IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 16, Issue 2, Ver. VI (Mar-Apr. 2014), PP 63-72
www.iosrjournals.org

A Survey on Quality Perspective and Software Quality Models

Sony Tripathi
[1](*Assistant Professor ,Computer Science &Engg., N.C.E.T/U.P.T.U, India)*

Abstract: *This paper presents acomprehensive overview ofQuality, definition of quality by different person and alsothis paper present overview on Software Quality Models, McCall's Quality Model, Boehm's Quality Model, Dromey's Quality Model, FURPS Quality Model, ISO 9126 Quality Modeland a comparison among these models are presented.*

Keywords: *Software Quality Models, McCall model, Dorsey's model, FURPS model, ISO 9126 model.*

I. Introduction

To understand the landscape of software quality it is central to answer the so often asked question: *what isQuality*? Once the concept of quality is understood it is easier to understand the different structures of qualityavailable on the market.Software Quality model is a vital to obtained data so that actions can be taken to improve the performance. Such improvement can be measured quality, increased customer satisfaction and decreased cost of quality. Software metrics and quality models play a pivotal role in measurement of software quality. Different researchers have proposed different software quality models to help measure the quality of software products. In our research, we are discussing the different software quality models and compare the software quality models with each other.

II. WHAT IS QUALITY

There are two major camps when discussing the meaning and definition of (software) quality [8]:

1) **Conformance to specification**: Quality that is defined as a matter of products and services whose measurable characteristics satisfy a fixed specification – that is, conformance to an in beforehand defined specification.

2) **Meeting customer needs**: Quality that is identified independent of any measurable characteristics. That is,quality is defined as the products or services capability to meet customer expectations – explicit or not.

2.1 Quality according to Crosby

In the book "Quality is free: the art of making quality certain" [2], Philip B. Crosby writes:
The first erroneous assumption is that quality means goodness, or luxury or shininess. The word "quality" is often used to signify the relative worth of something in such phrases as "good quality", "bad quality" and "quality of life" - which means different things to each and every person. As follows quality must be defined as "conformance to requirements" if we are to manage it. Consequently, the nonconformance detected is the absence of quality, quality problems become nonconformance problems, and quality becomes definable.

2.2 Quality according to Juran

In "Jurans's Quality Control Handbook" [5] Joseph M. Juran provides two meanings to quality:
The word quality has multiple meanings. Two of those meanings dominate the use of the word: 1) Quality consists of those product features which meet the need of customers and thereby provide product satisfaction. 2) Quality consists of freedom from deficiencies. Nevertheless, in a handbook such as this it is most convenient to standardize on a short definition of the word quality as "fitness for use".

2.3 Quality according to Ishikawa

Kaoru Ishikawa writes the following in his book "What is quality control? The Japanese Way" [4]:
We engage in quality control in order to manufacture products with the quality which can satisfy the requirements of consumers. The mere fact of meeting national standards or specifications is not the answer, it is simply insufficient. International standards established by the International Organization for Standardization (ISO) or the International Electro technical Commission (IEC) are not perfect. They contain many shortcomings. Consumers may not be satisfied with a product which meets these standards. We must also keep in mind that consumer requirements change from year to year and even frequently updated standards cannot keep the pace with consumer requirements. How one interprets the term "quality" is important. Narrowly

interpreted, quality means quality of products. Broadly interpreted, quality means quality of product, service, information, processes, people, systems etc.

2.4 Quality according to Shewhart

As referred to by W.E. Deming, "the master", Walter A. Shewhart defines quality in "Economic control ofquality of manufactured product" [6] as follows:
There are two common aspects of quality: One of them has to do with the consideration of the quality of a thing as an objective reality independent of the existence of man. The other has to do with what we think, feel or sense as a result of the objective reality. In other word, there is a subjective side of quality.

2.5 Quality according to Feigenbaum

The name Feigenbaum and the term total quality control are virtually synonymous due to his profoundinfluence on the concept of total quality control (but also due to being the originator of the concept). In "Total
quality control" [3] Armand VallinFeigenbaum explains his perspective on quality through the following text:
Quality is a customer determination, not an engineer's determination, not a marketing determination, nor a general management determination. It is based on upon the customer's actual experience with the product or service, measured against his or her requirements – stated or unstated, conscious or merely sensed, technically operational or entirely subjective – and always representing a moving target in a competitive market. Product and service quality can be defined as: The total composite product and service characteristics of marketing, engineering, manufacture and maintenance though witch the product and service in use will meet the expectations of the customer.

