Advances in Computer Science and Information Technology (ACSIT)
p-ISSN: 2393-9907; e-ISSN: 2393-9915; Volume 4, Issue 2; January-March, 2017, pp. 75-79

http://www.krishisanskriti.org/Publication.html

A Modified ISO 9026 Software Quality Model to Measuring COTS Components

Reena[1] and Pradeep Kumar Bhatia[2]

[1,2]GJU&ST University, Hisar
E-mail: [1]reenabokan1987@gmail.com, [2]pkbhatiagju@gmail.com

Abstract—*The quality of software is essential to an organization in making their commercial software. Good or poor quality to software plays an important role to some systems such as embedded systems, real-time systems, and control systems. As Component-based Software Development (CBSD) starts to be effectively used, some software vendors have commenced to successfully sell and licensed COTS components. One of the most critical tasks in CBSD is the selection of the COTS components that meet the user's requirements. Several models have been developed to help different type of user with quality issues. Since 2000, the construction of software of software started to depend on generated or manufactured components and it gives rise to new challenge for assessing quality, because these components introduce new concepts such as configurability, Reusability, availability and better quality. Our proposals have shown how to deal with the functional aspects of this evaluation process. Although, there is a lack of appropriate quality models that allow an effective assessment of COTS components. In this paper we propose a new quality model for CBSD based on ISO 9126 that defines a set of quality attributes and their associated metrics for the effective evaluation of COTS components. The goal of the new model is to guide organizations that are in the process of building COTS – based system to evaluate and choose the appropriate products and that is essential to be success of the entire system. In this paper, some new features added in previously existing model (ISO 9126 AND Dromey model) for better evaluation of COTS components. In the proposed model, new high level characteristic portability is inculcated along with some sub – characteristic such as reusability, adaptability and replaceability. Some new sub characteristics also added such as scalability, configurability, stability and self – contained.*

Keywords*: Quality Model, CBSE/ COTS, ISO 9126 Model, McCall's Model, And Alvaro Model.*

1. INTRODUCTION

Research on software quality is as old as software construction and the concern for quality products arise with the design of error – free program as well accuracy when used [2]. According to the IEEE Standard Glossary of Software Engineering [3,4,5] the quality of software product is defined as (1) the degree to which a system, component or process meets specified requirement and (2) the degree to which a system, component or process meets the needs or expectations of a user. Software used product can be custom made or commercial of the shelf software (COTS) that is build to fulfill the general needs of an industry and government. Over the last decade, the use of COTS product to implement significant portions of a software system has grown in both government as well as industry. Organizations that adopt a COTS –based system methodology generally expect more rapid and less costly system construction. Government organizations are particularly encouraged to use COTS products. According to market analysts the use of commercial off-the-shelf (COTS)-based products in computing, in data storage, security, networking, and collaboration tools is accelerating in U.S. Department of Defense (DOD) command, control, communications, computers, intelligence, surveillance, and reconnaissance (C4ISR) applications till 2015. Cloud computing and big- data technologies will complement use of COTS-based smart phones, tablets, wireless networks, and productivity applications of all kinds through 2020, according to analysts [7]. COTS component selection poses some questions to be addressed such as:

- How to make feasible the comparison of described COTS components from a given domain when selection is required?
- How features of COTS components may be reconciled with requirements?

To answer these questions a new model has been built that supports a standard set of quality characteristics suitable for evaluating COTS components, along with newly defined sets of sub-characteristics associated with them. The new model also avoids some of the limitations found in the existing models. The new model ignores quality characteristics that are not applicable to COTS components and is empowered with new ones that are important.

2. STUDY OF SELECTED BASIC AND DERIVED QUALITY MODELS:

A quality model has become an important requirement commercially as well as in government sector, so that industry avoids purchasing COTS component of questionable quality. A basic for a quality model might be to consider some of the

existing quality approaches. In [8], Jeffrey also suggested the software certification triangle. Software certification must consider one or more of three approaches:

- Accrediting developers for demonstrating specific skill sets.
- Assessing the codes behavior.
- Certifying that processes are properly followed.

