A Study on Software Quality Assurance in a recent trend

Amitav Saran[1],Khitish Kumar Gadnayak[2] ,Jagannath Ray[3]

[1,3]Associate Professor, Department of Computer Science Engineering, Gandhi Institute For Technology (GIFT), Bhubaneswar

[2] Assistant Professor, Department of Computer Science Engineering, Gandhi Engineering College, Bhubaneswar

ABSTRACT

Software quality is basic and essential in different types of organizations. When the quality of software is low may lead to damage of human life, mission disappointment, permanent injury, and financial loss in software engineering especially real-time software, control systems. SQA is a process that guarantees the developed software meets and complies with defined or standardized quality specifications. Successful SQA highly dependent on software quality measurements (metrics). Software quality model and software metrics are linked together through quality factors in order to offer measure method for software quality assurance. Software quality metrics emphasis on the quality aspects of the product, process, and project. Generally, software metrics are more closely related to the process and product metrics rather than project metrics.

Keywords- *Software Quality Assurance, Software Metric, Software Quality Factor*

I. INTRODUCTION

1.1 Background

Software Quality assurance (SQA) is a method to help accomplish quality. SQA can monitor that the software engineering process and methods used to guarantee quality. Software metrics manage the estimation of software product and software development process and it guides and evaluates software development [1]. SQA consists of three concepts: quality, software quality, and software quality assurance. Quality assurance has its root in guaranteeing the quality of an industrial physical software product, this is accomplished by inspection of the product and assessing its quality near is accomplishment or at the different stages of productions. Software, however, isn't as tangible as more physical products. Normally, a software artifact is its functionality and not its usage. There is no physical software product to assess; there is code and not continuously going with documentation. This "invisibility" nature of software adds to the difficulties of evaluating its quality. "Manufacturing products are visible; software products are invisible. Most of the faults in software products are invisible, as in the fact that parts of a software bundle may be absent from the beginning" [2].

Software quality estimation is about measuring to what degree the system or software processes desirable attributes. To guarantee software quality, we undertake software quality assurance and software quality control. Quality assurance differs from quality control in that quality control is a set of activities designed to evaluate the quality of a developed or industrial product. The assessment is conducted during or after the production of the product. Quality assurance, however, decreases the cost of assuring quality by a variety of activities performed during the development and manufacturing process.

1.2 Quality Attributes

Quality attributes are the general factors that affect the run time behavior, system design, and client experience. Every quality attribute can be used to quantify the product performance. Quality can be characterized differently. The quality definition may vary from individual to individual. But finally, there should be some standard. Quality is defined as [8].

❖ Fitness of reason-Edward Deming
❖ Degree of excellence-Oxford dictionary
❖ Best for user’s usage and marketing value- Feigenbaum
❖ The totality of attributes of an element that bears on its capability to satisfy stated or implied needs-ISO

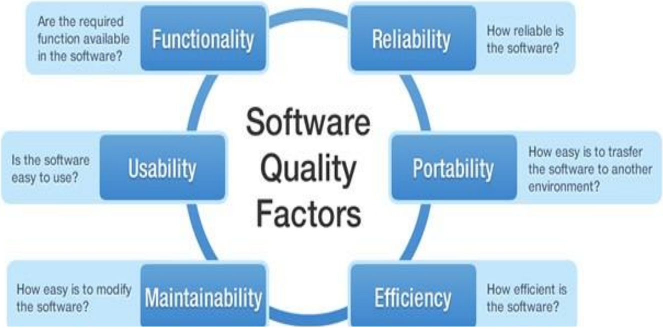
Are the required function available in the software?
Functionality
Reliability
How reliable is the software?
Software Quality Factors
Is the software easy to use?
Usability
Portability
How easy is to trasfer the software to another environment?
How easy is to modify the software?
Maintainability
Efficiency
How efficient is the software?

Figure 1 Software Quality Factors

1.3 Software Quality Metrics

A metric is a value stated with units related to a product. Software metrics categorized into product metrics, process metrics, and project metrics. Product metrics measure the efficiency of completing product goals for example size, complexity, design attributes, execution, and quality level. Process metrics measure the competence of execution of the product development process for example turnover rate. Project metrics measure the efficiency of the product development process, for example, schedule execution, cost execution, group performance.