III. QUALITY ASSURANCE

Quality assurance is a planned and systematic set of activities necessary to provide adequate confidence that products and services will conform to specified requirements and meet user needs.Quality assurance is a staff function, responsible for implementing the quality policy defined through the development and continuous improvement of software development processes. Quality assurance is an activity that *establishes* and *evaluates* the processes that produce products.If there is no need for process, there is no role for quality assurance.Quality assurance helps establish processes, sets up measurement programsto evaluate processes, identifies weaknesses inprocesses and improves them. Quality assurance is a management responsibility, frequently performed by a staff function. Quality assurance is concerned with all of the products that will ever be produced by a process. Quality assurance is sometimes called quality control over quality control because it evaluates whether quality control is working. Quality assurance personnel should never perform quality control unless it is to validate quality control.

IV. QUALITY CONTROL

Quality control is the process by which product quality is compared with applicable standards, and the action taken when nonconformance is detected. Quality control is a line function, and the work is done within a process to ensure that the work product conforms to standards and requirements. Quality assurance is a staff function, responsible for implementing the quality policy defined through the development and continuous improvement of software development processes.Quality control activities focus on identifying defects in the actual products produced.These activities begin at the start of the software development process with reviews of requirements, and continue until all application testing is complete. It is possible to have quality control without quality assurance. {For example, a test team may be in place toconduct system testing at the end of development, regardless ofwhether that system is produced using a software developmentmethodology}Quality control relates to a specific product of service. Quality control verifies whether specific attribute(s) are in, or are not in, a specific product or service. Quality control identifies defects for the primary purpose of correcting defects. Quality control is the responsibility of the team/worker. Quality control is concerned with a specific product.

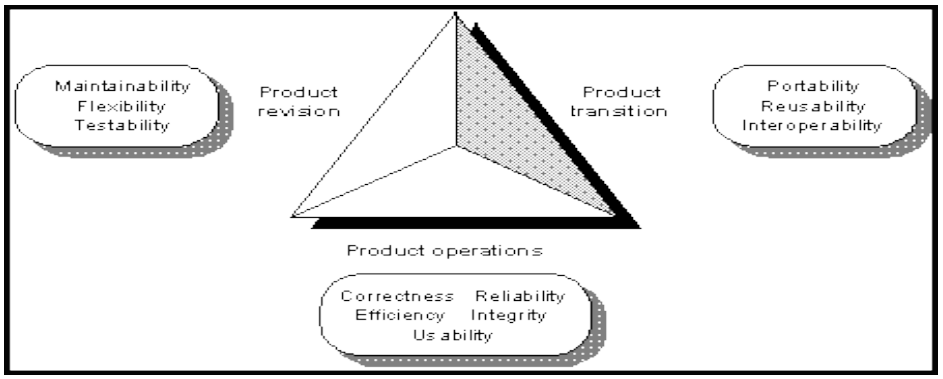
V. SOFTWARE QUALITY MODEL

Software Quality model is a vital to obtained data so that actions can be taken to improve the performance. Such improvement can be measured quality, increased customer satisfaction and decreased cost of quality. Software metrics and quality models play a pivotal role in measurement of software quality. A number of well known qualities models are used to build quality software. Different researchers have proposed different software quality models to help measure the quality of software products.

5.1 McCall software Quality Model

One of the more renown predecessors of today's quality models is the quality model presented by Jim McCall (also known as the General Electric's Model of 1977).McCall's quality model defines and identifies the quality of a software product through addressing three perspectives*:*

- ***Product operation*** is the product's ability to be quickly understood, operated and capable of providing the results required by the user. It covers correctness, reliability, efficiency, integrity and usability criteria.
- ***Product revision*** is the ability to undergo changes, including error correction and system adaptation. It covers maintainability, flexibility and testability criteria.
- ***Product transition*** is the adaptability to new environments, distributed processing together with rapidly changing hardware. It covers portability, reusability and interoperability criteria. Not all the software evolvability sub characteristics are explicitly addressed in this model. Analyzability is not explicitly included as one of the perceived aspects of quality. However, as the model is further detailed into a hierarchy of factors, criteria and metrics, some of the measurable properties and metrics are related to the achievement of analyzability, e.g. simplicity and modularity. Architectural integrity is not covered in the model. Moreover, none of the factors or quality criteria in the model is related to architectural integrity with respect to the understanding and coherence to the architectural decisions. This model is proposed for general application systems, and thus the domain-specific attributes are not explicitly addressed in the scope of the model.