The McCall's, Boehm's, FURPS, ISO/IEC 9126, ISO/IEC TR 15504-2 1998(E) and Dromey's models are called basic models, whereas Rawashdeh, Bertoa, GEQUAMO are known as non basic or derived models. All of the above models have been studied, and analyzed with the purpose of developing a new quality model that: (i) overcomes some of the existing limitations, (ii) features standard quality characteristic and associated sub-characteristics integrated from basic and non basic models in an attempt to combine the advantages of both types, and (iii) associates all categories of stakeholders involved in the process with the appropriate set of quality characteristics. As a result, the proposed model will be utilized as a tool for evaluating and selecting the appropriate COTS product.

2.1 Basic Quality Model

The structure of McCall's quality model is mainly three major categories product revision, product transition, product operation.The category product revision consists of maintainability, testability and flexibility quality attributes. The product transition category consists of portability, reusability, interoperability quality attributes and the product operation category consists of a set of quality attributes that includes correctness, reliability, usability, integrity, and efficiency. McCall's model aims to bridge the gap between users and developers by focusing on a number of software quality factors that indicate the views of both users and developers. The functionality of the software product was the one of aspect not considered directly by this model [10].

Boehm added some new features to McCall's model with emphasis on the Maintainability of a software product. This model also includes considerations involved in the evaluation a of software product with respect to the utility of the program. However it is quiet similar to McCall's in that it presents a hierarchy of characteristics, each of which contributes to overall quality. Boehm model is based on a wider range of characteristics and incorporates 19 criteria. Boehm's notation of successful software includes characteristics of hardware performance that are missing in McCall model [12].This model looks at utility from various dimensions, considering the types of user expected to work with the system once it is delivered. FURPS takes into account the five main characteristic that make up its name: Functionality, Usability, Reliability, Performance, and Supportability. One aspect not considered by this model is that it fails to take into account the software product's Portability [10].

The software industry's need to standardize the evaluation of software products using quality models, to answer this question the ISO (International Standard Organization) proposed a standard, which specifies six areas of importance for software evaluation, and, for each area, specifications that attempt to make the six areas measurable, these include: Functionality, Reliability, Usability, Efficiency, Maintainability and Portability. One of achievement of the ISO 9126 model is that it identifies the internal and external quality characteristics of a software product, but it does not show very clearly how these aspects can be measured [10].

Dromey developed a model consisting of eight highlevel quality attributes, namely the same six from ISO 9126 plus Reusability and Process Maturity [13, 14]. He suggested a more dynamic idea for modeling the process on three prototypes concerning quality. These are:

* Implementation quality model.
* Requirements quality model.
* Design quality model.

Through this section, our purpose is to understand the relationship between the attributes (characteristics) properties of the software product that affect the attributes of quality [10]. The disadvantage of this model is associated with Reliability and Maintainability. It is not feasible to judge both characteristics Reliability and Maintainability of a system before it is actually operational in the production area.

2.2 Derived Quality Model

From 1990, the development of software was based on components (CBSD). The Non Basic models Software development concentrated on the use of Commercial COTS Components. Various activities of the development of a product based on COTS available in the market are:

- Selection of appropriate COTS component from repository.
- Adaptation e.i, exact matching of COTS component with requirement.
- Integration e.i, integrate various COTS components with the help of glue code.

The Quality Model Bertoa [13] is based on the ISO 9126 Model [15]. 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 COTS components.

The next derived model is called GEQUAMO (Generic, Multilayered and Customizable Model), was created by E.Georgiadou [13] and consists of the gradual breakdown into sub layers of features and is intended to encapsulate the

various user requirements in a dynamic and flexible way. The user (enduser, developer, and manager) can build their own model reflecting the emphasis (weight) for each attribute or requirement.

Another derived model is Alvaro model considers a framework for the certification of software components in order to establish the elements of quality components [1, 2]. This frame considers four modules: 1) 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.