1.4 Software Quality Standard

Different quality standards are available for controlling and keeping the general production process and production environment. Four main quality standards in software engineering are listed [8].

A. Software Engineering Institute Capability Maturity Model (SEICMM)

The Software Engineering Institute at Carnegie-Mellon University, initiated by the U.S defense division to help enhance software development processes. CMM created by the SEI has five levels of organizational ‘maturity’ that determine the efficiency in supplying quality software [8].

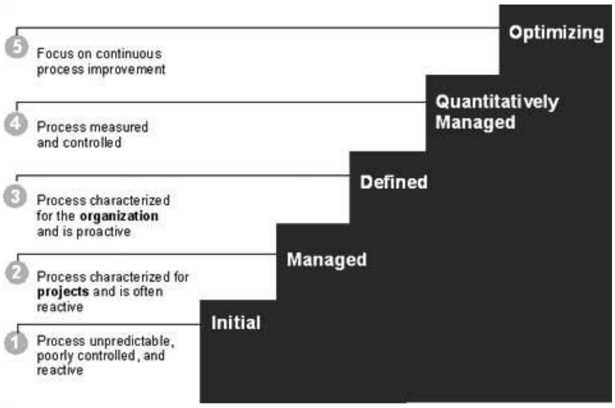
5
Focus on continuous process improvement
Optimizing
4
Process measured and controlled
Quantitatively Managed
3
Process characterized for the organization and is proactive
Defined
2
Process characterized for projects and is often reactive
Managed
1
Process unpredictable, poorly controlled, and reactive
Initial

Figure 2 Characteristics of the Maturity levels

which it will gossip until it sends the reply. Anonymous gossip is implemented over MAODV without much overhead. Buffer size is the limitation, when the old message gets stored in the buffer, there will be no place for the new messages, and the next limitation is the use of acknowledgment messages which is expensive in wireless networks.

B. International Organization For Standard

ISO 9000 provides guidelines for quality assurance, process-oriented approach towards quality management. It processes designing, documenting, implementing, supporting, monitoring, controlling, and improving [16].ISO 9000 defines the elements of a quality assurance system in general terms. These components include the organizational structure, procedure, processes, and resources required to implement quality planning, quality control, quality assurance, and quality enhancement. ISO 9001 is the quality assurance standard that applies to software engineering.

C. Institute of Electrical and Electronics Engineers (IEEE)

IEEE makes standards such as [8]:

- Standard for software testing documentation (IEEE/ANSI Std 829)
- Standard for software unit testing (IEEE/ANSI Std 1008)
- Standard for software quality assurance plans (IEEE/ANSI Std 730)

D. American National Standard Institute (ANSI)

American National Standards Institute is the essential business body in U.S, publishes some software-related standards in conjunction with IEEE and the American Society for quality.

II. OBJECTIVE & PROBLEM STATEMENT

Software characteristics might differ based on how the software is used. Therefore quality attributes also need to vary from software to software. The objectives are

- Identify software quality attributes that are relevant from the user perspective
- Study of Software Quality Factors
- To study various software quality models that enhance software quality assurance.

Software quality assurance is essential in software development processes no matter its size. SQA is a best practice for any software product. It helps to mitigate the risk involved in business logic like possible system failure, defects, the security issue, and data loss. Many software systems are available which provides similar types of functionality. But the user selects those which have good quality because they have not any idea about the architecture or language or database which are used for developing the model. The user selects the software based on the functionality or no security issue, bugs free, highly maintainable etc. Different requirements are:

1. Software quality is customer satisfaction
2. Software quality is conformance to requirements.
3. Software quality is a function of the percentage of the product that is free from defects.

III. LITERATURE REVIEW

3.1. What is Quality

Quality is an intangible concept.it is not easy to define it operationally, yet everybody feels when it is missing. The term good quality and poor quality are used in our everyday life to tell how good quality or bad product functions. Dictionary meaning of quality is usually focused on excellence or goodness of usage. Some technical writers define software quality in terms of "fitness for purpose" however more recent business thinking would not completely support this definition.

