TO DESIGN NEW QUALITY MODEL FOR EVALUATING COTS COMPONENTS

Reena[1], Manoj[2]

[1]Gurugram University, Gurugram

reenabokan1987@gmail.com

[2]Guru Jambheshwar University
GJU, Hisars,

manojkoslia91@gmail.com

Abstract: The purpose of this paper is to build an ISO 9126 based new quality model that describes quality characteristics for the successful assessment of COTS software and to guide the industries that are making COTS based system. Some new features are added as well as some dropped in existing model ISO 9126 for better evaluation of COTS components. In the proposed model, some new sub-characteristic such as availability, resource utilization, and capacity associated with high-level characteristic efficiency (performance) is included. Some new sub-characteristics are also added such as scalability, configurability, stability and self-contained.

Keywords: COTS, Quality model, ISO 9126, Component-Based Software Development (CBSD).

(Article history: Received: 16th January 2022 and accepted 28th June2022)

1. INTRODUCTION

Quality is one of the important parameters for evaluating any software. It is crucial for an organization to develop good quality commercial software. Some systems must always be good quality like real-time systems, control systems, and embedded systems. According to the analyst by 2023, the compatibility of Edge computing with a big-data paradigm will enhance the usage of COTS-based products such as wireless networks, smartphones, Internet-Of-Things (IOT) and tablets [4]. After 2000, the development of software products began to rely on inbuilt or existing components; hence it creates a new challenge in front of developing industries like quality assessment. To deal with quality issues, various models have been developed by researchers like Boehm, McCall's quality model. The component-based system introduces some new areas like reusability, configurability, availability and optimal quality. No appropriate quality model exists for the assessment of COTS software. 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 novel paradigm has been designed that endorse a standard range of quality attributes in addition to a newly identified range of sub-attributes relating to them appropriate for assessing COTS components. Few limitations identified in the existing paradigm are avoided by the proposed model. Quality attributes that are inapplicable for the COTS components are ignored in the proposed model and important one is added to make it empowered.

The paper is summarized as follows: study of selected quality models are discussed in section 2. Proposed methodology is presented in sections 3. Case study is given Section 4 and results and analysis part is presented in section 5. Conclusion part is discussed in section 6.

2. STUDY OF SELECTED BASIC AND DERIVED QUALITY MODELS

A quality model has begun to be significantly needed commercially as well as in the government sector so that industry avoids buying COTS components of questionable quality. The basics of quality models might be considered through existing quality models. The software certification triangle is suggested by Jeffrey [5]. Accreditation of software must follow one or more of three paradigms:

- Certify builders for showing their skills.
- Evaluation of the codes.
- Authorized activities are accurately followed.

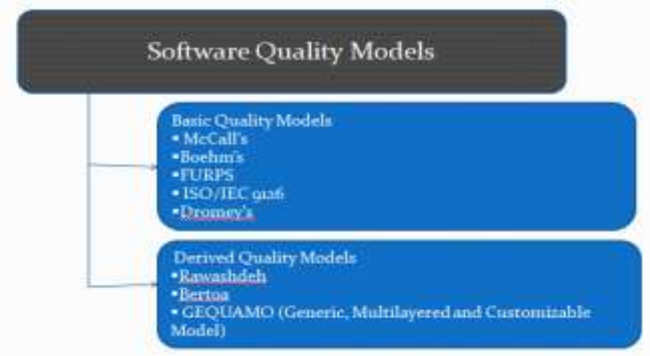

Fig. 1. Classification of Software Quality Models

Fig.1. depicts the classifications of software quality models. The McCall's, Boehm's, FURPS, ISO/IEC 9126 and Dromey's models are considered as basic quality models, while Rawashdeh, Bertoa, and GEQUAMO are considered as non-basic or derived models. For the development of the new quality model, all the above models whether it was basic or derived has been studied and analyzed. The aim of the proposed model is to overcome the limitations of the existing model. From basic and derived models, high-level quality characteristics with sub-characteristics are extracted in order to merge the advantages of both types. In the proposed model, all the classes of shareholders associated with a suitable range of quality characteristics are included. Consequently, the new model acts as a tool in the future for the evaluation and selection of appropriate COTS products.