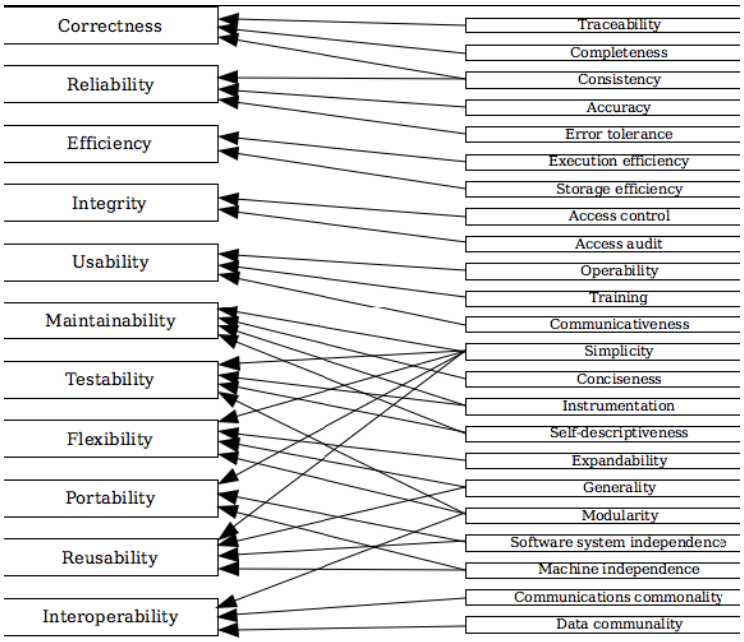


McCall quality perspective

***5.1.1 McCall* Quality Factors**

There are various factors that affect software quality. In this article we will look at these factors proposed by various people and organization.

- ***Correctness:***The extent to which a program satisfies its specifications.
- ***Reliability:***The extent to which a program its specifications.
- ***Efficiency:***The amount of computing sources and code required and code required by a program to perform its function.
- ***Integrity:***Extent to which access to software or data by unauthorized persons can be controlled.
- ***Usability:***The ease with which a user is able to navigate to the system.
- ***Maintainability:***Effort required to fix and test the error.
- ***Flexibility:***Effort required to modify an already operational program.
- ***Testability:***Effort required to test a program so that it performs its intended function.
- ***Portability:***Effort required to port an application from one system to another.
- ***Reusability:***Extent to which a program / sub-program that can be re-used in another applications.
- ***Interoperability:***Extent required to couple one system to another.



McCall quality factors

5.2 Boehm's Software Quality Model

Boehm [1976, 1978] introduced his quality model to automatically and quantitatively evaluate the quality of software. This model attempts to qualitatively define the quality of software by a predefined set of attributes and metrics. Boehm's quality model [7]represents a hierarchical structure of characteristics, each of which contributes to the total quality. The model begins with the software's general utility, i.e. the high level characteristics that represent basic high-level requirements of actual use. The general utility is refined into a set of factors and each factor is composed of several criteria which contribute to it in a structured manner. The factors include: (i) ***portability*;** (ii) ***utility*** which is further refined into reliability, efficiency and human engineering; and (iii) ***maintainability*** which is further refined into testability, understandability and modifiability. Neither in the Boehm quality model is all the software evolvability sub characteristics explicitly addressed. Analyzability is partially addressed through the characteristic *understandability*, which describes that the purpose of the code is clear to the inspector. However, none of the factors or measurable properties describes the capability to analyze the impact at the software architecture level due to a change stimulus. Architectural integrity is not covered in the model.

The high-level characteristics represent basic high-level requirements of actual use to which evaluation ofsoftware quality could be put – the general utility of software. The high-level characteristics address three mainquestions that a buyer of software has:

• ***As-is utility:*** How well (easily, reliably, efficiently) can I use it as-is?
• ***Maintainability:*** How easy is it to understand, modify and retest?
• ***Portability:*** Can I still use it if I change my environment?

The intermediate level characteristic represents Boehm's 7 quality factors that together represent the qualities expected from a software system:

• ***Portability (General utility characteristics):*** Code possesses the characteristic portability to the extent that it canbe operated easily and well on computer configurations other than its current one.