3.2. Different perspective of Quality

Based on diverse perspectives on quality, different definitions are given to quality. Five fundamentals perspectives of quality as introduced by Garvin [18] define quality differently. These five perspectives and definitions they give quality are described below:

3.2.1. Transcendental

This view associates quality with the "innate excellence" of a product.it emphasizes on the idea that quality is universally recognized and measurable, which indicates high achievements and inflexibility standard. This perspective designates the fact that product development ways strive for producing the ideal "best" characteristics of a product the user wants and the attempt to achieve the ideal, although the ideal best product may not be produced at the end. This strived for the ideal "best" characteristic of a product is the transcendental viewpoint of quality.

3.2.2 User-based

This perspective of quality emphases on users' satisfaction. A product is said to have a good quality when users are satisfied with 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 with 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 view of users on product quality in a specific context of operation and functionality on the product.

3.2.3 Manufactured based (Conformance to requirements)

This is another perspective that defines quality as the "conformance to requirements". A product should conform to a specific set of design requirements established at the beginning of the production. Any deviation from the requirements indicates low product quality.

3.2.4 Product based

In this perspective quality of a product is determined based on its internal characteristics and the weights assigned to them according to their level of importance. Attaching weights to the attributes is a cumbersome task. Besides, since the selection of the attributes and values might be susceptible to subjectivity, it may be difficult to arrive at an agreed-upon definition of quality, which is acceptable by all types of users. Nevertheless, this perspective is advocated by experts who believe that measuring internal quality can give contextual independent evaluation and help to improve the external quality or quality in the use of a product [19].

3.2.5 Value-based

In this approach, quality is defined in terms of the relations between the value and cost of a product. A product with good quality provides performance at an acceptable price and conformance at an acceptable cost. Existing quality definitions fall into the five basic perspectives through the views of the users and product manufacturers would be different. The implication of different perspectives of quality indicates the different viewpoints of users and developers. High quality for user is related to high performance and improved features of product/service that satisfies their needs. Developers, on the other hand, take a different line of thought, a product to be referred to as having high quality, it has to conform to outlined specifications [18].

3.3 Software Quality

Software quality includes all attributes and important features of a product or an activity that relates to the fulfillment of a given requirement. Wikipedia states software quality as “In the context of software engineering, software quality measures how well software is design, and how well the software conforms to the design. Software quality also defines how well the software fulfills with or conforms to a given requirement, based on functional requirements. That characteristic can also be described as the fitness for purpose of a piece of software or how it compares to competitors in the marketplace as a valuable product” [17].

3.4 Software Quality Assurance Model

The software quality model help to enhance the execution, such as imprints measured quality, increase client satisfaction, reduction of cost for quality. In this section five quality models are described, McCall’s quality model, Boehm’s quality model, Dromey quality model, FRUPS quality model, and ISO 9126 quality model.

3.4.1 McCall’s Software Quality Model

The McCall software quality model [3] has three quality of software product:

- Product transition -adaptableness to new condition
- Product revision –capability to undergo changes
- Product operation-will be operation characteristics

This model contains 11 quality factors. The quality factors describe various types of system characteristics and quality criteria are attributes to one or more of the quality factors. Figure 3 is represented as the factors and criteria of McCall quality model.