2.1 BASIC QUALITY MODEL

Product revision, transition and operation are the three major parts of McCall's quality model. Flexibility, testability and maintainability quality characteristics make the product revision class whether reusability, interoperability, and portability quality characteristics contribute to product transition class. A set of high-quality characteristic that makes product operation class are usability, integrity, correctness, efficiency, and reliability. McCall's quality model targeting those high-level characteristics which are important from the point of the user as well as developer, hence it fills the gap among the users and developers. The functionality attributes are not directly considered by this model [6]. A few new attributes added by Boehm in McCall's model by focusing on the maintainability of a software product. The evaluation considerations include in this model corresponding to the use of the program. Hence, unlike McCall's, it introduces the order of attributes in which every attribute supported the overall quality of the product. Boehm model includes a large number of attributes and combines 19 criteria. Successful software Boehm's document incorporates all the attributes of hardware performance which are absent in the McCall model [7]. Five main attributes of FURPS are Reliability, Functionality, Performance, Usability, and Supportability. The portability attribute is not considered by this model. Using quality models, the software developer organization focuses to standardize the assessment of COTS products. For software assessment, a standard proposed by ISO that identified six regions of importance i.e. Usability, Functionality, Efficiency, Portability, Reliability, and Maintainability. The ISO 9126 model specifies the quality attributes of software into the internal and external quality attributes, although this model does not clarify how to achieve these aspects [8]. Dromey quality model consists of eight quality attributes by the addition of two quality attributes i.e. Process Maturity and Reusability in ISO 9126 [9, 10]. Dromey provides three modeling activities which are:

- A quality paradigm for implementation.
- Quality approach for the requirements.
- A quality paradigm for the design phase.

One of the drawbacks of the Dromey quality model is related to maintainability and reliability. Actually, it is impossible to determine both the attributes prior to system implementation in the production line.

Table I. Analysis of Basic Software Quality Models

Models	Number of attributes	Number of sub-attributes	Relationship
McCall	11	22	x:y
Boehm	7	18	x:y
FURPS	5	25	1:x
Dromey	7	-	x:x
ISO/ 9126	6	35	1:x

- 1:x - Various sub-attributes are subjected to one high- level quality attribute (e.g., FURPS and ISO 9126 model).
- x:y - Each high-level quality attribute is associated to one or several sub-attributes (e.g. Boehm's model).
- x:x - Each high-level quality attribute is associated with other high-level quality attributes (e.g. Dromey model).

Table I. depicts the relationship between the high levels of attributes and sub-attributes of basic quality models. Through this section, our purpose is to understand the relationship among the attributes which influence the quality of software product.

2.2 DERIVED QUALITY MODEL

After 1990, the focus of software developer industries was the shift from a traditional software development technique to Components-Based Software Development (CBSD). The usage of commercial components is emphasizing by a derived quality model in software development. Different types of activities involved in COTS-based software development are:

- Selection of appropriate COTS components from the repository.
- Adaptation i.e. exact matching of COTS component with the requirement.

- Integration i.e. integrates various COTS components with the help of glue code.

- **Bertoa Derived Quality:** Bertoamodel is derived from ISO 9126. This model differentiates attributes that are important for the assessment of particular COTS components.

- **Generic, Multilayered and Customizable Model (GEQUAMO)**: This was another derived model developed by E. Georgiadou. The above model composed of regular disruption into sub-layers of attributes. The design intention of this model is to fulfill the different customer needs powerfully. By deciding weight for each and every attributes, the end-user or application developer can make their own model [11].

- **Alvaro Model:** This has given software components certification framework [12] [13]. The framework of this model consists of four segments:
  - The model used to determine the quality features of the component.
  - The technical approval framework decides the paradigms that will be utilized to assess the characteristics given by the model.
  - To design comprehensive certification standard for component.
  - A set of metrics used to assess features of the components in a controlled environment define by frame.

- **Rawashdeh Model:** ISO 9126 and Dromey acts as base model for theRawashdeh model [RAW 2006]. To focus on the user requirement is the aim of this model. Four significant footsteps have been set up by this model to generate a quality model.

