Proc. of the Intl. Conf. on Recent Trends In Computing and Communication Engineering -- RTCCE 2013

ISBN: 978-981-07-6184-4

Comparison Of Various Software Quality Models

[Kavita Sharma, Kumud Sharma]

***Abstract*—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. In our research, we are discussing the different software quality models and compare the software quality models with each other.**

***Keywords*—Software Quality Models, McCall model, FURPS model, ISO 9126 model.**

Objectives—To begin with there are some common objectives:-

- The presence, or absence, of these attributes can be measured objectively.
- The degree to which each of these attributes is present reflects the overall quality of the software product.
- These attributes facilitate continuous improvement, allowing cause and effect analysis which maps to these attributes, or measure of the attribute.

I. Introduction

"Quality comprises all characteristics and significant features of a product or an activity which relate to the satisfying of given requirements". Software is critical in providing a competitive edge to many organizations, and is progressively becoming a key component of business systems, products and services. The quality of software products is now considered to be an essential element in business success.

Kavita Sharma Asstt.Professor in Computer Science
D.A.V Centenary College Faridabad
India

Kumud Sharma Asstt. Professor in Computer Science
D.A.V Centenary College Faridabad
India

Furthermore, the quality of software product is very important and essential since for example in some sensitive systems – such as, real-time systems, control systems, etc. – the poor quality may lead to financial loss, mission failure, permanent injury or even loss of human life. There are several definitions for "software Quality" term, for examples, it is defined by the IEEE [1990] as the degree to which a system, component or process meets specified requirements and customer (user) needs (expectations). Pressman [2004] defines it as "conformance to explicitly stated functional and performance requirements, explicitly documented development standards, and implicit characteristics that are expected of all professionally developed software." The ISO, by contrast, defines "quality" in ISO 14598-1 [ISO, 1999] as "the totality of characteristics of an entity that bear on its ability to satisfy stated and implied needs. There are a number of quality models in software engineering literature, each one of these quality models consists of a number of quality characteristics (or factors, as called in some models). These quality characteristics could be used to reflect the quality of the software product from the view of that characteristic. Selecting which one of the quality models to use is a real challenge. In this paper, we will discuss the contents of the following quality models:

1) **Boehm's Quality Model.**
2) **McCall's Quality Model.**
3) **FURPS Quality Model.**
4) **ISO 9126 Quality Model.**

In addition, we will focus on a comparison between these quality models, and find the key differences between them.

1) 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 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.

Proc. of the Intl. Conf. on Recent Trends In Computing and Communication Engineering -- RTCCE 2013

ISBN: 978-981-07-6184-4

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 portability**,** utility which is further refined into reliability, efficiency and human engineering; and 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.

2) **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 Electrics Model of 1977). McCall's quality model defines and identifies the quality of a software product through addressing three perspectives:

a) 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.

b) Product revision is the ability to undergo changes, including error correction and system adaptation. It covers maintainability, flexibility and testability criteria.

c) 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.

3) **FURPS Software Quality Model**

The characteristics that are taken into consideration in FURPS model are:

a) Functionality includes feature sets, capabilities and security;

b) Usability includes human factors, consistency in the user interface, online and context-sensitive help, wizards, user documentation, and training materials;

c) Reliability includes frequency and severity of failure, recoverability, predictability, accuracy, and mean time between failure (MTBF);

d) Performance prescribes conditions on functional requirements such as speed, efficiency, availability, accuracy, throughput, response time, recovery time, and resource usage;

e) Supportability includes testability, extensibility, adaptability, maintainability, compatibility,

Configurability, serviceability and localizability / internationalization. Architectural integrity is not covered in the model. 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.

4) **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.

Proc. of the Intl. Conf. on Recent Trends In Computing and Communication Engineering -- RTCCE 2013

ISBN: 978-981-07-6184-4

A. METHODOLOGY

This section discusses the chosen research methodology. A discussion of alternative research methods is included in Appendix at the last of conclusion.

1) The First method selected will use questionnaires to measures the impact of lifecycles on the factors that influence the outcomes of software project. The resultant data will be used to drive a model selection framework. Which will suggest how suggest how suitable a lifecycle model is for a given project. The framework recommendations will be examined and discussed using a set of case studies.

2) Questionnaires provide a structured approach to gathering date and that closed questions, providing a limited list of responses, ensures easy transcription for processing. This makes closed questions suitable for gathering lifecycle impact data, while open questions can be used to solicit and would typically be gathered from free-text entry boxes.

3) In my experience, project lifecycle model choice is often made automatically and without consciously considering alternative models. This suggests volunteers may need time to gather their thoughts before answering model selection questions. Questionnaires are ideal in this situation, since there is no (realistic) set time limit for completing them.