• ***Reliability (As-is utility characteristics):*** Code possesses the characteristic reliability to the extent that it can beexpected to perform its intended functions satisfactorily.

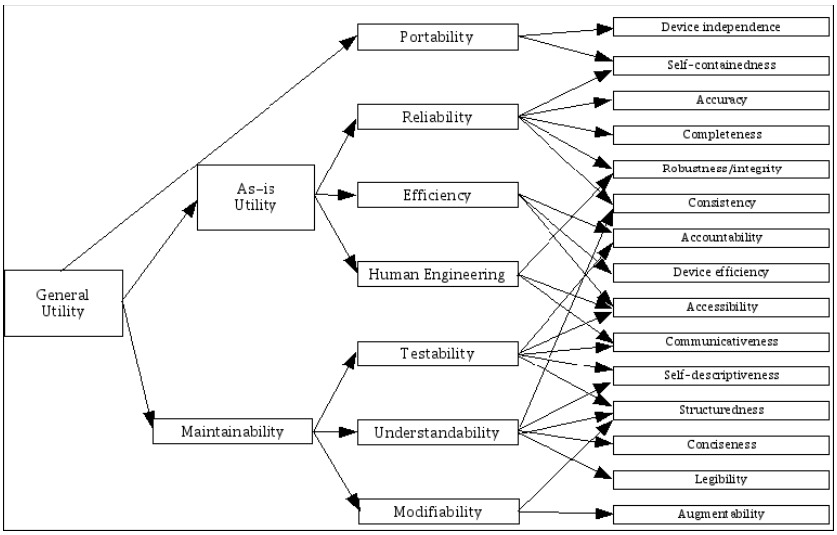
• ***Efficiency (As-is utility characteristics):*** Code possesses the characteristic efficiency to the extent that it fulfillsits purpose without waste of resources.

• ***Usability (As-is utility characteristics, Human Engineering):*** Code possesses the characteristic usability to theextent that it is reliable, efficient and human-engineered.

• ***Testability (Maintainability characteristics):*** Code possesses the characteristic testability to the extent that it facilitates the establishment of verification criteria and supports evaluation of its performance.

• ***Understandability (Maintainability characteristics):*** Code possesses the characteristic understandability to theextent that its purpose is clear to the inspector.

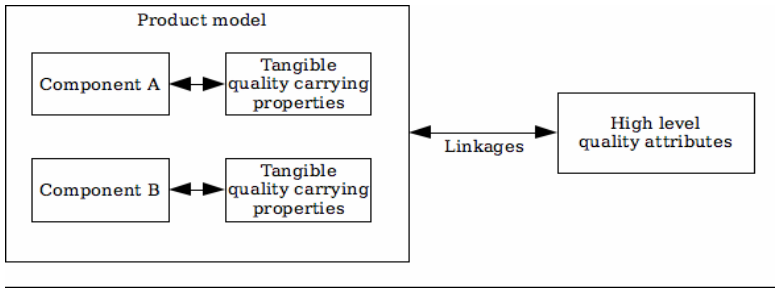
• ***Flexibility (Maintainability characteristics, Modifiability):*** Code possesses the characteristic modifiability to theextent that it facilitates the incorporation of changes, once the nature of the desired change has been determined.(Note the higher level of abstractness of this characteristic as compared with augmentability).