Table II. Analysis of Derived Software Quality Models

Models	Number of attributes	Number of sub-attributes (Process)	Number of sub-attributes (Product)
Bertoa	6	5	10
Alavaro	6	10	13
Rawashdeh	6	12	6

Table II showing that the number of high-level attributes of all three derived models is the same but the number of sub-attributes is different during both the phases run time as well as the life cycle of the product. The derived quality models can be user point of view (Rawashdeh) or either product-oriented (GECUAMO) or for particular domains (Bertoa). Overall, these models are useful for the COTS components assessment. Efficiency, Maintainability, Usability, Portability, Reliability, and Functionality are the characteristics considered in the majority of the models also available in more recent models.

2.3 ADDRESSING THE ISSUES

In McCall's quality models Functionality attribute has been ignored. An idea of how to assess quality features has not been provided by Boehm's quality model. The portability feature has been omitted by the FURPS model while in ISO-9126; it is not specifying how to assess quality attributes. Hence, no existing quality model is optimal, particularly in a case of software developed using a COTS. To produce an efficient product, the process followed to develop the software should be effective but the product efficiency feature has not been considered by the existing models.

3. Proposed Methodology

To design one appropriate model for all the COTS-Based Systems is the objective of this work. The base for creating a new model is the ISO 9126 and FURPS. Some features are also extracted from Alvaro and Rawashdeh models. ISO 9126 is selected as base models because it contains all features that are common in all six models. The proposed model is designed using four steps [14] that are:

- Select a valid small set of high-level characteristics, and then decompose each characteristic into a set of sub-characteristics using a top-down approach.
- For the COTS component assessment, it is necessary to differentiate among the internal and external metrics. Internal characteristics such as the size of the product during the design and coding phase are assess by internal metrics while external metrics assess the external characteristics like Reliability, performance during testing and implementation phase.
- Type of users (end-user, or application developer) is identifying for each and every high-level quality attribute.
- By putting the piece together, Build a new model that is based on ISO-9126 and FURPS, some features are also inculcated from the Alvaro and Rawashdeh models. Implementation of the methodologyis given below:

Step 1. ISO 9126 contains quality attributes of all six models, hence ISO 9126 is selected as a base for the proposed model as shown in table III.

Then ISO 9126 is customized that controls COTS assessment specifications. A brief discussion of high-level quality characteristics of ISO 9126 is already done above. Almost all the features of ISO 9126 model are adapted in proposed model except the portability, because a component designed based upon the frequency of re-use in different environments. Replaceability, adaptability, and reusability associated with portability are also dropped. It is mention in [15] that many people prefer to talk about Performance rather than (efficiency given in ISO 9126) and use other sub-classifications. So, some new sub-attributes such as availability, speed, and capacity associated with efficiency (performance) are added in the proposed model. Here, arrow symbol represent the presence of the attribute.

Table III. Quality Characteristics of Basic and Derived Quality Models

Software Quality	Boehm	M cCall	ISO9126	FURPS	Dromey	Bartoa	Alvaro	Rawashdeh
Testability	↑	↑	↑			↑	↑	↑
Correctness		↑						
Efficiency	↑	↑	↑		↑	↑	↑	↑
Reliability	↑	↑	↑	↑	↑	↑	↑	↑
Understandability	↑		↑			↑	↑	↑
Functionality			↑	↑	↑	↑	↑	↑
Flexibility		↑						
Human Engineering	↑							
Integrity		↑						
Interoperability		↑				↑	↑	↑
Maturity			↑			↑	↑	↑
Maintainability	↑	↑	↑		↑	↑	↑	↑
Changeability			↑			↑	↑	↑
Portability	↑	↑	↑	↑	↑	↑	↑	
Reusability		↑			↑		↑	
Usability	↑		↑	↑	↑	↑	↑	↑
Supportability			↑	↑				
Scalability							↑	
Manageability								↑

Step 2. External metrics more appropriate for COTS components as already discuss. Sometime, internal metrics indirectly measure the external metrics. In this work, in order to help the stakeholder for suitable selection of COTS component, a new set of attributes has been added for the product as well as for the process [16] such as availability, resource utilization, speed and capacity associated with high-level characteristic efficiency (performance). Some new sub-characteristics are also added such as scalability, configurability, stability and self-contained.

To refine performance, component-based technology emerges as the latest technology has unlocked the development of novel solutions. These are:

- **Efficiency:** Efficiency is responsible for measuring the accurate performance of software product under define conditions.