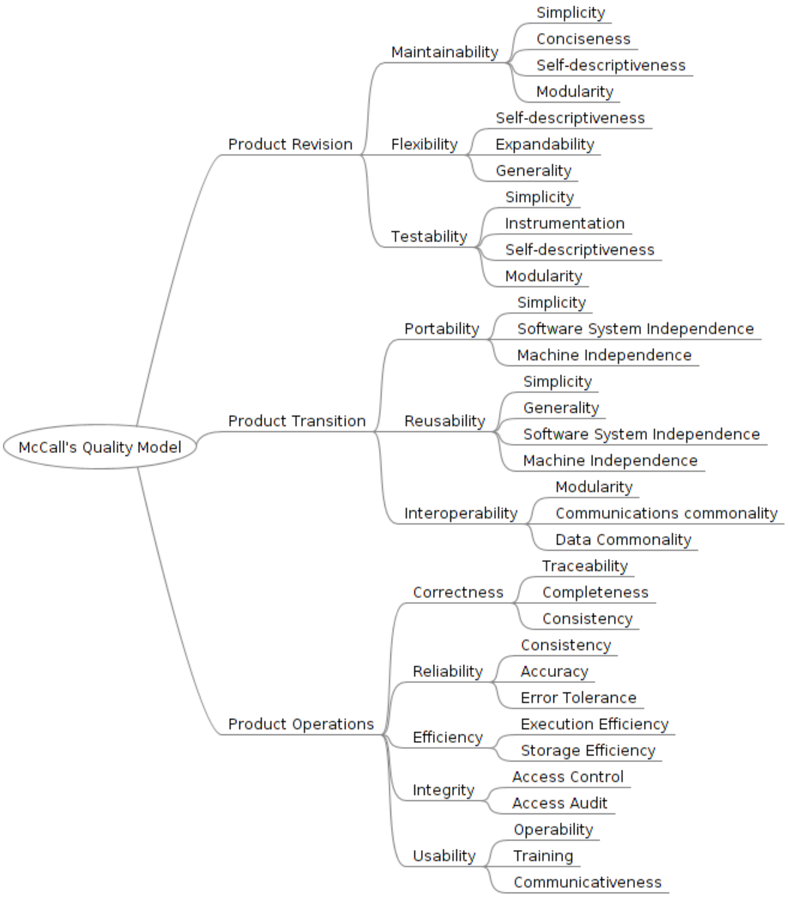

Figure 3 McCall Software Quality Model

3.4.2 Boehm's Software Quality Model

Boehm quality model attempts to automatics and qualitatively assess the quality of software. The high-level characteristics address three classifications; general utility into as unity, maintainability, portability.in the intermediate level of characteristics, Boehm quality model has seven quality factors like portability, reliability, efficiency, usability, human engineering, understandability, flexibility ([7],13]).Figure 4 denotes the quality factors and criteria of Boehm quality model.

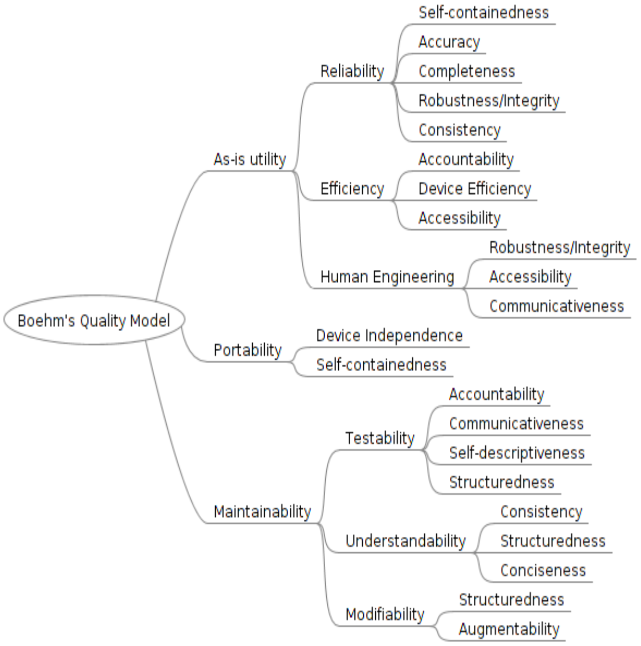

Figure 4 Boehm Software Quality Model

3.4.3 Dromey's Software Quality Model

Dromey quality model proposed a framework for assessing the requirement design and implementation phases. The high-level product properties for the implementation quality model contain correctness, internal, contextual, and descriptive ([9], [10]). Figure 5 is represented as the factors and criteria of the Dromey quality model.

