[1] Assistant Professor in Computer Science, Govt. College Barwala (Pkl)
[2] Assistant Professor in Computer Science, S D College, Ambala Cantt

Abstract
Quality of a software product has become an important task for many companies. Software quality is having multi-dimensional content which may be distinguished and measured easily. It is a made up of many characteristics. Because of that, quality is captured in a form of many models which shows the characteristics and their relationships. These models are very useful and that shows the overall quality of software products. Different companies use different quality models based upon their requirements. The purpose of this Paper is to describe the main models with their strong and weak points. We have also made a comparative study of various software quality models.

KEY WORDS
Software Quality, software Quality Models.

1. INTRODUCTION
Software quality is conformity to expressly explicit practical and performance necessities, expressly documented development standards, and implicit characteristics that are expected of all professionally developed code."

According to the IEEE Standard Glossary of Software Engineering Terminology [1,5,7], the quality of software products is defined as 1) the degree to which a system, component or process meets specified requirements and 2) the degree to which a system, component or process meets the needs or expectations of a user.
Quality is the combination of features and characteristics of a product or a service which has ability to satisfy the given requirements(ANSI/ASQC A3/1978). Software quality has been categorized into two parts by Deutsch et al. [3] as software procedure quality and software product quality.All Software related elements like technology, tools, people, organization and equipment were used in software procedure quality. However, software product quality consists many aspects like document clarity and integrity, design trace-ability, program reliability and test integrity. A quality model is usually defined as a set of characteristics and the relationships between them which actually provides the basis for specifying all the requirements of quality and evaluating it [4].

2 Software Quality Models

2.1 THE ISO 9126 STANDARD QUALITY MODEL
The ISO 9126 standard was created by the ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission). This model has the subsequent subjects: quality model, external metrics, internal metrics, and quality in use metrics. The aim of the Quality Model is to define the nonfunctional (quality) characteristics of software products.

The ISO 9126-1 software quality model identifies following main quality features, namely:

1 Functionality – The capability of the software product is to produce functions which meet stated and implied desires when software is employed under specified conditions.

2 Reliability- The capability of the software product to keep a specific level of performance once used under specified conditions.

3 Portability - The capability of the software product to be transferred from one environment to another.

4 Efficiency- The capability of the software product to produce desired performance, relative to the quantity of resources used, under specified conditions.

5 Usability - The capability of the software product to be understood, learned, used and engaging to the user, when used under specified conditions.

6 Maintainability - The capability of the software product to be modified. Modifications may include corrections, enhancements or adaptations of the software to changes in the environment, and in needs and functional specifications.

2.2 Mc Call Software Quality Model.

McCall software quality model is developed by Jim McCall in 1977. This is used to maintain harmony between the users and the developers. Successful software is produced that meets all the user needs in accordance with the developer's point of view McCall shows three main perspectives for characterizing the quality factors of a software product.

These perspectives are:-

- Product revision (includes the ability to change).
- Product transition (includes adaptability to new environments).
- Product operations (includes basic operational characteristics).

Product revision

The product revision shows quality factors that influence the ability to change the software product, these factors are:-

- Maintainability, the ability to find and fix a defect.
- Flexibility, the ability to make changes required as dictated by the business.
- Testability, the ability to validate the software requirements.

Product transition

The product transition shows quality factors that influence the ability to adapt the software to new environments:-

- Portability, the ability to transfer the software from one environment to another.
- Reusability, the ease of using existing software components in a different context.
- Interoperability, the extent, or ease, to which software components work together.

Product operations

the product operation shows quality factors that influence the extent to which the software fulfils its specification:-

- Correctness, the functionality meets the specification.
- Reliability, the extent to which the system fails.
- Efficiency, system resources (including cpu, disk, memory, network) usage.
- Protection from unauthorized access.
- Usability.

2.3 Boehm Model

Bohem quality models is the quality model presented by Barry W. Boehm. In the stage,approved changes are determined by applying particular strategies and cost-benefit evaluations to a set of proposed changes. The approved changes are accompanied by their own budgets, which will largely determine the extent and type of resources used.

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 can be moved easily and well on the other computer configurations.

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

• Efficiency (As-is utility characteristics): Code possesses the characteristic efficiency to the extent that it fulfills its purpose without any wastage of resources.

• Usability (As-is utility characteristics, Human Engineering): Code possesses the characteristic usability to the extent 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 analysis of its performance.

• Understandability (Maintainability characteristics): Code possesses the characteristic understandability to the

extent that its purpose is evident to the inspector.
• Flexibility (Maintainability characteristics, Modifiability): Code possesses the characteristic modifiability to the extent that it facilitates the incorporation of changes, once the nature of the desired change has been determined.

2.4 The FURPS Model

Grady B. R. and Hewlett introduced the idea of FURPS model [2] which characteristics into two different necessities such as Functional Needs(F) which is outlined by expected input & output and Non Functional Needs in which U stands for Usability (includes human factors, aesthetic, documentation of user and material of training), R stands for Reliability (includes frequency and severity of failure, recovery to failure, time among failure), P stands for Performance (includes useful requirements) and the S stands for Supportability (includes backup, requisite of design, implementation, interface).