- **Availability** is the degree to which a software component is operational and available when required for use.
- **Resource utilization** is the use of resources by software product for appropriate time in pre-defined conditions.
- **Capacity:** Capacity is the maximum amount of work done by component or software in a set period of time.

➢ **Configuration:** Configuration is the sub-characteristics of the product. Configuration is the arrangement of the functional unit according to their nature, number, and main characteristics. Sometimes there is a need for composite configuration effort for a few COTS components, configuration capabilities not only required within the COTS component but it also required in some other software like an operating system.

➢ **Scalability:** The ability of a system to manage the growing need of the work load. Scalability is the sub-characteristics of the process.

➢ **Stability:** Stability is the sub characteristics of the COTS applications. It track the vendor status such as commitment for producing COT product, Support for the various COTS version and financial relation between vendor and producer company.

➢ **Self-contained:** A component is a self-contained deployable software module containing data and operation which provides services to other components.

Step 3. The third step is to identify stakeholder, a stakeholder is a person, group or organization that has interest or concern in an organization. Through the organization's objectives, actions and policies, the Stakeholders can be affected or can affect the others. End-user or any other person in an organization acts as stakeholders who operate on the system.

Table IV. Attributes and Sub-Attributes of Proposed Quality Model

Attributes	Sub - Attributes (Product)	Sub - Attributes (Process)
Functionality	Accuracy level, Security	Suitability, Compliance, Interoperability, Compatibility, **Self-Contained**
Efficiency (Performance)	Time and Resource behavior, **Availability, Capacity**	**Scalability**
Maintainability	**Stability**	Changeability, Testability
Reliability	Suitability, Recoverability rate	Frequency of Maturity
Usability	**Configurability**	Learnability rate, Understandability level, Operability, Extent of Complexity
Manageability	Quality management	Quality management

Manageability | Quality management | Quality managen

Table IV depicts the attributes and sub-attributes of the new quality model. The proposed quality model is designed by adapting common attributes of the majority of existing models that are appropriate for the COTS component selection process.

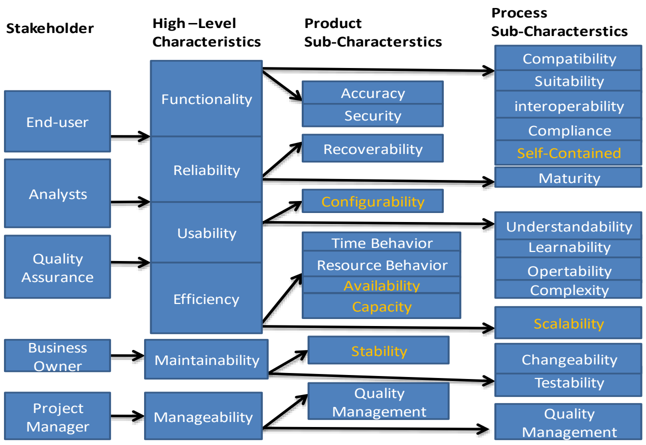

Fig. 2. Architecture of New Quality Model for Evaluating COTS-Based System

To empower the propose quality model, some new attributes are also added. A novel set of sub-attributes availability, speed, throughput and capacity associated with performance high-level attributes has been described and included. The new features are highlighted in fig. 2.

Step 4: Finally, a new methodology is designed for the assessment of the component. This new model will be advanced with all new characteristics; hence it will help the vendors in a better way.

4. CASE STUDY (MOBILE APPLICATION)

The proposed model is implemented on mobile application. The output is compared to existing study and we found that the proposed perform better as compared to existing model. The QFD to be used in this paper is in helping the developers know which quality attributes matter most to the customer. This is by collecting the customer requirements that have been mined from customer reviews. This will base on paper by [17] that mined customer reviews from 20 renowned applications in Apple store. They actually inspected 6390 reviews across 15 deferent classes. They found the most notable requirements were as showed up in Table V.