The last derived model of our study is Rawashdeh Model [11]. The main objective of this model is 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 quality model [12] that is:

- Identify a small group of high level quality attributes, then using a top down technique each attribute is decomposed into a set of sub- attributes.
- Distinguish between internal and external metrics. Internal measure internal attributes such as specifications, size or source code, and external system behavior during testing operations.
- Identification of users (end-user, developer) for each quality attributes.
- Built the new quality Model is with ideas of ISO 9126 and Dromey Model.

The derived models can be either product oriented (GECUAMO), or for particular domains (Bertoa) or adapted from the point of view of a user (Rawashdeh). Overall these model useful for COTS components. The quality characteristics found in the majority of the models are: Efficiency, Maintainability, Portability, Usability, Functionality and reliability, these have been present in more recent models, as described above. In order to examine, compare and come to a conclusive result, it is shown in Table 1 below. The table shows six hierarchy models among the ones that have been studied here, each associated with its software quality attributes.

2.3 Addressing the Issues : In these quality models McCall's ignored Functionality, Boehm's contains a diagram without any suggestion about measuring the quality characteristics, and FURPS model ignored Portability, ISO-9126 does not show very clearly how the attributes can be measured. Thus, there is an absence of any kind of metrics that could help in evaluating quality characteristics objectively, in particular when the underlying software project is based on a COTS component. In addition, none of the existing models attempts to relate certain characteristic with the type of stakeholders (enduser, analyst) that are most concerned with such characteristic. Among the available models for evaluating software quality, none of McCall's, Boehm's, FURPS, ISO/ICE, and Dromey explicitly considers product efficiency and process-effectiveness. An efficient product is obtained only when correct physical design and programming practices are applied; product effectiveness is determined by activities involving requirement identification, interface design and general network design.

3. PROPOSED METHODOLOGY

The purpose of creating our new work is to build one suitable to model for a variety of COTS-based systems. The base for building our model is the ISO 9126, and some features from dromey and alvaro models, simply because it includes the common software quality characteristics that are supported by the other six models. In this new model some features also taken from Alvaro model. The model is design using four steps [19] that are:

1. Identify a small set of valid, high-level quality attributes, then a top-down approach decompose each attribute into a set of subordinate attributes.
2. Distinguish between internal and external metrics. For COTS components, it is important to observe such difference, specifically; the internal metrics measure the internal attribute of a product (e.g. specification or size) during the design and coding phases, known as 'white box' metrics [13]. Whereas external metrics (Reliability, Performance) specialize in the system behavior during testing and component operation, from an outsider view. In fact, external metrics, known as 'black-box', are more appropriate for COTS components.
3. Identify Stakeholders (type of users) for each high-level quality attribute.
4. Put the piece together; constructing the new model that implements ideas from international standards: ISO-9126, Dromey, some features are also inculcated from the Alvaro model.

3.1 Implementation of methodology

Step1. The proposed model is based on ISO 9126 because it includes the common software quality characteristics as shown in table 2.

Table2. Quality characteristics of Boehm, McCall, ISO 9126, FURPS and Dromey model.

Software Quality	Boehm	McCall	ISO 9126	FURPS	Dromey
Testability	x	x		x	
Correctness		x			
Efficiency	x	x		x	x

Reliability	x	x	x	x	x
Understandability	x			x	
Functionality			x	x	x
Flexibility		x			
Human Engineering	x				
Integrity		x			
Interoperability		x			
Maturity				x	
Maintainability		x		x	x
Changeability					x
Portability	x	x		x	x
Reusability		x			x
Usability	X	x			

Then we apply tailoring on ISO 9126 that harness COTS evaluation requirements. Brief discussion of high level quality characteristics of ISO 9126 is already done above.