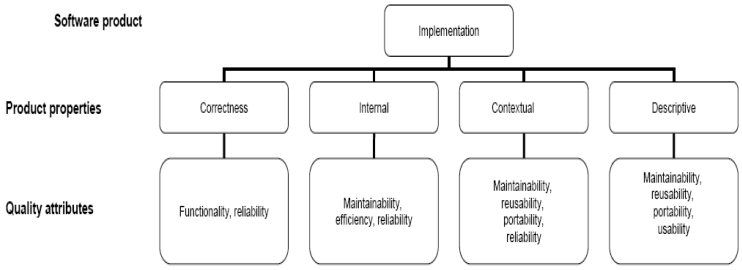
Boehm's quality factors

5.3 Dromey's Software Quality Model

Dromey's proposes [1] a working framework for evaluating Requirement determination, design and implementation phases. The framework consists of three models, i.e. *Requirement quality model*, *Design quality model* and *Implementation quality model*. The high-level product properties for the implementation quality model include: (i) *Correctness* evaluates if some basic principles are violated, with functionality and reliability as software quality attributes; (ii) *Internal* measures how well a component has been deployed according to its intended use, with maintainability, efficiency and reliability as software quality attributes; (iii) *Contextual* deals with the external influences on the use of a component, with software quality attributes in maintainability, reusability, portability and reliability; (iv) *Descriptive* measures the descriptiveness of a component, with software quality attributes in maintainability, reusability, portability and usability. In this model, characteristics with regard to process maturity and reusability are more explicit in comparison with the other quality models. However, not all the evolvability sub characteristics are explicitly addressed in this model. Analyzability is only partially covered within the *contextual* and *descriptive* product properties at individual component level, though none of these product properties describes the capability to analyze the impact at the software architecture level due to a change stimulus. Architectural integrity is not fully addressed despite the *design quality model* takes into account explicitly the early stages (analysis and design) of the development process. The focus of the *design quality model* is that a design must accurately satisfy the requirements, and be *understandable*, *adaptable* in terms of supporting changes and developed using a mature process. However, it is not sufficient for capturing architectural design decisions. Extensibility is not addressed as an explicit characteristic to represent future growths. Testability is implicitly embedded in the *internal* product property. Domain-specific attributes are not addressed. Moreover, one disadvantage of the Dromey model is associated with reliability and maintainability, as it is not feasible to judge them before the software system is actually operational inthe production area.



Dromey's software quality model



Principle of Dromey's model

5.4 FURPS Software Quality Model

The characteristics that are taken into consideration in FURPS model [10] are:

- ***Functionality***includes feature sets, capabilities and security;
- ***Usability***includes human factors, consistency in the user interface, online and context-sensitive help, wizards, user documentation, and training materials;
- ***Reliability***includes frequency and severity of failure, recoverability, predictability, accuracy, and mean time between failure (MTBF);
- ***Performance***prescribes conditions on functional requirements such as speed, efficiency, availability, accuracy, throughput, response time, recovery time, and resource usage;
- ***Supportability***includes testability, extensibility, adaptability, maintainability, compatibility, Configurability, serviceability, installability, and localizability / internationalization. Architectural integrity is not covered in the model. None of the characteristics or sub characteristics in the model is related to architectural integrity with respect to the understanding and coherence to the architectural decisions. Moreover, one disadvantage of this model is that it fails to take account of the software portability. Domain-specific attributes are not addressed either in the model.

5.5 ISO 9126 Software Quality Model

ISO 9126 is an international standard for the evolution of software. The standard is divided into four parts which address respectively the following subjects: Quality model, External metrics, internal metrics and quality in use metrics. ISO 9126 Part-1 is an extension of previous work done by McCall (1977), Boehm (1978), FURPS etc. ISO 9126 specifies and evaluates the quality of a software product in terms of internal and external software qualities and their connection to attributes. The model follows the factor-criteria-metric model and categorizes software quality attributes into six independent high-level quality characteristics: functionality, reliability, usability, efficiency, maintainability and portability. Each of these is broken down into secondary quality attributes, e.g. maintainability is refined into analyzability, changeability, stability, testability and compliance to standards, conventions or regulations. One may also argue if the enhancement-with-new features type of change is embedded within the types of modifications defined in the quality model, i.e. corrections, improvements or adaptations of the software to changes in environment, requirements and functional specifications.

The new quality model defined in ISO/IEC 9126-1 recognizes three aspects of software quality and defines them as follows: (the full definition is given as it is pertinent to the discussion that ensues).

5.5.1 External quality

External quality is the totality of characteristics of the software product from an external view. It is the quality when the software is executed, which is typically measured and evaluated while testing in a simulated environment with simulated data using external metrics. During testing, most faults should be discovered and eliminated. However, some faults may still remain after testing. As it is difficult to correct the software architecture or other fundamental design aspects of the software, the fundamental design remains unchanged throughout the testing. (ISO/IEC, 2001a)

5.5.2 Internal Quality

Internal quality is the totality of characteristics of the software product from an internal view. Internal quality is measured and evaluated against the Internal Quality requirements. Details of software product quality can be improved during code implementation, reviewing and testing, but the fundamental nature of the software product quality represented by the Internal Quality remains unchanged unless redesigned. (ISO/IEC, 2001a)

5.5.3 Quality in Use

Quality in use is the user's view of the quality of the software product when it is used in a specific environment and a specific context of use. It measures the extent to which users can achieve their goals in a particular environment, rather than measuring the properties of the software itself. (ISO/IEC, 2001a)

