is the close relation with the opinion of users and the success of any product will depend on the fact of being used or not.

Table 2 was constructed using the sub-characteristics of the model. However and because these features are included in larger characteristic, it is possible that the presence of a feature implies that other has to be present. For example, the transferability is related with some aspects of portability and adaptability.

Table 2: Comparison of Basic Models

Characteristic	McCall	Boehm	FURPS	Dro-mey	ISO-9126	ISO-25010
Accuracy					X	X
Adaptability			X			X
Analyzability					X	X
Attractiveness					X	X
Changeability					X	X
Correctness	X					X
Efficiency	X	X		X	X	X
Flexibility	X					
Functionality			X	X	X	X
Human Engineering		X				
Installability					X	X
Integrity	X					X
Interoperability	X					X
Maintainability	X			X	X	X
Maturity					X	X
Modifiability						X
Operability					X	X
Performance			X		X	X
Portability	X	X		X	X	X
Reliability	X	X	X	X	X	X
Resource utilization					X	X
Reusability	X			X		X
Stability					X	X
Suitability					X	X
Supportability			X		X	X
Testability	X	X			X	X
Transferability						X
Understandability		X			X	X
Usability	X		X	X	X	X

Comparison among tailored oriented models is more difficult because they use the model in a particular context. The models can be either product oriented (GECUAMO), or for particular domains (Bertoa) or adapted from the point of view of a user (Rawashdeh). Table 3 has been made with almost the same features as the basic models. However, it must be noted that the absence of a feature does not invalidate any model.

Table 3: Comparison of Tailored Quality Models

Characteristic	Bertoa	Gecuamo	Alvaro	Rawashdeh
Accuracy	X		X	X
Adaptability		X	X	
Analyzability				
Attractiveness				

Changeability	X		X	X
Compliance	X	X	X	X
Configurability			X	
Compatibility				X
Correctness		X		
Efficiency			X	X
Fault Tolerance			X	
Flexibility				
Functionality	X	X	X	X
Human Engineering				
Installability				
Integrity				
Interoperability	X		X	X
Learnability	X	X	X	X
Maintainability	X		X	X
Manageability				X
Maturity	X	X	X	X
Modifiability				
Operability	X			X
Performance				
Portability	X		X	
Recoverability	X			X
Reliability	X		X	X
Replaceability	X		X	
Resource utilization	X	X	X	X
Reusability	X		X	
Scalability			X	
Stability			X	
Security	X		X	X
Self Contained			X	
Suitability	X		X	X
Supportability				
Testability	X	X	X	X
Time Behavior	X		X	X
Understandability	X	X	X	X
Usability	X	X	X	X

CONCLUSIONS

The overall conclusion is that there are very general models for assessing software quality and hence they are difficult to apply to specific cases. Also there exist tailored quality models whose range is in small domain, using as starting model the ISO 9126. Models for Free/Open source emphasize the participation of community members. Tailored Quality Models originated from the Basic Models basic consider a specific domain and selects the features and sub features to consider. The model created in this way is for a specific, particular product or from the point of view of a user domain. Therefore have limitations. The ISO 9126 model was updated in 2007 by the ISO 25010 that redefines the fundamental characteristics increasing them from six to eight. In the future, the developing of models will have to consider these characteristics. Future works will have as a main reference this model. In the case of Free Software, the aspects of user communities should be considered as a feature of high level because of the level of influence in both the construction and the product acceptance. In all the models studied none has incorporated the aspect of communication as one of the quality factors. At the present time, there is a need for quality components for communications at all levels and especially in complex systems, where it becomes a critical factor because of the Internet. Finally, we note that in most of the studied models the factors and criteria have the same value which is relative because it depends on the application domain. For example aspects of transferability can be crucial in software that is installed on different machines.

The importance of the web evaluation framework has been proposed by three-level structures, which are quality characteristics, quality sub-characteristics and measurable criteria (indicators). In the first level, the web evaluation

framework proposed five quality characteristics which included Aesthetics, Ease of Use, Multimedia, Rich Content and Reputation. Aesthetics and Reputation are the main parts of this paper. The second level characteristic is broken down by several Sub-characteristics. Each Sub-characteristic is inherited from parental quality characteristics, however only Aesthetics and Ease of Use have Sub-characteristics, and others such as Multimedia, Rich Content and Reputation are directly divided into the third level – measurable indicators. Last, the website quality metrics calculate the quality criteria through several evaluation formulae giving results with the meaningful quality scores.

REFERENCES

1. *A. Alvaro, Almeida and S. R. d. L. Meira (2005), "Quality Attributes for a Component Quality Model", in Tenth International Workshop on Component-Oriented Programming, WCOP Glasgow, Scotland.*
2. *B. Cornelia, G. Craig, M. Angus and W. Paul (2001), "Establishing a Measurement Programme for the World Wide Web", in Proceedings of the 2001 Symposium.*
3. *B. Fogg, Persuasive Technology: Using Computers to Change What We Think and Do (Interactive Technologies): Morgan Kaufmann, December 2002.*
4. *C. Calero, J. Ruiz and M. Piattini, "Classifying Web Metrics Using the Web Quality Model", in Online Information Review, April 2005, pp 227-248.*
5. *C. Consortium (2000), "Poor Project Management – Problem of E-Projects", Cutter Consortium: The Cutter Edge.*
6. *Abhishek Anurag et al., A Case Study Of Existing Quality Model Based On Defects & Tests Management Of Embedded Software System, International Journal of Computer Science Engineering and Information Technology Research (IJCSEITR), Volume 8, Issue 2, May-June 2018, pp. 15-30*
7. *C. Fombrun and M. Shanley (1990), "What's in A Name? Reputation Building and Corporate Strategy", the Academy of Management Journal, pp 233-258.*
8. *C. Georgenes (2007), How to Cheat in Adobe Flash CS3, (How to Cheat in): The Art of Design and Animation: Focal Press.*
9. *G. Brajnik (2001), "Towards Valid Quality Models for Websites", in 7th Conference on Human Factors and the Web, Madison, Wisconsin.*
10. *G. Gledec (2005), "Quality Model for the World Wide Web", in 8th International Conference on Telecommunications - ConTEL2005 Zagreb, Croatia, pp 281-287.*
11. *Luis Olsina1 and Gustavo Rossi (2000), "Web Engineering: A Quantitative Methodology for Quality Evaluation and Comparison of Web Applications", Doctoral Thesis presented at Facultad de Ciencias Exactas, Universidad Nacional de La Plata, Argentina.*
12. *N. Bevan (1999), "Quality in Use: Meeting User Needs for Quality", Journal of Systems and Software, vol. 49, pp 89-96.*

13. *R. Fitzpatrick (1999), "A Process for Appraising Commercial Usability Evaluation Methods, Human-Computer Interaction: Ergonomics and User Interfaces", in HCI International Munich: Lawrence Erlbaum.*
14. *S. Gibbs (1995), "Multimedia Component Frameworks", in Object-Oriented Software Composition, pp 305-319.*
15. *S. J. Barnes and R. T. Vidgen (2006), "Data Triangulation and Web Quality Metrics: A Case Study in E-government", Proceedings of the Information and Management, Publisher Elsevier Science Publishers B. V. Amsterdam, the Netherlands, pp767 - 777.*