Step2. The next step in the proposed methodology, distinction between internal and external metrics, is already described and reasoning led us to consider the external metrics 'black-box' as more appropriate for COTS components and internal metrics as white box, sometime it is used to measure the external metrics indirectly. In this work, a set of new characteristics added for the product as well as for the process explained in [22] has been adopted in order to help the stakeholder for appropriate selection of COTS component.The emergence in the software industry of component-based technology has opened the door for the development of new solutions to improve application portability. These are:

- **Portability** is the high level characteristics added in proposed model. As we all know portability is the ability of a software to run (with little or no modification) on different platform, or with different version of the same hardware or program. Components enable the building of families of products based on standard building blocks. The COTS components also enable portable, since a component is not linked to a particular hardware platform.
- By adding portability three new features also add up for the product. These **are Reusability, Replaceability and Adaptability**. The idea of component-based software development is to build an application by assembling a set of reusable components that can be independently deployed, configured and connected together, thus reusability is an important factor for COTS component. The replaceable component is one for which another component can be substituted without substantial modification to the new component. The COTS component should be replaceable so that project developer can replace if it not meet the requirement. Adaptability is also an important feature. With changing requirement, or environment the COTS product can adapt the new change.
- **Configuration** is the sub-characteristics of product; configuration is the arrangement of functional unit according to their nature, number, and chief characteristics. Some COTS component requires a concerted configuration effort, not only with the configuration capabilities within the COTS component, but also the operating system may require a reasonable amount of configuration to meet the requirement of the component. Our proposed model considered thisfeature also.
- **Scaleability** is the capability of a system network or process to handle a growing amount of work. Scaleability is the sub – characteristic of process.
- **Stability** is the sub characteristics of the COTS product. It check, how long will the vendor commit on paper to 1) producing, 2) supporting the version of COTS product under consideration. 3) How financially stable is the vendor and parent company.
- A component is **self contained** deployable software module containing data and operation which provide services to other component.

Step3. The third step is to identify stakeholder, stakeholder is a person, group or organization that has interest or concern in an organization. Stakeholders can affect or be affected by the organization's / company actions, objectives and policies. Stakeholders may be the end-user who interacts with the system and everyone else in an organization that may be affected by its installation.

Table 3: Quality Model for COTS Components

Characteristics	Sub-characteristics (Product)	Sub- characteristics (Process)
Functionality	Accuracy, Security	Suitability, Interoperability, Compliance, Compatibility
Reliability	Recoverability	Maturity
Usability	configureability	Learnability, Understandability, Operability, Complexity
Efficiency	Time behavior, Resource behavior	Scaleability
Maintainability	Stability	Changeability, Testability
Manageability	Quality management	
Portability	Adaptability, Reuseability, Replaceability	

Table 3 shows the characteristics that constitute our new model. Consequently, we have adapted to our model the common characteristics that are found and agreed upon by the majority of the existing models, and also these are consistent with COTS component evaluation criteria. New

characteristics are added, and these are necessary to empower our proposed model. Next, a new set of sub-characteristics has been defined and added associated with each high-level characteristic that is supported by the proposed model, this was done by breaking down the characteristics into two categories; one set supports the development process (process) and the second one supports the operational state on the production area (product). The new features are highlighted in figure 2.

Step 4: Finally, new model is builds for the evaluation of COTS component. This new model will be advanced with all new characteristics. The proposed model will help the vendors.

