Proceedings of the 2nd International conference on Electronics, Communication and Aerospace Technology (ICECA 2018)
IEEE Conference Record # 42487; IEEE Xplore ISBN:978-1-5386-0965-1

User's Perspective of Software Quality

Dr.Jagannath Singh, Nigussu Bitew Kassie Assistant Professor,
School of Computer Engineering
KIIT Deemed to be University,
Bhubaneswar-751024, Odisha, India.
Email:jagannath.singhfcs@kiit.ac.in, kingbitew@gmail.com

Abstract—In a decade ago software engineering exploration progressively focused on software quality improvement and assessment, however the major of these researchers focus on the internal/development perspective i.e. they did not consider users point of views. Nevertheless users mainly attention in the quality of performance intended functions efficiently without knowing how the software product is developed, how work form inside, or its internal quality. The achievement of software companies is totally relies upon the users satisfaction, which they choose to use a software product or not.Hence ,in software development strong attention must given to users satisfactions. Software Quality research on users satisfaction help the developers to develop a software that meet users requirements and accepted by end users. In this article based on well known software quality models and the users of software product we intended to discuss the factors as well as sub factors of software quality that influence user's satisfactions.

***Keywords*—Software Quality Factor; Software Quality Model; Software Quality and Users Perspective**

I. INTRODUCTION

In a decade ago software engineering exploration increasingly focused on software quality improvement and assessment. The achievement of software companies is totally relies upon the users satisfaction. The quality of software is increasingly being vital and the users demanding higher quality.At its most fundamental, increasing the quality of software product normally means diminishing the number of defects in the software product. Defects can result from the occurred the development process that introduced faults into the software products. The most cost effective way of handling a defect is to prevent it. In this situation, software quality is accomplished through process improvement, increasing staff knowledge and skill, etc.

Research in software engineering exploration progressively focused on software quality improvement and assessment, however the major of these researchers focus on the internal/ development perspective i.e. they did not consider users point of views [1]. Recently, software quality exploration on user's satisfaction this help developers to develop software that meet users requirements and accepted by end users.

II. USER'S VIEW OF SOFTWARE QUALITY

This perspective of quality emphases on user's satisfaction. A product is said to have a good quality when users are satisfied in using it. This is to mean that if the product meets the purpose for which it was designed and developed in the first place and users are satisfied in using it, then it has a good quality. Users have different views on product usability depending on their needs. This view of quality indicates a more personalized views of users on product quality in a specific context of operation and functionality on the product.

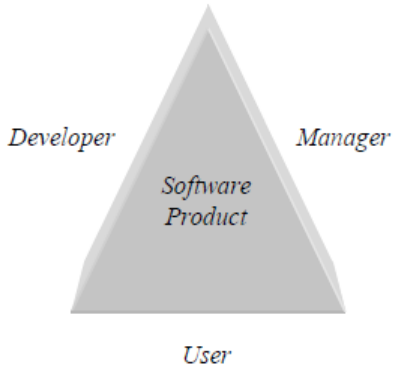

Fig. 1. Software quality views

III. RELATED WORK

A. SOFTWARE QUALITY MODELS

Many researchers depicted quality in hierarchical way to understand and measure software quality. These models are comprehensive tools applicable for assessing quality of any type of software product at any stages of its development life cycles. The models illustrate how software quality attributes relate one another and the assessment approaches that should be employed to assess quality. In this sections, some of the foremost software quality models are briefly reviewed.

1) McCall's Model

The model is shown in figure 3 below. McCall identified 11 quality factors and every factors at least one quality criteria are defined as shown in figure below, one the first column that contains the three operations in software development life cycle. Quality factors that can't be measured directly are placed. The quality criteria on the right side indicate the tangible and specific attributes of software products to which values can be assigned to measure the higher levels quality factors [2].