Table V. Client Quality attributes

Complaint type	Rank	Median (%)
Functionality Inaccuracy	1	26.68
Features demand	2	15.13
Crashing of application	3	10.51
Problem related to network	4	7.39
Design of interface	5	3.44

Exclusion of features	6	2.73
Hidden cost	7	1.54
Compatibility	8	1.39
Privacy and Ethical	9	1.19
Unresponsive app	10	0.73
Uninteresting content	11	0.29
Resource heavy	12	0.28
Net specific		**13.25**

The twelve recognized quality issues from convenient application customers will be used as the customer essentials for our QFD. The standard for this assessment was ISO/IEC 25010, which is the current standard for quality for all item including versatile applications used in previous studies [18].

5. Results AndDiscussion

There are five critical stages in realizing the QFD to ensure the quality referenced by customers is deciphered in the huge quality ascribes of the outcome.

Stage1. Customer Quality requirements: In this quality necessities of the customer are gathered. This is done through dismembering customer reviews from applications stores. Twelve quality issues are represented as showed up in table V. They are named "Customer Quality requirements" (CQR) which in table V are the dissent type.

Table VI. Absolute Weight

		Functional suitability				Performance Efficiency				Compatibility		Usability							Reliability				Security					Portability					Mainatainability			
Customer quality characteristic	Importance	Functional completeness	Functional Correctness	Functional	Self –Contained	Time Behaviour	Resource Utilization	Capacit	Scale-ability	Co-existence	Interoperability	Appropriateness	Learnability	Operability	User Error protection	User Interface Aesthetics	Accessibility	Configu	Maturit	Availability	Fault Tolerance	Recoverability	Confidentiality	Integrity	Non-repudiation	Authenticity	Accountability	Modularity	Reusability	Analyzability	Modifiability	Testability	Adaptability	installability	Replaceability	Stability
Functional Error	26.68	c	b	c					b	a		a			c	a		b				b	c										c	a		c
Feature request	15.13	b	c	c	c				b									c																		
App crashing	10.51	a	a	a		c	c	b	b		c		a	C	b	a		a			b	b										c	c	a		b
Network problem	7.39					c	b		c		a							b							c	a	a									
Interface design	3.44				b				b			b	b	b	b	b	c	b	c																	b
Feature removal	2.73	c	c	b	a	a	a	a		c	a	c	a	a	c	a	a	c																	c	c
Hidden cost	1.54								c									c					c	b	a	c	b									
Compatibility	1.39	c	c	c	c	c	c	b	b	b	b	c						b			c								a		a	a	a	a		c
Privacy and Ethical	1.19								c							c		c				b	b	b	b	b	b									
Unresponsive app	0.73	b	b	b	b	b	b	b	c	c	c	c	a	c	c	a	a			a	a	c								a			c	c	a	a
Uninteresting cont	0.29	a	a	b	a																															
Resource heavy	0.28				b		b	b	b	c	c							b			a	c					a	a		c			c	c		b
Absolute weight		**229.1**	**315.24**	**173.86**	**90.54**	**67.17**	**114.03**	**118.92**	**549.42**	**50.41**	**57.19**	**125.55**	**44.93**	**67.41**	**215.97**	**75.18**	**13.73**	**424.9**	**10.32**	**0.73**	**99.77**	**348.45**	**95.37**	**24.57**	**34.42**	**22.72**	**32.24**	**0.28**	**1.39**	**0.84**	**1.39**	**32.92**	**115.99**	**41.61**	**8.92**	**221.2**

Stage2. Rating the criticalness of the customer quality attributes which is done by calculating the repeat of complaints or sales on explicit quality trademark. This is taken by the paper made by [5]. There are 12 customer complaints that are trapped in Table 6.

Stage3. Standard quality: This depicts the quality to the extent properties that can be assessed by a known standard. In this QFD the quality standard is given by the ISO/IEC 25010 that sets standards for a wide scope of programming including flexible application programming. Architects can use the thirteen quality credits and 42 quality sub-characteristics to measure the idea of the outcome.