2.5 DROMEY'S QUALITY MODEL

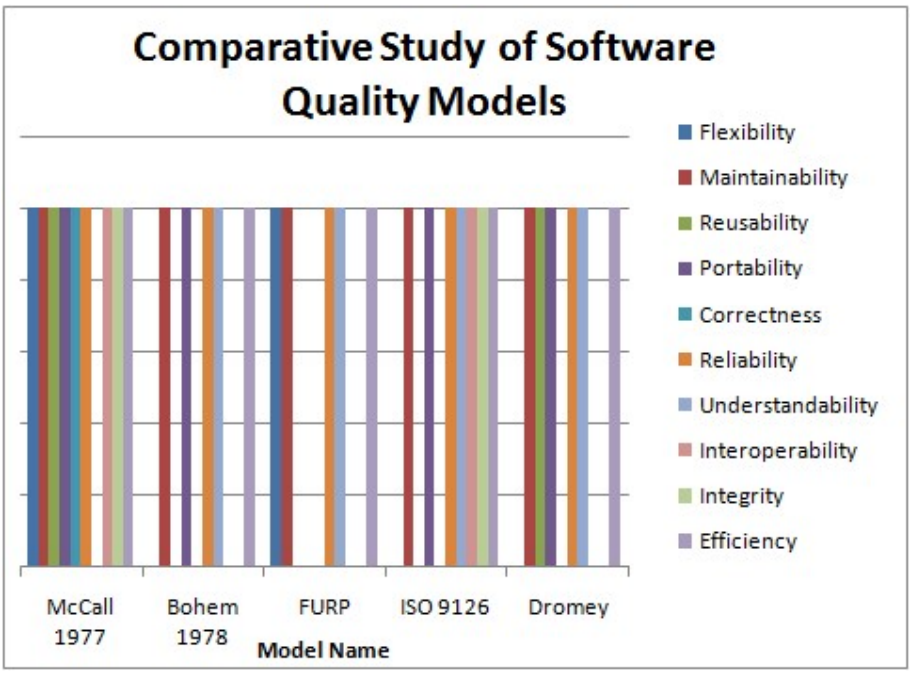
Dromey G. R. [6] quality model is based on analysis criteria. In other words, it aims at searching out the quality of the product when each software product has different quality then the other.Dromey offers the subsequent examples of what he means by software components for each of the various models:
• Variables, functions, statements, etc. can be considered Components of the Implementation model;
• A requirement can be considered a component of the requirements model;
• A module can be considered a component of the design model;
According to Dromey (1995), these components all possess intrinsic properties that can be classified into four categories:
• Correctness: find out if some basic principles are violated.
• Internal: Estimate how well a component has been deployed according to its intended use.
• Contextual: Deals with the external influences by and on the use of a component.
• Descriptive: Estimate the descriptiveness of a component (for example, does it have a meaningful name.

3. COMPARISON OF ABOVE MODELS

Table 1 shows a comparison of essential models relating to the main characteristics. Flexibility is related to the manufacturing process [8] and is taken into account as a side of maintainability. From the table we conclude that reliability is a common characteristic to all the models. The reason is the close relation with the opinion of users and the success of any software product will depend on the fact of being used or not.

Factors	**McCall 1977**	**Boehm 1978**	**FURP**	**ISO 9126**	**Dromey**
Flexibility	Y		Y		
Maintainability	Y	Y	Y	Y	Y
Reusability	Y				Y
Portability	Y	Y		Y	Y
Correctness	Y				
Reliability	Y	Y	Y	Y	Y
Understandability		Y	Y	Y	Y
Interoperability	Y			Y	
Integrity	Y			Y	
Efficiency	Y	Y	Y	Y	Y

Table 1 Comparison between Different Quality Models



4. Conclusions

The overall conclusion is that there are very general models for assessing software quality and hence they are difficult to apply to specific cases. Also there exist tailored quality models whose range is in small domain, using as starting model the ISO 9126. Tailored Quality Models originated from the Basic Models basic consider a specific domain and selects the features and sub features to consider. The model created in this way is for a specific, particular product or from the point of view of a user domain. Therefore have limitations. Software quality engineering needs a quality model that is usable throughout the software lifecycle and that it embraces all the perspectives of quality model suitable for such a purpose, through the comparative evaluation of existing quality models and their respective support Quality engineering.

In all the models studied none has incorporated the aspect of communication as one of the quality factors. At the present time, there is a need for quality components for communications at all levels and especially in complex systems, where it becomes a critical factor because of the internet.

Finally, we note that in most of the studied models the factors and criteria have the same value which is relative because it depends on the application domain. For example aspects of transferability can be crucial in software that is installed on different machines.

References

[1] IEEE. (1990). IEEE Std 610.12-1990 (1990)- "IEEE Standard Glossary of Software Engineering Terminology,"

http://web.ecs.baylor.edu/faculty/grabow/Fall2013/csi3374/secure/Standards/IEEE610.12.pdf

[2] Ghezzi, Jazayeri, C. M. and Mandrioli, D.1991. Fundamental of software Engineering, Prentice–Hall, NJ, USA.

[3] Chang, C Wu C., Lin, H. 2008. Integrating Fuzzy Theory and Hierarchy Concepts to Evaluate Software Quality, Software Quality Control, 16(2), pp. 263-267.

[4] ISO 9126. 2000E. Standard ISO/IEC , Information technology- Software product quality – Part1: Quality Model, ISO/IEC FDIS 9126-1: 2000(E)

[5] IEEE. (1998). Standard for Software Maintenance, Software Engineering Standards Subcommittee of the IEEE Computer Society.

[6] Deutsch, M.S. and Wills, R. R. 1988. Software Quality Engineering, A Total Technical and Management Approach, Prentice-Hall Inc.

[7] Samadhiya Durgesh & Wang Su-Hua & Chen Dengjie.(2010), "Quality Models: Role and Value in Software Engineering," 2nd International Conference on Software Technology and

Engineering(ICSTE). Pp 320-324.

[8] 27 Ghayathri J & Priya E. M. (2013) "Software Quality Models: A Comparative Study," International Journal of Advanced Research in Computer Science and Electronics Engineering

(IJARCSEE) ,Volume 2, Issue 1, pp 42-51.