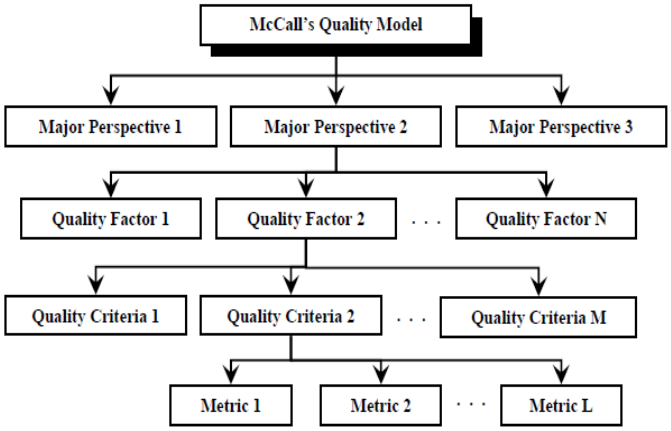

Fig. 2. The structure of McCall's quality model

The quality factors define features of a system and the quality criteria are the result of simplifying the factor into more specific and measureable characteristics. Some of the criteria are attributed to one or more quality factors.

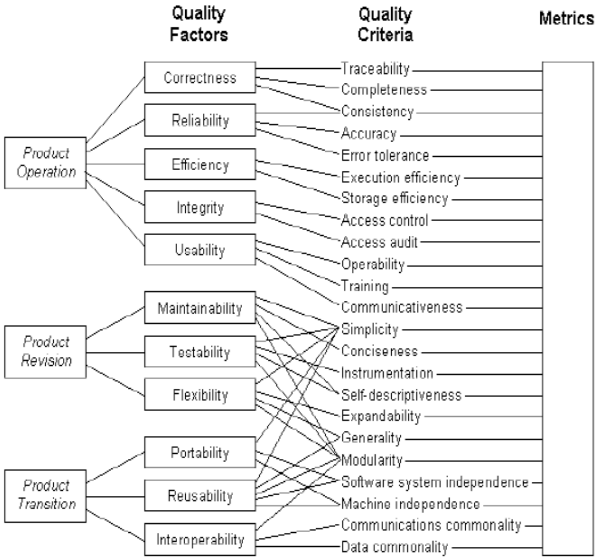

Fig. 3. McCall's quality model

2) Boehm's Model

This model has similar attributes like the McCall's model in that it represents a hierarchical list of quality attributes categorized as high level attributes, intermediate attributes, and primitive attributes. The high level attributes named General Utility and it indicates the overall system quality [3], [4]. It comprises of three categories of characteristics, namely:

- As-is utility (ease to use, reliable and efficiency)
- Maintainability (ease to modify and ease to retesting)
- Portability (capability of adopting new environment)

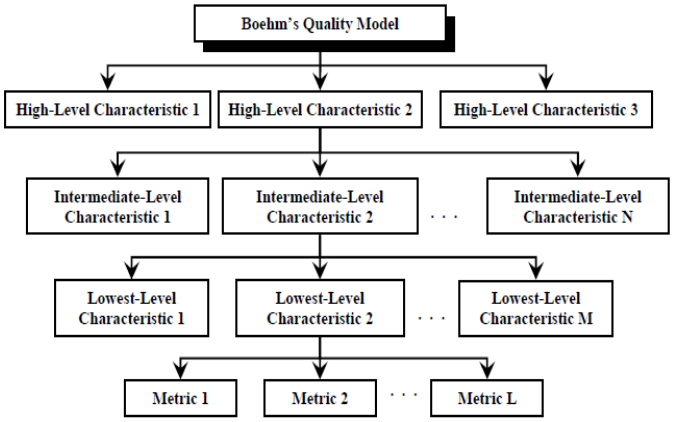

Fig. 4. The structure of Boehm's quality model

The intermediate characteristics consist of seven quality attributes each related to the three high level attributes. Portability refers to the property of a software product that can still continue to function, even if it is transferred to a different environment. Maintainability indicates how easy it is to modify the product, re test and understand the changes. As-is utility indicates the efficiency, reliability and easy to use characteristics of a product [5], [6].the intermediate attributes consist 7 quality characteristics shown in the middle of the model in figure 4.the primitive attributes at the lowest level of the hierarchy indicates of a foundation for defining quality attributes.

