

A QUALITY MODEL FOR ADAPTABILITY

Debdatta Chatterjee
M.Tech Software Engineering
KIIT University
Bhubaneswar, Orissa
debdatta9874@gmail.com

Dr. R.K. Malu
Professor of
KIIT University
Bhubaneswar, Orissa
malubbsr@yahoo.com

Abstract-Software quality is one of the major components of software development industry. Every user wants to have software enriched with the best quality factors. If a user is not satisfied with the quality of the product it will definitely be not considered as a quality product. To ensure the quality of the product several quality model has been introduced. One of the most renowned software quality models McCall quality model is briefed in this paper. A short description of 11 quality factors is given in our research work. To bridge up the gap between the developer and user we have suggested one quality factor adaptability and a quality model is also approached.

Keywords- software quality, quality factors, McCall quality model, adaptability

I. INTRODUCTION

Software Quality is one of the main ingredients in Software Engineering. The product which full fills the needs of a user should also be verifiable and acceptable in range of a software quality. Different researchers had described the software quality in different type of philosophical manners. We can say that software quality in the bridge which helps to build up the gap between the user and developer. There is not any specific definition of software quality; it changes over time and users user. Quality is need to be planned and controlled throughout the entire project life cycle. To increase the quality of a product different quality models is introduced, McCall initialised McCall's Quality Model in the year 1977. In the very next year(1978) Boehm's Quality Model is introduced. International Organization for Standardisation adopted ISO-9126 in the year 1991 to evaluate software quality. ISO 9001, which deals with the quality assurance of the process that are used to develop products, is complimented by ISO-9126[1, 2]. ISO-9126 was based on McCall's Quality Model and Boehm's Quality Model. In the year 1995 R. Geoff Dromey presented another quality model which in known as Dromey's Quality Model. The youngest quality model is named as FURPS which is presented by Robert Grady [5].

The main aim of those quality models is to increase the quality of products/system. It is very necessary to measure whenever there comes question of improvement. But software quality being highly subjective is very difficult to measure. The most effective method of software measurement is users' feedback.

In the below sections we will have a brief a description about the McCall Quality Model and have our proposed work.

II. MCCALL QUALITY MODEL

McCall Quality Model is the first quality model which was introduced. It is proposed by Jim McCall in the year 1977. He tried to build up the gap between the user and the developer by approaching quality factors from the both perspective. This quality model mainly focused towards system development [3]. McCall defined 11 quality factors and 41 metrics in his quality model. McCall divided the 11 quality factors into three aspects. Those are:

A. PRODUCT REVISION

Product revision helps to increase the ability to change the product. It has a big role to manipulate the changing software product. The quality factors are:

- *Maintainability*: Product that are easily maintainable and user has to spent money in maintenance.
- *Flexibility*: Product should be flexible enough to change with environment. Flexibility has three dimensions, those are changeable, time and cost.
- *Testability*: It provides the intermediate results of computations which are predefined to assist testing.

B. PRODUCT OPERATION

The main function of product operation is to quickly understand, operate and provide the result. It gives us an overview of the whole product operation. The quality factors are:

- *Correctness*: It helps to free the product from error or fault, and confirms it standard. So it can be said the degree to which a system is free from defects.
- *Reliability*: The product should be reliable enough to use. We can get the reliability of the product by simply making the ratio of number of production re- runs and production runs.
- *Efficiency*: Efficient enough to remove defects.
- *Integrity*: It provides an unique access, and protect from illegal access
- *Usability*: It gives us the calmness and comfort of software.

C. PRODUCT TRANSITION

The main purpose of product transition is to increase the capacity of adaption of a software in any environment. The quality factors are:

- *Portability*: It increases the capability to transmit the software from one to another platform.
- *Reusability*: The ability to reuse the product several time.
- *Interoperability*: The major software components work together.

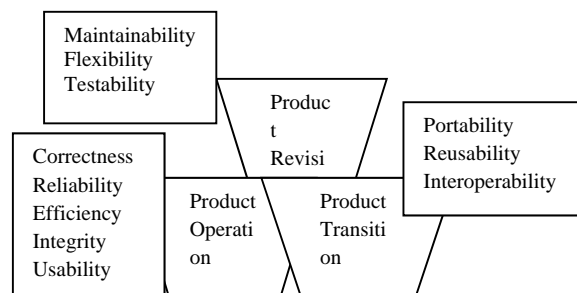


Figure 1: McCall Quality Model

III. PROPOSED WORK

In McCall Quality Model [4] he has introduced 11 quality factors which enrich the quality model. In our proposed work we have tried to introduce software adaptability quality factor. The quality factor adaptability and flexibility both are parallel

Quality factors in software quality, but flexibility is for short time period and adaptability is for long time period.

According to ISO/IEC 9126-1[6] software adaptability has been defined as "the capability of the software product to be adapted for different specified environments without applying actions or means other than those provided for this purpose for the software considered. "

Adaptability is the ability of software to adapt any other platform/ environment that it belongs. It helps the software to be platform independent. It brings more versatility in the software and users get the usability to use the software in any environment. Adaptability not only makes the software platform independent but also it increases the capability to adapt any changes and can work as users requirement. It is one of the most important non functional requirements.

The main two properties of adaptability are:

- i. Adaptability makes software evolvable.
- ii. Adaptability makes software more scalable.

As per the Factors, Criteria Metrics, software adaptability can be decomposed into below quality criteria.