5.5.4 ISO 9126 Quality Factors

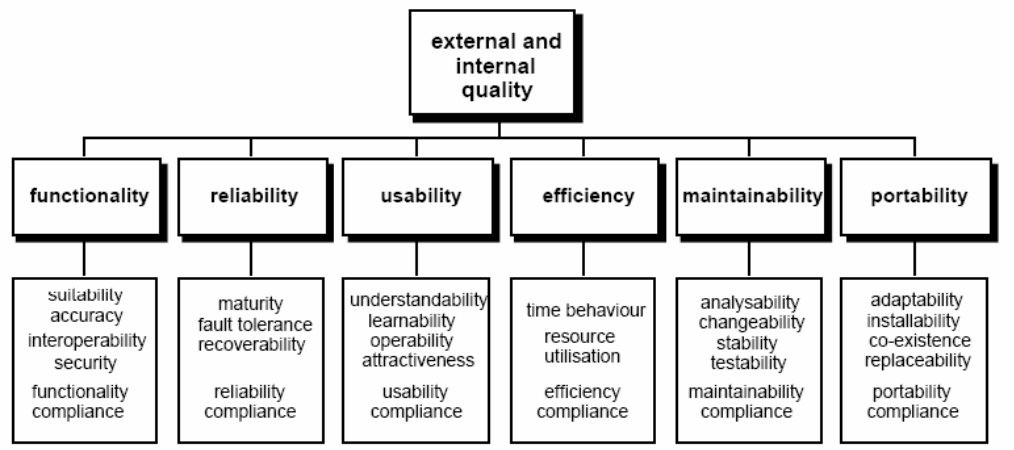
The ISO 9126[11] standard identifies six key quality attributes.

- ***Functionality*** **-** degree to which software satisfies stated needs.
- ***Reliability*** **-** the amount of time the software is up and running.
- ***Usability*** **-** the degree to which software is easy to use.
- ***Efficiency*** **-** the degree to which software makes an optimum utilization of the resources.
- ***Maintainability*** **-** the ease with which the software can be modified.
- ***Portability*** **-** the ease with which software can be migrated from one environment to the other.

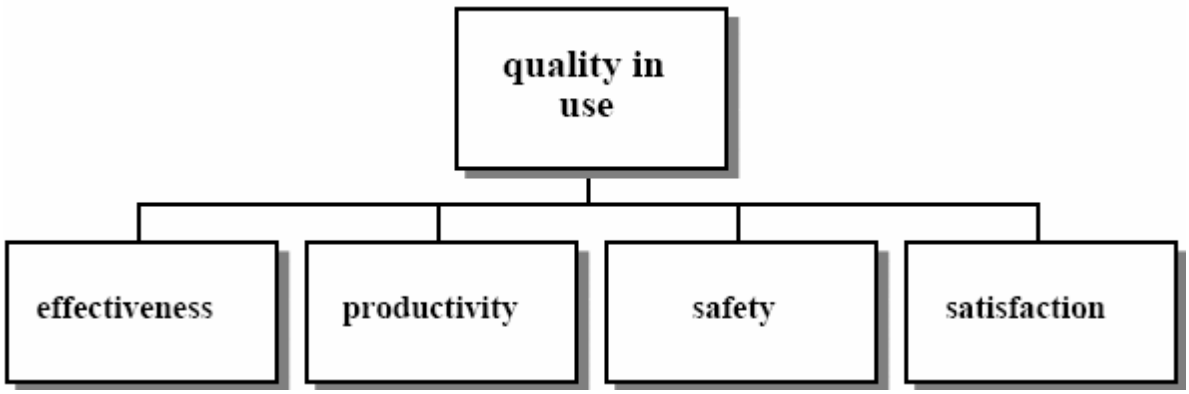
Each quality factors and its corresponding sub-factors are defined as follows:

- ***Functionality:*** A set of attributes that relate to the existence of a set of functions and their specified properties. The functions are those that satisfy stated or implied needs.
- ***Suitability:*** Attribute of software that relates to the presence and appropriateness of a set of functions for specified tasks.
- ***Accuracy:*** Attributes of software that bare on the provision of right or agreed results or effects.
- ***Security:*** Attributes of software that relate to its ability to prevent unauthorized access, whether accidental or deliberate, to programs and data.
- ***Interoperability:*** Attributes of software that relate to its ability to interact with specified systems.
- ***Compliance:*** Attributes of software that make the software adhere to application related standards or Conventions or regulations in laws and similar prescriptions.
- ***Reliability:*** A set of attributes that relate to the capability of software to maintain its level of performance under stated conditions for a stated period of time.
- ***Maturity:*** Attributes of software that relate to the frequency of failure by faults in the software.
- ***Fault tolerance:*** Attributes of software that relate to its ability to maintain a specified level of performance in cases of software faults or of infringement of its specified interface.
- ***Recoverability:*** Attributes of software that relate to the capability to re-establish its level of performance and recover the data directly affected in case of a failure and on the time and effort needed for it.
- ***Usability:*** A set of attributes that relate to the effort needed for use, and on the individual assessment of such use, by a stated or implied set of users.
- ***Understandability:*** Attributes of software that relate to the users' effort for recognizing the logical concept and its applicability.
- ***Learnability:*** Attributes of software that relate to the users' effort for learning its application (for example, operation control, input, output).
- ***Operability:*** Attributes of software that relate to the users' effort for operation and operation control.
- ***Compliance:*** Attributes of software that make the software adhere to application related standards or conventions or regulations in laws and similar prescriptions.
- ***Efficiency:*** A set of attributes that relate to the relationship between the level of performance of the software and the amount of resources used, under stated conditions.

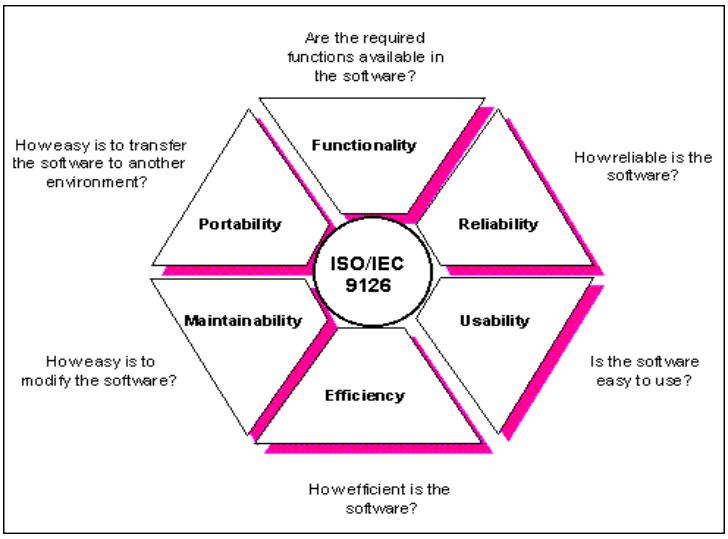
- ***Time behavior:*** Attributes of software that relate to response and processing times and on throughput rates in performing its function.
- ***Resource behavior:*** Attributes of software that relate to the amount of resources used and the duration of such use in performing its function.
- ***Maintainability:*** A set of attributes that relate to the effort needed to make specified modifications.
  Analyzability: Attributes of software that relate to the effort needed for diagnosis of deficiencies or causes of failures, or for identification of parts to be modified.
- ***Changeability:*** Attributes of software that relate to the effort needed for modification, fault removal or for environmental change.
- ***Stability:*** Attributes of software that relate to the risk of unexpected effect of modifications.
- ***Testability:*** Attributes of software that relate to the effort needed for validating the modified software.
- ***Portability:*** A set of attributes that relate to the ability of software to be transferred from one environment toanother.
- ***Adaptability:*** Attributes of software that relate to on the opportunity for its adaptation to different specifiedenvironments without applying other actions or means than those provided for this purpose for the softwareconsidered.
- ***Installability:*** Attributes of software that relate to the effort needed to install the software in a specifiedenvironment.
- ***Conformance:*** Attributes of software that make the software adhere to standards or conventions relating toportability.
- ***Replaceability:*** Attributes of software that relate to the opportunity and effort of using it in the place ofspecified other software in the environment of that software.