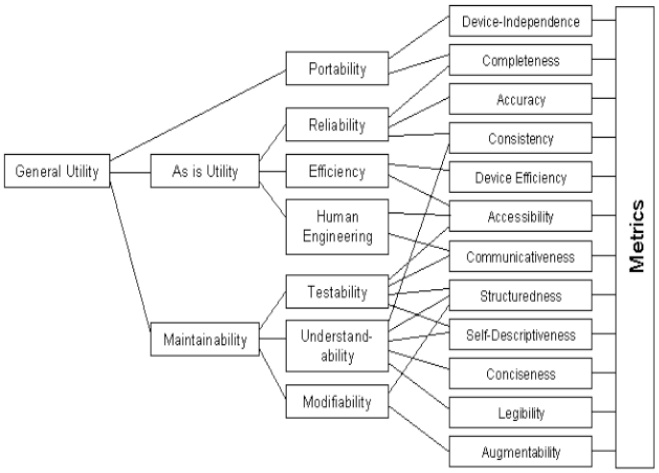

Fig. 5. Boehm's software Quality Model

3) FURPS Model

This model classifies quality attributes as functional and non-functional. FURPS stand for functionality, usability, reliability, performance and supportability. This model does not include the portability quality attributes and also does not clearly show the metrics or assessment approaches to use. FURPS+ is the extended version of FURPS model by IBM Rational software [7]. Figure 6 represents FURPS model.

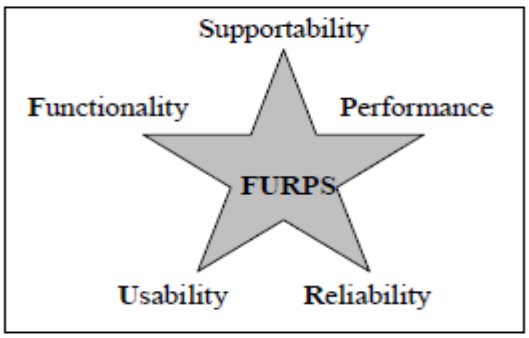

Fig. 1. Fig. 6. FURPS software quality model

Functionality includes the capabilities and security attributes that correspond with the purpose of the software and the security mechanism at which the product operates. Usability includes consistency, user interface and aesthetics the product. Reliability indicates the frequency of failure, recoverability, accuracy and mean time between failures. The Performance of a product indicates the functional requirements of the product like: speed, efficiencys and so on. Supportability consists of the attributes like testability, adaptability, maintainability, etc.

4) Dromey's Model

This model emphasizes the idea that software quality characteristics necessary for assessing quality of software products should match properties of the software products. Dromey views a software product as having various components; each one carrying tangible quality properties [9]. This model consists three basic elements:

- Components of the product model
- Tangible quality carrying properties
- High level quality attributes

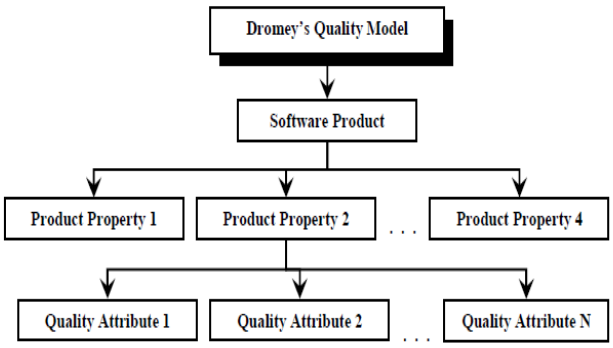

Fig. 7. The structure of Dromey's quality model

The model assumes that the components of a product model consists descriptive,contextual, internal and correctness software quality factors [2].