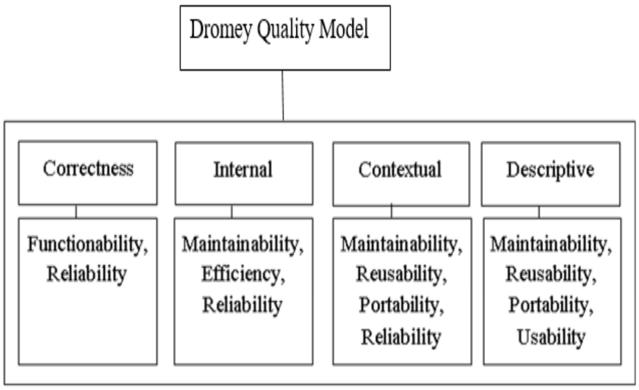
Dromey Quality Model
Correctness
Internal
Contextual
Descriptive
Functionability, Reliability
Maintainability, Efficiency, Reliability
Maintainability, Reusability, Portability, Reliability
Maintainability, Reusability, Portability, Usability

Figure 5 Dromey Software Quality Model

3.4.4 FURPS Software Quality Model

The FURPS model initially presented bay Grady [4], then it is extended by IBM Rational software ([5], [6]) into FURPS+. The "+" sign shows that such requirements as design restrictions, implementation requirements, interface requirements, and physical requirements [5]. Figure 6 is represented as the factors and criteria on FURPS quality model.

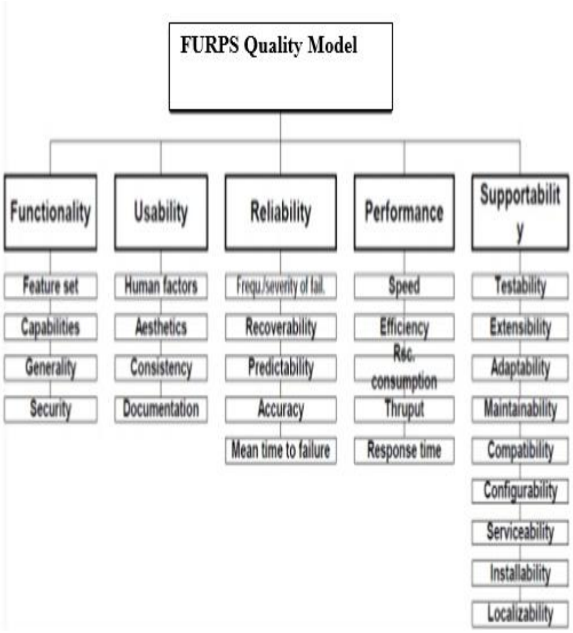

Figure 6 FURPS Software Quality Model

3.4.5 ISO 9126 Software Quality Model

ISO 9000 gives rules for quality assurance [14]. ISO 9000 is the process-oriented approach to quality management [15].it processes designing, documenting, implementing, supporting, monitoring, controlling, and improving[16].ISO 9126 is an international standard for the assessment of software. It has four parts; quality model, external metrics, internal metrics, and quality in use metrics [11].

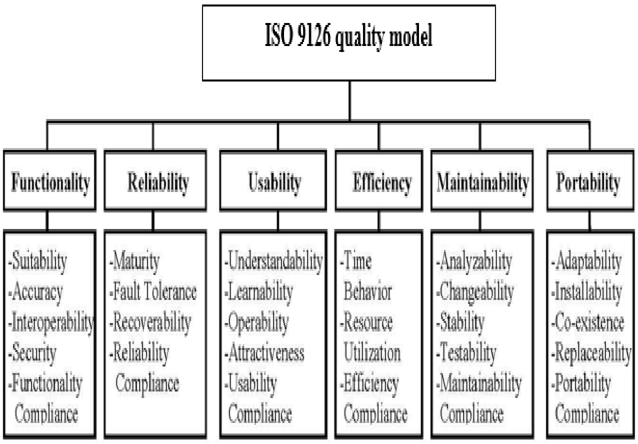

Figure 7 ISO 9126 Software Quality Model

The quality model described above contains some factors in common. But those quality models has some drawbacks like:

⇔ One of well knows the quality model is McCall model which contains 11 software quality factors. Nonetheless, this model did not consider one of the significant attributes of software quality, i.e. functionality. Functionality is what a product, and such as a software application or computing device can do for the user. Therefore functionality is an important factor to evaluate the software so that it would meet the user's needs.