3-layer model for internal and external quality. Adapted from (ISO/IEC, 2001a)



quality in use factors



ISO 9126 quality model

VI. Comparison

Criteria/goals	***McCall, 1977***	***Boehm, 1978***	***ISO 9126, 1993***	**Dromey's Model**	**FURPS Model**
Correctness	*	*	maintainability		
Reliability	*	*	*	*	*
Integrity	*	*			
Usability	*	*	*		
Efficiency	*	*	*	*	*
Maintainability	*	*	*	*	*
Testability	*		maintainability		
Interoperability	*				
Flexibility	*	*			*
Reusability	*	*		*	
Portability	*	*	*	*	
Clarity		*			
Modifiability		*	maintainability		
Documentation		*			
Resilience		*			
Understandability		*		*	*
Validity		*	maintainability		
Functionality			*	*	*
Generality		*			
Economy		*			
Performance					*
Supportability					*

VII. Conclusion

In this paper finally it is concluded that Quality is a prime concern for any organization. A quality product must fulfill the customer need and meet the specification. Definition of quality by different person present different perspective of quality. Crosby is a clear "conformance to specification" quality definition adherer.. Crosby also emphasizes that it is important to clearly define quality to be able to measure and manage the concept.JuranInstead defines quality as "fitness for use" – which indicates references to requirements and products characteristics. As follows Juran's definition could be interpreted as a "conformance to specification" definition more than a "meeting customer needs" definition. Ishikawa's perspective on quality is a "meeting customer needs" definition as he strongly couples the level of quality to every changing customer expectations. He further means that quality is a dynamic concept as the needs, the requirements and the expectations of a customer continuously change. Ishikawa also includes that price as an attribute on quality – that is, an overprized product can neither gain customer satisfaction and as follows not high quality. Although Shewhart's definition of quality is from 1920s, it is still considered by many to be the best and most superior. Shewhart talks about both an objective and subjective side of quality which nicely fits into both" conformance to specification" and "meeting customer needs" definitions.Feigenbaum's definition of quality is unmistakable a "meeting customer needs" definition of quality. In fact, he goes very wide in his quality definition by emphasizing the importance of satisfying the customer in both actual and expected needs. Feigenbaum essentially points out that quality must be defined in terms of customer satisfaction, It is clear that Feigenbaum's definition of quality not only encompasses the management of product and services but also of the customer and the customer's expectations.In this research paper, we have also studied different types of software quality models like McCall, ISO 9126, Dromey's etc. From the 17 characteristics, only one characteristic is common to all quality models, that is, the, reliability. Also, there are only three characteristics (i.e., efficiency, usability and, portability) which are belonging to four quality models. Two characteristic is common only to three quality models, that is, the, functionality and maintainability characteristics. Two characteristic belong to two quality models, that is, the, testability and reusability characteristics. And, nine characteristics (i.e. flexibility, correctness, integrity and interoperability in McCall's quality model; human engineering, understandability and, modifiabilityin Boehm's quality model; performance and, supportability in FURPS quality model) are defined in only one quality model. Furthermore, it can be noted that the, testability, interoperability and, understandability are used as factors/attributes/characteristics in some quality models. However, in ISO 9126-1, these factors/attributes/characteristics are defined as sub characteristics. More specifically, the testability is belonging to the maintainabilitycharacteristic, the understandability is belonging to the usability characteristic, and the interoperability is belonging to the functionality characteristic. From our point of view, the ISO 9126-1 quality model is the most useful one since it has been built based on an international consensus and agreement from all the country members of the ISO organization.

REFERENCES

[1] R. G. Dromey, "A *model for software product quality".* IEEE Transactions on Software Engineering, 21(2):146– 162, 1995.
[2] Crosby, P. B., *Quality is free : the art of making quality certain*, New York : McGraw-Hill, 1979.
[3] Feigenbaum, A. V., *Total quality control*, McGraw-Hill, 1983.
[4] Ishikawa, K., *What is total quality control? : the Japanese way*, Prentice-Hall, 1985.
[5] Juran, J. M., *Juran's Quality Control Handbook*, McGraw-Hill, 1988.
[6] Shewhart, W. A., *Economic control of quality of manufactured product*, Van Nostrand, 1931.
[7] Boehm, B. W., Brown, J. R., Kaspar, H., Lipow, M., McLeod, G., and Merritt, M., *Characteristics ofSoftware Quality*, North Holland, 1978.
[8] Hoyer, R. W. and Hoyer, B. B. Y., "What is quality?",*Quality Progress*, no. 7, pp. 52-62, 2001.
[9] http://en.wikipedia.org/wiki/ISO/IEC_9126
[10] http://en.wikipedia.org/wiki/FURPS
[11] file:///J:/software%20model/Software%20quality%20metrics%20and%20model%20_%20Fred%20Beringer.htm
[12] http://www.sei.cmu.edu/publications;/articles/qualtiy-profile/index.html