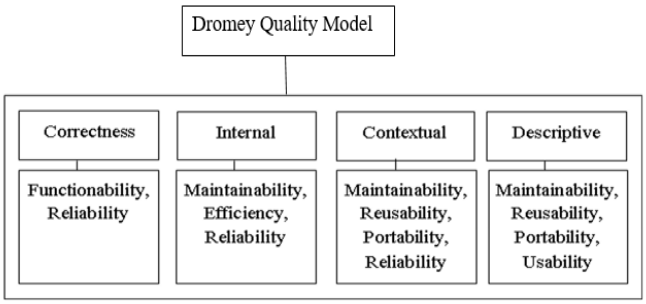

Fig. 8. Dromey's software quality model

5) ISO 9126-1 Quality Standard

The International Standard Organization (ISO) 9000 gives rules for quality assurance [9]. ISO 9000 is process oriented approach towards quality management [10]. It processes designing, documenting, implementing, supporting, monitoring, controlling and improving [11]. The ISO 9126 address software quality form the product perspective through its four parts i.e. Quality model, External model, Internal model and quality in use model. ISO 9126-1 series of standards (ISO 9126, 2001-2003) address software quality from the product perspectives through its four parts.

- Part I Quality model
- Part II External model
- Part III Internal model
- Part IV Quality in use model

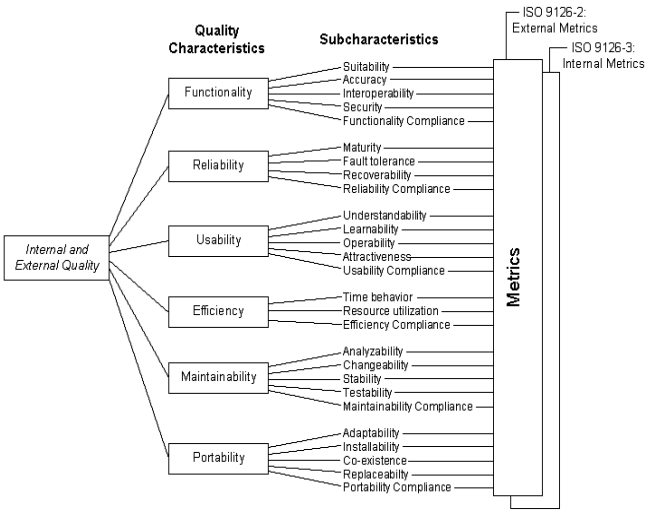

Fig. 9. ISO 9126-1 quality standard

The external and internal quality characteristics are shown with the three layers in figure 9 . As can be seen from the lists of the quality characteristics, the model shares similar quality characteristics from McCall and Boehm models. Even though it consists of characteristics, sub characteristics and quality measures; the quality characteristic list is not complete and fixed. So that according to the type of the software under evaluation and the reasons behind the evaluation, necessary

characteristics, which are not mentioned in the model, can be introduced. The ISO model therefore acts as a starting point for conduction software evaluation; it can be adopted to include essential quality characteristics of the software product under consideration, so to speak.

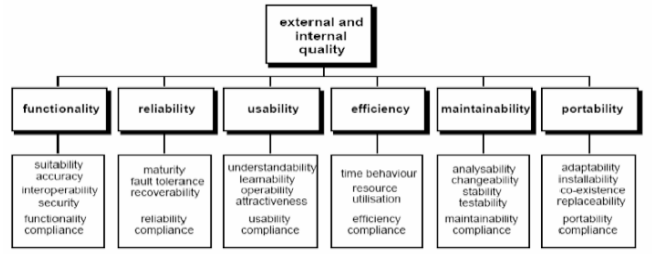

Fig. 2. Fig. 10. ISO 9126-1 model internal and external approaches

Fig. 11. ISO 9126-1 quality in use model

IV. USER'S PERSPECTIVE SOFTWARE QUALITY FACTORS

By end user's a number of software characteristics are considered. After abroad literature review [12],[13], this study has establish the most commonly occurring software quality attributes. After identifying the most often occurring 27 software quality attributes, this study perceived these attributes from the end user's perspective.In below we discussed 10 important factors according to user's perception.