⇔ Boehm's proposed a quality model based on the user's needs but like McCall quality model it did not include the important quality attributes i.e. functionality.

⇔ FURPS quality model has one major drawback, it did not consider the one of the important quality attributes that is portability. Portability is the ability of the software to work in different environments. Furthermore, the client's computing environment might keep changing and accordingly, the software also needs to adapt to a new operating system (computing environment) probability can be significant attributes and therefore cannot be neglected.

Factors	**Quality Model**				
	McCall	**Boehm**	**Dromey**	**FURPS**	**ISO 9126**
Maintainability	✓		✓		✓
Flexibility	✓				
Testability	✓	✓			
Correctness	✓				
Efficiency	✓	✓	✓		✓
Reliability	✓	✓	✓	✓	✓
Integrity	✓				
Usability	✓		✓	✓	✓
Portability	✓	✓	✓		✓
Reusability	✓		✓		
Interoperability	✓				
Human Engineering		✓			
Understandability		✓			
Modifiability		✓			
Functionality			✓	✓	✓
Performance				✓	
Supportability				✓	
17	**11**	**7**	**7**	**5**	**6**

Table 1 Quality factors addressed in the quality models

IV. CONCLUSION

In this research, SQA problems are identified and solutions are suggested to cope with those problems and improve the software quality. Software organizations can only get a respectable position in the Global Market if they concentrate on quality. SQA plays a very important role in the business of Software Company because the only factor which results in getting consistent.

REFERENCE

[1]. Ma Y, He K, Du D, Liu J, Yan Y. A complexity metrics set for the large-scale object-oriented software system, in Proceedings of the Sixth IEEE International Conference on Computer and Information Technology, Washington, DC, USA. 2006; 189-189

[2]. Daniel Galin. Software Quality Assurance from Theory to Implementation, Pearson Addison-Wesley 'the Computer Journal' Vol. 47, No. 6, 728

[3]. McCall JA, Richards PK, Walters GF. Factors in software quality, RADC TR-77-369: 1977. (Rome: Rome Air Development Center)

[4]. Grady, RB. Practical software metrics for project management and process improvement, Prentice Hall; 1992

[5]. Jacobson I, Booch G, Rumbaugh J. The unified software development process, Addison Wesley; 1999.

[6]. Krruchtem P. The rational unified process: an introduction, Addison Wesley; 2000

[7]. Boehm BW, Brown JR, Lipow M. Quantitative evaluation of software quality, In Proceeding of the 2nd International Conference on Software engineering. 1976; 592605.

[8]. Qaim Mehdi Rizvi. Software Quality Assurance: An Introductory Note; 2014

[9]. Dromey RG. A model for software product quality. IEEE Transaction on Software Engineering. 1995; 21:146-162

[10]. Dromey RG. Concerning the Chimera (software quality). IEEE Software. 1996; 1:3343.

[11]. DeepshikhaJamwal.Analysis of Software Quality Models for Organizations, in International Journal of Latest Trends in Computing; 2010

[12]. Stephen H. Kan. Metrics and Models in Software Quality Engineering, Second Edition; .85-126

[13]. Boehm BW, Brow JR, Lipow M, McLeod G, Merritt M. Characteristics of software quality. North Holland Publishing. Amsterdam, the Netherlands; 1978.

[14]. Tomar AB, Thakare VM. A systematic study of software quality models, International Journal of software engineering & application. 2011; 12(4):61-70.

[15]. ISO 9001:2005, Quality management system Fundamentals and vocabulary; 2005

[16]. ISO 9001:2001, Quality management system Requirements; 2001.

[17]. Pressman**,** Scott **(**2005**),** Software Engineering: A Practitioner's Approach (Sixth, International ed.), McGraw-Hill Education Pressman, p. 388

[18]. D. Garvin, "What does product quality really mean?" in MIT Sloan Management Review. vol. 26 Boston: Massachusetts Institute of Technology, 1984, pp. 25 - 43.

[19]. B. Kitchnham and S. Lawerence, "Software Quality: The elusive target,"IEEE Software, vol. 13, pp. 12-21, 1996.