A Critical Survey on Quality Models in Software Engineering

Saurabh Pandey[#1], Sheikh Fahad Ahmad[#2], Mohd. Hussain[*3]

[#]*Deptt of CSE, Integral University, Lucknow, India*

[*]*Department of Computer Science, Madina University, Saudi Arabia*

***Abstract*— The quality of software is the major feature in present computing environment. It is a correctly organized and established collection of activities. Software quality models play a critical role in the measurement of software quality. There are a number of diverse quality models are used to build quality software. There are many software quality models used that improve the quality of software. Software Quality is directly related to maintenance efforts, so we can say that Quality software must be maintainable. In this research paper, we are discussing the diverse software quality models and reliability to achieve quality.**

BBe

***Keywords*— Software Quality, Quality Models, Quality Engineering , Reliability, , Boehm's Quality Model, Evans & Marciniak Quality Model, , ISO 9126, Dromey's Quality Model , FURPS Quality Model, SEI Quality Model, ISO 25000 (SQuaRE), Mc Call Quality Model, Deutsch & Willis Quality Model.**

I. INTRODUCTION

Quality is said to be degree with the help of level of customer satisfaction. The quality of a software product is now measured be an important aspect in business success. In real time systems such as weather meteorology quality is one of the mandatory parts because poor quality is entirely undesirable and may lead to injury, loss of human life, financial loss and hazards.

A high quality product is connected with a number of quality factors. These can be addressed in a number of quality models. Software Quality models represent abstract and simplified overview which positively affect the quality. In this paper, we discuss the following quality models:

- McCall's Quality Model (1977)
- Boehm's Quality Model (1978)
- Evan's & Marciniak Quality Model (1987)
- Deutsch & Willis Quality Model (1988)
- ISO 9126 Quality Model (1991)
- Dromey's Quality Model (1992)
- FURPS+ Quality Model 1992)
- SEI Model (1995)
- ISO 25000 (SQuaRE) Model (2011)

In totaling we will study a brief study of reliability engineering to enhance quality. Section 2 present an overview of the nine quality models. Section 3 describe Reliability as a major aspect to improve quality. At last, Section 4 which describes this topic with the help of some research surveys inside conclusion.

II. SOFTWARE QUALITY MODELS

A. McCall Model

First quality model was proposed in 1977-78 by Jim Mc Call and Joseph. This model is sometimes also known as General Electric's Model. The first one is three major perspectives:

- Product Operation
- Product Revision
- Product Transition

Each major perspective consists of quality factors. There are 11 different quality factors and each quality factor consists of quality criteria. There are 23 quality criteria.

- **Product Operation:** The software system capability to quickly understand and provided that relevant results required by the user. Its quality factors contain Correctness, Reliability, Efficiency, Integrity and usability.
- **Product Revision:** The ability of the system to feel changes which involves error correction as well as adaptation. Its quality factors include Maintainability, Flexibility and Testability
- **Product Transition:** Adaptability of the product to novel environments (changing hardware and software).

B. Boehm's Quality Model

- The quality model represents a hierarchical constitution of characteristics, each contributes the total quality. It have characteristics high level, intermediate level and lowest level.
- High level characteristics signify basic high level requirements to assess software quality. There are three characteristics, which are:
  - **As-is utility**: To address how the software product can be simply and efficiently use. Its intermediate

characteristics comprise Reliability, Efficiency and Human Engineering.

- **Maintainability:** To address how to easily adapt, retest and understand the software product with respect to quality. Its intermediate characteristics contain Testability, Understandability and Modifiability.
- **Portability:** The software product is portable if it performs fine in different computing environment (change in environment).

C. Evans & Marciniak Quality Model:

This model is complementary of McCall model, it describe some factors that we can be grouped into three categories , that are as follows:

- **Design:** Describes design constraints imposed on the system. It includes Correctness, Maintainability and Verifiability.
- **Performance:**Addresses performance related issue. It includes Efficiency, Integrity, Usability and Reliability.
- **Adaptation:** Change in environment, technology, software system should be adaptable to new environment. It include Flexibility, Portability, Reusability, Interoperability and Expandability.

D. Deutsch & Willis Quality Model:

This model is also an alternative model have emerge after the McCall Model. The essential and important factors beneath this model are as follows:

- Functional (Operation)
- Performance (Non-functional)
- Change
- Management

These attributes can be comprehensible with the help of McCall model as we already described as an alternative approach.

E. ISO 9126 Quality Model

Used for product evaluation. In that one standard and technical reports that are as follows:

- ISO IS 9126-1: Quality Model [ISO, 2001]
- ISO TR 9126-2: External Metrics [ISO, 2003]
- ISO TR 9126-3: Internal Metrics [ISO, 2003]
- ISO TR 9126-4: Quality in Use Metrics [ISO, 2004]
- The first document ISO 9126-1 quality model has two main parts consisting of Internal and External Quality model and Quality in Use model.
- External and Internal Quality model hold some attributes /characteristics that are:
- Functionality: Ability to meet the desired objectives and behaviour. It contain the subsequent sub-characteristics: Suitability, Accuracy, Security, Interoperability and Functionality Compliance.
- Reliability: Ability to improve performance with reference to time, resources etc. It contains the following sub-characteristics: Maturity, Fault tolerance, Recoverability, Reliability Compliance.
- Usability: Ability to be used by certain conditions. It contains the subsequent sub-characteristics: Operability, attractiveness and learnability.
- Efficiency: Ability of the software product to give appropriate performance, relative to the amount of resources used, under particular conditions. It contains the following sub-characteristics: Time behavior, Resource utilization, Efficiency Compliance.
- Maintainability: The ability of the software product to be modified. Modifications engage corrections, improvements or adaptation of the software in new environments. It contain the subsequent sub-characteristics: Analyzability, Changeability, Stability, Testability and Maintainability Compliance.
- Portability: The ability of the software product to be transferred to new environment (adaptability to change). It contain the subsequent sub-characteristics: Adaptability, Install ability, Co-existence, Replace ability,
- Portability Compliance: The quality in use model has the subsequent characteristics:
  - Effectiveness
  - Productivity
  - Safety
  - Satisfaction