A. Functionality

It considered as crucial factor in a software quality. Functionality has three sub factors:

- Suitability - indicates that software provides sets of appropriate functions for specific user's requirements.
- Accuracy - is the capability of a software to deliver accurate results.
- Security - the capacity of software to control unauthorized access of information and data. Figure 12 represents characteristics of the software Functionality.

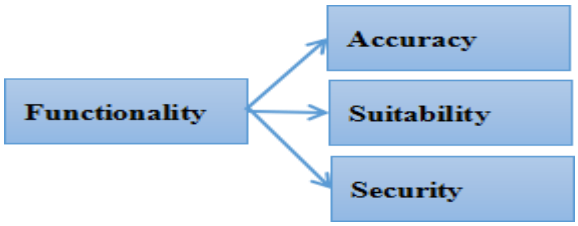

Fig. 12. Software Functionality Characteristics

B. Reliability

It is the capacity of a system to perform the intended function correctly without any failure. Reliability can be measured as the probability of the fact that a software will not fail to complete its proposed functions over a stated time interval. This factor consists of three sub factors:

- Maturity - is the capability of the software product to avoid failure due to errors in the software.
- Fault tolerance - is the capability of the software to maintain a certain level of performance during either faults in the software or infringement of its interface.
- Recoverability- is the capability of the product to reestablish to a certain level of performance and recover data affected during failure. Figure 13 represents Reliability software quality factor and its sub factors.

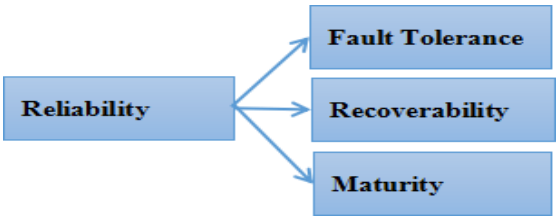

Fig. 13. Software Reliability Characteristics

C. Usability

It defined as "the ease of use for a given function" [11].It describes how well the system meets the user's requirement. The system should be user friendly and easy to learn. Usability has four sub factors:

- Learnability- indicates the learning effort users put to learn use the software.
- Training - indicates the effort required to teach how to use the software to the users of software .
- Communicativeness - it associated to how well the software communicates to the users.
- Usability compliance - software adhere to guides, regulations and standards related to usability. Figure 14 represents Usability software quality factor and its sub factors.

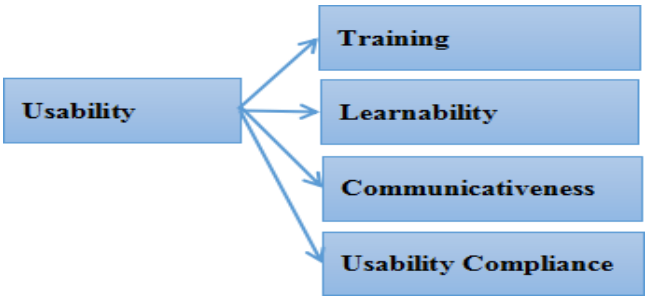

Fig. 14. Software Usability Characteristics

D. Efficiency

It defined as ”a set of attributes that convey to the relationship between the level of performance of the software and the amount of resources used under stated condition ” [10].This factors consists of two sub factors:

- Execution efficiency - indicates the time a software takes to perform tasks
- Storage efficiency - indicates the capacity of the software to use appropriate type and amount of resources when performing its functions. Figure 15 represents Efficiency software quality factor and its sub factors.

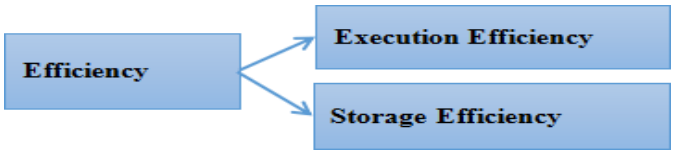

Fig. 15. Software Efficiency Characteristics

E. Maintainability

Maintainability is the capability of the system to under go changes without any difficulty. It is the probability that a system can be repaired in a defined environment within a specified period of time. It consists of three sub factors:

- Changeability - the capability of the software to enable implementation of specific modifications.
- Simplicity - the implementation of the software should be in the most understandability manner
- Self Descriptive- software that provide explanation of the implementation of the software.Figure 16 represents Maintainability software quality factor and its sub factors.