Stage4. Quality relationship structure: An organization is made to choose the association between the customer quality essential (CQR) and the quality standard (SQR). It checks the association between what the customer needs and standards available to ensure they get it. In QFD, relationship esteem (RV) is resolved. The way of thinking describes the associations as strong that is distributed the value 9, moderate that is consigned the value 3 and weak that is given out the value 1. Where there is no relationship no value is given as shown in table VI. The relationship structure is showed up in table 6 [19][20].

b-denotes 9 c-denotes 3 a-denotes 1

Stage5. Standard Quality trademark weight by significance: The connection between the customer quality requirements and the standard quality to fulfill them can be resolved. First we compute the Standard Quality Characteristic (SQR), which is given by the measure of copying the relationship value (RV) with the customer quality relative weight (CQRW).

$$SQW = \sum_{SQ=1}^{n} (RV * CQRW) \qquad .... (1)$$

Previous Study Results

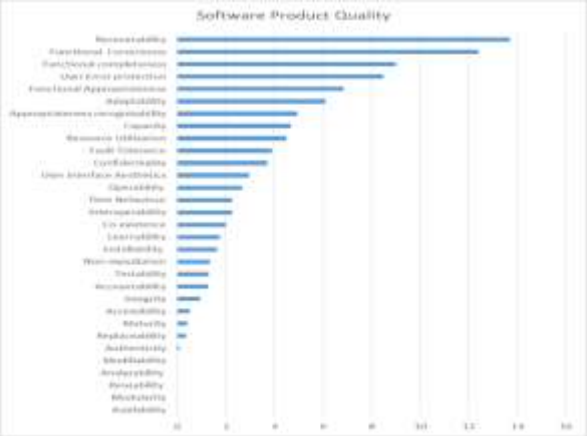

Fig. 3. Previous Study outputs

In previous study, the essential features of mobile application are functionality, suitability and usability. Portability and maintainability are least important as depicted in fig. 3.

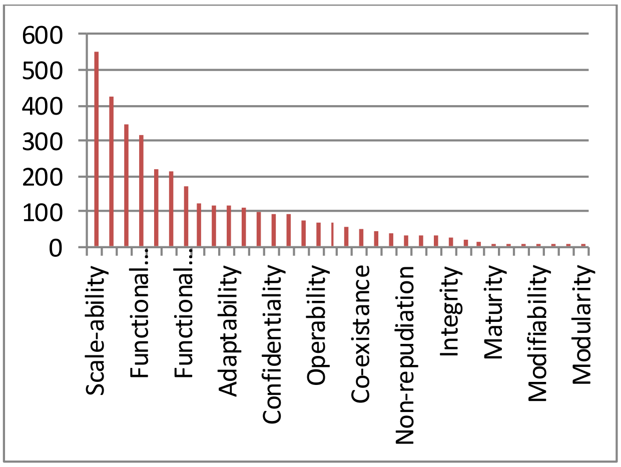

Fig. 4. Current model Results

After implementation of proposed COTS model on case study, scalability, recoverability, stability and functionality are the most important feature in mobile application while accessibility, replace-ability and modifiability are the least important shown in fig. 4. During utilization, quality is assessed by user using certain essential features like usefulness, pleasure and effective.

6. Conclusion

The analysis of existing quality models helps us to take the benefit from them and also avoid the limitations. A new quality model for assessment of the quality of COTS software has been designed based upon ISO 9126 and FURPS. The aim of the proposed methodology is to give direction to those industries that are building software system based on COTS paradigm. In the proposed model, some new sub-characteristics such as availability, resource utilization, and capacity associated with high-level characteristic efficiency (performance) are included. Some new sub-characteristics are also added such as scalability, configurability, stability and self-contained. The proposed model is implemented on mobile application. Customer reviews from 20 renowned applications in Apple store has been collected and 6390 reviews inspected across 15 deferent classes. The twelve recognized quality issues from convenient application customers are used as the customer essentials for our QFD. The standard for this assessment was ISO/IEC 25010, which is the current standard for quality for all item including versatile applications used in previous studies. After implementation of proposed COTS model on case study, scalability (with absolute weight 549.42) boost performance, stability (424.9) support usability and self-contained sub-attributes of functionality are the most important feature according to Client Quality attributes (table V) in mobile application while replace-ability and modifiability are the least important.

The output is compared to existing study and we found that the proposed model perform better as compared to existing model. Although our new model has traits specialization and refinements, yet it misses some quality attributes that can be achieved in future research work.

REFERENCES

[1] Chandrasekar, A., Rajesh, S., & Rajesh, P. (2014). A research study on software quality attributes. *International Journal of Scientific and Research Publications*, *4*(1), pp. 14-19.