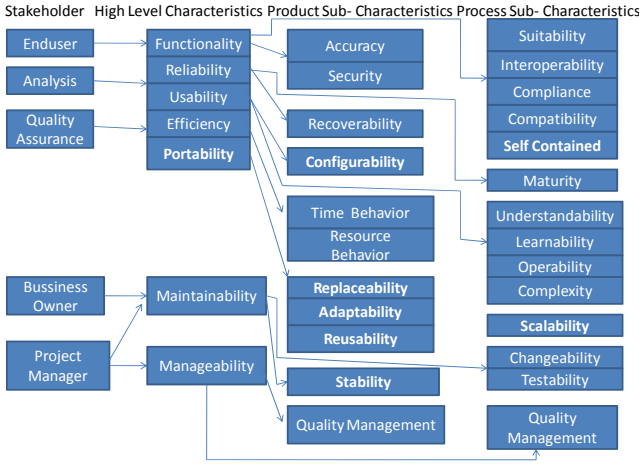

Figure 2. A proposed Quality Model For Component/ COTS – based system

4. CONCLUSION AND FUTURE WORK

The number of COTS-based systems being built commercially continues to increase, that why the need for a model that ensures quality characteristics of such systems becomes a necessity. Several models, including basic and non-basic, specializing in measuring the quality of software products have been described. The characteristics of such models have been studied, analyzed and their limitations outlined. We defined a four-step approach to guide the process of building the new model that is specialized in evaluating COTS components. The analysis study assisted us to benefit from existing general quality models and simultaneously avoiding repetition of such limitations. Subsequently, justified high-level characteristics have been projected and a new set of sub-characteristics has been added and defined for each one. This is apt by breaking down the characteristics into two categories; 'the Process' and 'the product'. The difference between internal and external metrics led us to realize that external metrics is more appropriate for COTS components. A major advantage of the new model is the addition of some new features, so that project developer can easily select desired COTS component. Although our new model has a features specialization and improvement over existing models, but it lacks some quality characteristics. This can be accomplished in future research.

REFERENCES

[1]A Chandrasekar, M.SudhaRajesh, and M. P. Rajesh, "A Research Study on Software Quality Attributes", *International Journal of Scientific and Research Publications* 4, no. 1, 2014, pp. 1-4.

[2] X Cai, R. Michael, L. K. F. Wong, and K Roy, "Component-based software engineering: technologies, development frameworks, and quality assurance schemes", In *Software Engineering Conference, 2000. APSEC 2000. Proceedings. Seventh Asia-Pacific*, IEEE, 2000, pp.372-379.

[3]A Alvaro, E. S. Almeida, and S. L.Meira, "Quality attributes for a component quality model," *10th WCOP/19th ECCOP, Glasgow, Scotland* 2005,pp. 31-37.

[4]Jenie, A.P. "Software Engineering: Software Process and Modelling", 2011.

[5] Kalaimagal, S and R Srinivasan, "A retrospective on software component quality models," *ACM SIGSOFT Software Engineering Notes* 33, no. 6, 2008,pp.1-10.

[6] PBotella,X. B. Illa, J. P.Carvallo, X Franch and C Quer. "Using Quality Models for Assessing COTS Selection", In *WER*, 2002, pp. 263-277.

[7]A Rawashdeh and B.Matalkah, "A new software quality model for evaluating COTS components", *Journal of Computer Science* 2, no. 4, 2006, pp. 373-381.

[8] V Jeffrey, Certification: Reducing the hidden costs of poor quality. Reliable Software Technologies, IEEE Software, 1999.

[9]Musa K. and J.Alkhateeb, "Quality model based on cots quality attributes", *International Journal of Software Engineering & Applications* 4, no. 1, 2013.

[10] M. F.Bertoa and A.Vallecillo, "Quality attributes for COTS components", 2002.

[11] Mittal S. and P. K. Bhatia, "Software component quality models from ISO 9126 perspective: A review", *IJMRS's International Journal of Engineering Sciences* 2, no. 2 2013.

[12]Callaos N. and B.Callaos, "Designing with systemic total quality", *EDUCATIONAL TECHNOLOGY-SADDLE BROOK NJ*- 34, 1994, pp. 29-29.

[13]Bertoa M. and A.Vallecillo "Quality Attributes for COTS Components," I+D Computación, Vol 1, Nro 2, 2002, pp.128-144.

[14] Miguel J.P., D Mauricio and G Rodríguez," A review of software quality models for the evaluation of software product", *International Journal of Software Engineering & Applications (IJSEA*), Vol.5, No.6, November 2014, pp. 31-53.

[15] ISO/IEC IS 9126-1. (2001). Software Engineering - Product Quality – Part 1: Quality Model. International Organization for Standardization, Geneva, Switzerland.