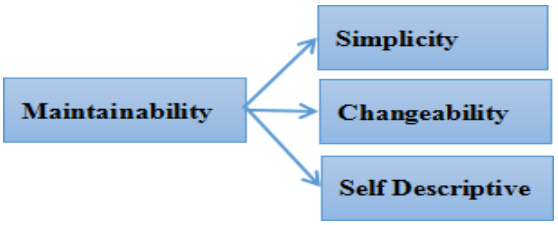

Fig. 16. Software Maintainability Characteristics

F. Portability

It is the capacity of a software to run well and effectively on multiple computer configurations, different operating system and different hardware. It has four sub factors:

- Adaptability - the capability of a software product to be adopted in specific environments without using additional effort.
- Installability - the capability of the software to be installed in specific environment.
- Co-existence - the capability of a software product to co-exist with independent software sharing common resources.
- Replaceability - the software product can be used in place of another software for a similar purpose.Figure 17 represents Portability software quality factor and its sub factors.

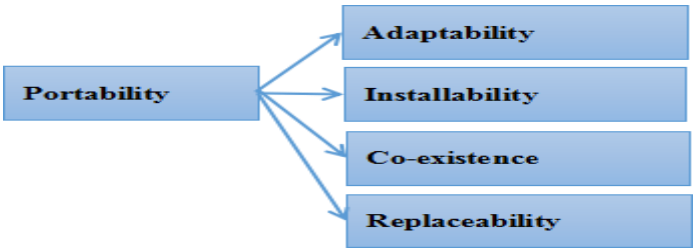

Fig. 17. Software Portability Characteristics

G. Understandability

It is the ease of learning/understanding the software. Software product should be simple to understand to the clients, so that the users can use the software easily and properly.It consists of three sub factors:

- Legibility - the level of ease which parts of the software may be recognized and organized into coherent pattern
- Structuredness -modularity of software
- Consistency - how far the software is consistent/ uniform in terms of GUI,terminologies and notations. Figure 18 represents Understandability software quality factor and its sub factors.

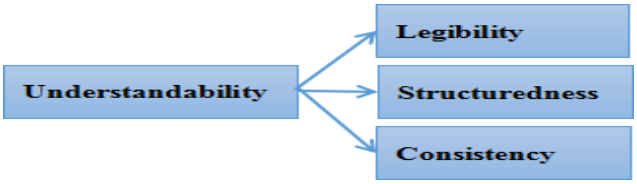

Fig. 18. Software Understandability Characteristics

H. Interoperability

It indicates the capacity of software that relates to its ability to interact with specified system. It consists of two sub factors:

- Communication Commonality - the software that provide the use of standard protocols.
- Data Commonality - the software that provide the use of standard data representation. Figure 19 represents Interoperability software quality factor and its sub factors.

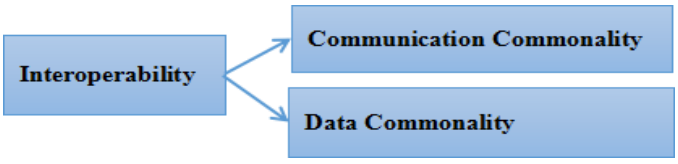

Fig. 19. Software Interoperability Characteristics

I. *Operability*

It indicates the capacity of the software to be easily operated by end users. Users of the software must comfortable with the manner through which services and content are presented in the software and be able to use the software easily without being frustrated or confused.It consists of four sub factors:

- Ease of Use - the degree to which the software product make easy for users to operate and control
- User Error Protection
- Technical Accessibility
- Technical Learnability

Figure 20 represents Operability software quality factor and its sub factors.