[2] Cai, X., Lyu, M. R., Wong, K. F., &Ko, R. (2000, December). Component-based software engineering: technologies, development frameworks, and quality assurance schemes. In *Proceedings Seventh Asia-Pacific Software Engeering Conference. APSEC 2000* (pp. 372-379). IEEE.

[3] Alvaro, A., Almeida, E. S., &Meira, S. L. (2005). Quality attributes for a component quality model. *10th WCOP/19th ECCOP, Glasgow, Scotland*, 31-37.
[4] Jenie, A.P. "Software Engineering: Software Process and Modelling", 2011.
[5] Kalaimagal, S., & Srinivasan, R. (2008). A retrospective on software component quality models. *ACM SIGSOFT Software Engineering Notes*, *33*(6), 1-10.
[6] BotellaLópez, P., Burgués Illa, X., Carvallo Vega, J. P., Franch Gutiérrez, J., &Quer, C. (2002). Using quality models for assessing COTS selection. In *WER 2002: V International Workshop on Requirements Engineering* (pp. 263-277).
[7] Rawashdeh, A., &Matalkah, B. (2006).A new software quality model for evaluating COTS components. *Journal of Computer Science*, *2*(4), 373-381.
[8] Voas, J. (1999). Certification: reducing the hidden costs of poor quality. *IEEE Software*, *16*(4), 22.
[9] Musa, K., &Alkhateeb, J. (2013). Quality model based on COTS quality attributes. *International Journal of Software Engineering & Applications*, *4*(1), 1.
[10] Bertoa, M. F., &Vallecillo, A. (2002). Quality attributes for COTS components. I+ D Computación, 1 (2): 128–144.
[11] Mittal, S., & Bhatia, P. K. (2013). Software component quality models from ISO 9126 perspective: A review. *IJMRS's International Journal of Engineering Sciences*, *2*(2).
[12] Callaos, N., &Callaos, B. (1994). Designing with systemic total quality. *Educational technology*, *34*(1), 29-36.
[13] Manuel, F., & Antonio, V. (2002). Quality attributes for COTS components. *I+ D Computacion*, *1 (2)*, 128-143.
[14] 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*.
[15] Lochmann, K. (2014). *Defining and Assessing Software Quality by Quality Models* (Doctoral dissertation, TechnischeUniversitätMünchen).
[16] Huang, C. Y., Leung, H., Leung, W. H. F., & Mizuno, O. (2012). Software quality assurance methodologies and techniques. *Advances in software engineering*, *2012*.
[17] Khalid, H.,Shihab, E.,Nagappan, M.,& A.E.H.Member, (2015).WhatDoMobileAppUsersComplain About?,"*Inst.Electr. Electrornic Eng.*, pp. 1–7,2015.
[18] Kuria, J. (2019). Defining Mobile Quality Attributes Using Quality Function Deployment, International Journal of Scientific and Research Publication, Vol. 9(6), pp 718-725.
[19] Karnavel, K., & Dillibabu, R. (2014). Development and application of new quality model for software projects. *The Scientific World Journal*, *2014*.
[20] Nistala, P., Nori, K. V., & Reddy, R. (2019, May). Software quality models: A systematic mapping study. In *2019 IEEE/ACM International Conference on Software and System Processes (ICSSP)* (pp. 125-134). IEEE.

AUTHORS PROFILE

Dr. Reena Boken

Dr. Reena Boken is currently an Assistant Professor in Engineering and Technology Department, in Gurugram University Gurugram. She obtained her PhD in Computer Science & Engineering Department from Guru Jambheshwar Science & Technology University, Hisar Haryana, India. Her current research interests are Software Testing, Quality Model, COTS, IOT and machine learning. Presented many papers in Conferences and Seminars, and published a lot of papers in National and International Journals.

Manoj obtained his Bachelor's degree in Computer Science and Engineering from Guru Jambheshwar University of Science and Technology, Hisar, India. Then he obtained his Master's degree in Computer Science and Engineering from Guru Jambheshwar University of Science and Technology, Hisar, India. Currently, he is an Assistant Professor at the Department of Computer Science and Engineering, GJUS&T, Hisar, Haryana, India. His current research interests are Wireless Networks, Mobile Computing and machine learning. Presented many papers in Conferences and Seminars, and published a lot of papers in National and International Journals.