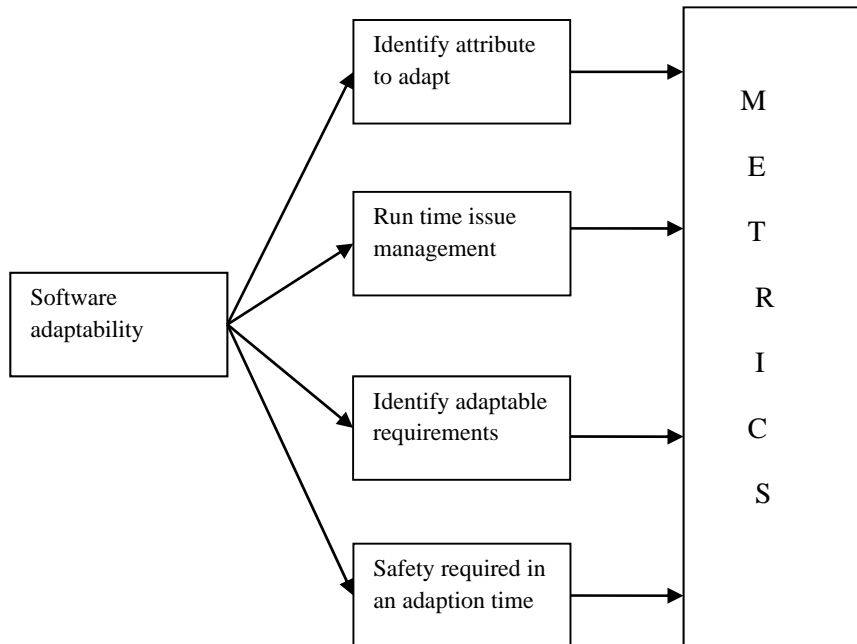


Figure 2: Factors Criteria Metrics for Software Adaptability

The concept of adaptability increases the trust worthiness of understand ability and reuse ability. But the major problem of adaptability is it lead to large program that mostly contains chunk of code.

The main metrics attribute of adaptability are:

- i. Simple and computable
- ii. Empirically and intuitively persuasive.
- iii. Consistent and objectives.
- iv. Programming language independent
- v. An effective mechanism for quality feedback.

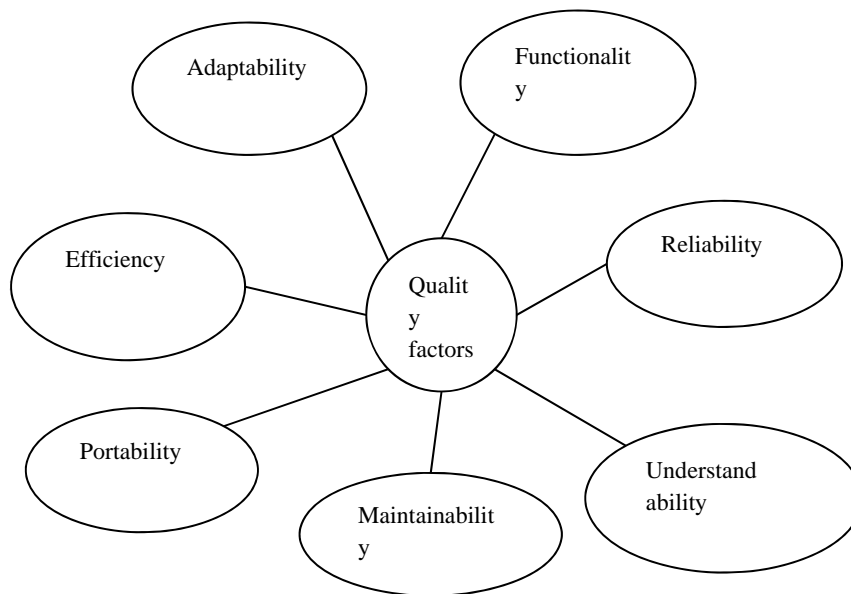
New Approach

Figure 3: Software Quality Factors

IV. CONCLUSION AND FUTURE WORK

Software quality factors are one of the main constraints of software development procedure. If a user has to pay plenty of money after the delivery of software it won't be considered as a worthy software. Many researchers have identified lots of quality models. But we need to have specific quality factors rather than a junk of software quality factors.

Software adaptability makes the software more platform flexible and if any changes are made in the software it increases the adaptation power also to absorb the changes. It makes the software simple and computable. Programmers gain the independency of using any programming language. It is an effective mechanism for quality feedback.

The proposed model is introduced to increase the software quality so that the gap between the developer and the user can be bridged up. The software should be enriched with effective quality factors.

REFERENCES

- [1] J. Sligo, M. Diaz, "How Software Process Improvement Helped Motorola," IEEE Software, Volume.14, no. 5 September-October, 1997, pp. 75-81.
- [2] W. Frakes, C. Fox "The Quality Approach: Is It Delivering?," Comm. ACM, Vol. 40, 1997, pp. 25-29.
- [3] P. K. Richards, J. A. McCall, and G. F. Walters, "Factors in Software Quality", Nat'l Tech. Information Service, no. 1977 Vol. 1, 2 and 3.
- [4] Pfleeger, S.L., Kitchenham, B. "Software quality: the elusive target [special issues section]", IEEE Software, no. 1, 1996, pp. 12-21.
- [5] R. B., Grady, "Practical software metrics for project management and process improvement", Prentice Hall, 1992.
- [6] <http://www.essi.upc.edu/~qms/DesCOTS/CQM/ExtISO.html>