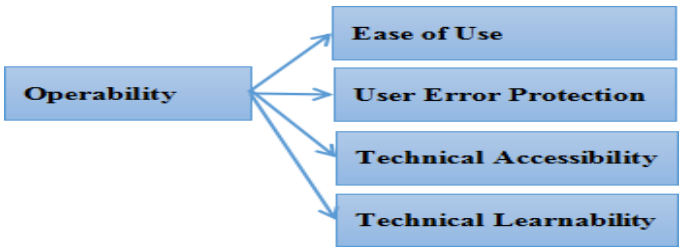

Fig. 20. Software Operability Characteristics

J. *Aesthetic*

The user interface of the software product should be attractive, enjoyable and pleasant enough for users to create an emotional appeal to use it. It consists of four sub factors:Consistent Text Layout,Page Layout,Font Size and Font Color. Figure 21 represents Aesthetic software quality factor and its sub factors.

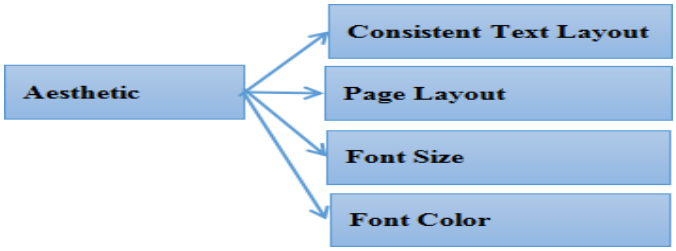

Fig. 21. Software Aesthetic Characteristics

V. CONCLUSION

In this study, the quality factors defined based on the well-known models and the perception of the software user's. This article found that Functionality, Reliability, Usability, Efficiency, Maintainability, Portability, Understandability, Interoperability, Operability, Aesthetic were attributes which have more impact on software quality based on user's perspective. Software quality factors as well as sub factors of software quality that influence user's satisfactions are identified and discussed.

REFERENCES

[1] Boehm, B., Chulani, S., Verner, J., & Wong, B. (2007, May). Fifth workshop on software quality. In Software Engineering-Companion, 2007. ICSE 2007 Companion. 29th International Conference on (pp. 131- 132). IEEE.

[2] M.-A. Ct, W. Suryn, and E. Georgiadou, "Software Quality Model Requirements for Software Quality Engineering," in Software Quality Management 2006, pp. 31-50.

[3] Boehm, B. W., Brown, J. R., & Lipow, M. (1976, October). Quantitative evaluation of software quality. In Proceedings of the 2nd international conference on Software engineering (pp. 592-605). IEEE Computer Society Press.

[4] Boehm, B., Brown, J. R., Kaspar, H., Lipow, M., MacLeod, G. J., & Merritt, M. J. Characteristics of Software Quality. 1978.

[5] P. Berander, L.-O. Damm, J. Eriksson, T. Gorschek, K. Henningsson, P. Jnsson, S.Kgstrm, D. Milicic, F. Mrtensson, K. Rnkk, and P. Tomaszewski, "Software Quality Attributes and trade-offs," L. Lundberg, M. Mattsson, and C. Wohlin, Eds.: Blekinge Institute of Technology, 2005.

[6] R. Fitzpatrick, "Software Quality: Definitions and strategic issues," 1996.

[7] Al-Qutaish, R. E. (2010). Quality models in software engineering literature: an analytical and comparative study. Journal of American Science, 6(3), 166-175.

[8] Dromey, R. G. (1995). A model for software product quality. IEEE Transactions on Software Engineering, 21(2), 146-162.

[9] J. A. McCall, et al., "Factors in Software Quality," Griffiths Air Force Base, N.Y. Rome Air Development Center Air Force Systems Command, 1977.

[10] Berander, P., Damm, L. O., Eriksson, J., Gorschek, T., Henningsson, K., Jnsson, P., ... & Tomaszewski, P. (2005). Software quality attributes and trade-offs. Blekinge Institute of Technology.

[11] ISO, "Software Engineering- Product Quality- Part 1: Quality model," 2001.

[12] Mebrate, T. (2010). A framework for evaluating academic websites quality from students perspective.

[13] https://simplicable.com/new/software-quality-examples