B. ANALYSIS/COMPARISON

In this research paper, we have studied different types of software quality models like McCall, ISO 9126, 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 belonging to four quality models. Two characteristics is common only to three quality models, that is , the functionality and maintainability characteristics. Two characteristics 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 modifiability in 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 maintainability characteristic, 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.

1) There are some criteria‟s/goals that support Boehm model are: Testability, Understandability, Efficiency Modifiability, Reliability, Portability, and Human Engineering.

2) There are some criteria‟s/ goals that support McCall model are: Correctness, Maintainability, Testability, Flexibility, Reliability, Usability, Interoperability Reusability, Integrity, Efficiency, and Portability.

3) There are some characteristics/attributes/goals that support FURPS model is: Reliability, Usability, Functionality Performance, and Supportability

4) There are some criteria‟s/goals that support ISO 9126 model are: Reliability, Maintainability, Portability, Usability, Functionality, and Efficiency.

D. PROPOSAL

1) Define your organization‟s need and goals for software quality. If you don‟t know what your organizations need in terms of quality, you should not waste time on a software quality model. Answer the following questions:

a) What are your customer quality priorities?

b) Do you have processes in place to monitor customer preferences?

c) How do your customers feel about your products and services versus those of yours competitors?

d) Do you have special needs such as regulations or safety concerns?

e) Do you have special needs such as regulations or safety concerns?

f) Do you have contractual or legal requirements to use a particular model?

2) Identify which quality elements are most important to your business goals.

3) Choose a model. Based on elements you selected in step 2.

4) Develop details and examples to explain the software quality factors which are most important to your organization. These will help communicate priorities to your teams.

5) Build the quality factors and the quality model into your development and test methodologies.

Proc. of the Intl. Conf. on Recent Trends In Computing and Communication Engineering -- RTCCE 2013

ISBN: 978-981-07-6184-4

6) If you have accomplished the steps above, you have organized a software development, test and quality process which systematically addresses the software quality elements which match your organization's strategic goals. But things change, so periodically recalibrate with your staff and customers to ensure agreement on goals and priorities.

E. *Recommendations*

- Identify a small set of agreed-upon, high-level quality attributes, and then, in a top-down fashion decompose each attribute into a set of subordinate attributes.
- Distinguish between internal and external metrics.
- Identify type of users for each high-level quality attributes.
- Put the pieces together; constructing the new models that implement ideas from international standards: ISO-9126.

F. *CONCLUSION*

We have studied different types of software quality models in software engineering each of these quality models consists of number of characteristics. Selecting which one of the quality models to use is a real challenge. In this paper, we have discussed and compared the following quality models:

1) McCall's Quality Mode.
2) Boehm's Quality Model.
3) FURPS Quality Model.
4) ISO 9126 Quality Model

Based on the discussion of the five quality models and on the comparison between them, the following comments could be written.

1) In McCall's quality model, the quality is subjectively measured based on the judgment on the person(s) answering the questions (yes or no questions).

2) Three of the characteristics are used in the ISO 9126-1 quality model as sub-characteristics from other characteristics.

3) The FURPS quality model is built and extended to be used in the IBM Rational Software Company. Therefore, it is a special-purpose quality model, that is, for the benefits of that company.

4) The ISO 9126-1 quality model is the most useful one since it has been build based on an international consensus and agreement from all the country members of the ISO organization.

G. *Future Work*

After studied of all these models we can build a new software quality model. During creating a new model the analysis step assisted us to benefit from existing general quality models and simultaneously avoiding repetition of such limitations.

References

[1] Hoyer, R. W. and Hoyer, B. B. Y., "What is quality?" Quality Progress, no. 7, pp. 52-62, 2001.
[2] Geoff, D., "A model for Software Product Quality", IEEE Transactions on Software Engineering, 21(2nd): 146-162.1995.
[3] B. W., Brown,et.al, "Characteristics of Software Quality. North Holland Publishing, Amsterdam, The Netherlands, 1978.
[4] M. Broy, F. Deissenboeck, and M. Pizka. Demystifying maintainability. In Proc. 4th Workshop on Software Quality (4- WoSQ), pages 21–26.
[5] Breivold, H.P. et.al, „Using Software Evolvability Model for Evolvability Analysis", Mälardalen University, 2008.
[6] Forrest Shull, et.al, "Contributions to Software Quality", IEEE Software, vol. 23, no. 1, pp. 16-18, Jan/Feb, 2006.

About Author (s):

[Software Quality is the conformance to explicitly stated functional and performance requirements, explicitly documented development standards, and implicit characteristics that are expected of all professionally developed software.]