F. Dromey's Quality Model:

This quality model has been projected by Dromey. It provide a quality evaluation from requirement, design and implementation phases. In this software product properties are as classified as follows:

- **Correctness:** It evaluate if some fundamental principles are violated, with quality attributes as Functionality and Reliability.
- Internal: Measures how well a software component has been deployed according to its intended use, with quality attributes as Maintainability, Efficiency and Reliability.
- Contextual: It handles with external influences on the use of software with quality attributes as Maintainability, Reusability, Portability and Reliability.
- Descriptive: Procedures the descriptiveness with quality attributes as Maintainability, Efficiency, Reliability and Usability.

G. FURPS and FURPS+ Quality Model

The FURPS model was offered by Robert Grady at Hewlett Packard, and then it has been complete by IBM Rational Software into FURPS+ where + indicates for design requirements, implementation requirements, interface requirements and physical requirements. This FURPS model

explains characteristics with two but various kinds of various requirements that are as follows :

- Functional Requirements (F) that can be distinct as expected input and output.
- Non-Functional Requirements are performance oriented, in which U stands for Usability, R stands for Reliability, P stands for Performance (includes functional requirements) and S stands for Supportability. All the FURPS+ requirements are under the category of non-functional rations.

H. SEI Quality Model

The month of December [1995-96], on the basis of technical information[CMU] Software Engineering Institute (SEI) available a detail report on the wattage of various quality factors (attributes) . This report provide four software quality attributes:

- Performance
- Dependability
- Security

This model is not acceptable by many quality engineers since it does not guarantee that software product is reliable.

I. ISO 25000 (SQuaRE)

SQuaRE stand for The Systems and software Quality Requirements and Evaluation, this model is effortlessly describe in two ways:

- A quality in use model that mitigate to the contact in exacting domain. This model is applicable to human-computer system principally includes computer systems in use as well as software product in use.
- A product quality model that narrate to dynamic property of the computer system and static properties of software. This model is suitable to both computer as well as software products.

III. RELIABILITY

As Reliability is ordinary with approximately to all models, so we can say that it is an important aspect to enhance quality. Software Reliability is the probability of failure-free software operation for a specific period of time in a specified environment. Reliability is necessary for quality and quality in turn gives customer satisfaction which will lead to reduce the presence of risk. At last we can say that:

Reliability → Quality → Customer Satisfaction

IV. CONCLUSION

We have studied different quality models, each of which consists of a number of characteristics to measure quality. The customer feedback makes an important role in measuring software quality. It must be noticed that phases in SDLC like requirement, planning, designing, coding and implementation should be done in a systematic way to have a quality product.
The quality model should be selected on the basis of the following factors:

- Reliability factor
- Requirement factor

The relationship between customer-developer to determine quality can be described as follows on the basis of their satisfaction-quality assurance level

- Customer-satisfied, developer-assured -The software is quality product
- Customer-not satisfied, developer-assured -The software is not a quality product.
- Customer-satisfied, developer-not assured -The software is a quality product.
- Customer-not satisfied, developer-not assured-The software is not a quality product.

So we can say that if developer is not fully guaranteed about quality, we can't say about the quality because the customer may be contented or not satisfied, in this our research indicates that if customer is satisfied about their product even developer have disbelief about what they develop are good or not, the customer is satisfied we can say that software is quality product without stressing developers thoughts. Many such cases occur in any organization, so this should also be taken gravely otherwise rigorous testing and evaluation may cause over financial statement that is totally unacceptable. The quality engineers should center on customer satisfaction in selecting quality model.

REFERENCES

[1] ISO/IEC 25010:2011. Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) System and software quality models

[2] Khashayar Khosravi and Yann-Gaël Guéhéneuc, 2004, A Quality Model for Design Patterns

[3] Namita Malhotra, Shefali Pruthi, 2012. International Journal of Recent Technology and Engineering (IJRTE), ISSN: 2277-3878,

[4] Rafa E.Al-Qutaish, Quality Models in Software Engineering Literature: An Analytical and Comparative Study. Journal of American Science, 2010

[5] International Journal on Advancements in computing Technology Volume III NOV 2011

[6] Alain Abran, ISO/IEC SQuaRE. The second generation of standards for software product quality. November 3-5, 2003 Marina del Rey, CA, USA

[7] Daniel Galin, 2004. Software Quality Assurance From theory to implementation. Pearson Education Limited

[8] Spector, A. Z. 1989. Achieving application requirements. In Distributed Systems.

[9] Lars-Ola Damm, Jeanette and Patrik Berander., Software quality facts and trade-offs. Blekinge Institute of Technology June 2005.

[10] Rafa E. Al-Qutaish, Jean Desharnais1 and Naji Habra2, AN ISO Standard MODELS That Measure SOFTWARE PRODUCT QUALITY, Alain Abran1

[11] J. Timber, Alan and Robert Grady A brief theory to Reliability to acheve Quality 1892, pp.68–73.