

Compilation of Software Quality Factors and Criteria along with their Description for a Quality Product

Basit Habib
Bahauddin Zakariya University Multan

ABSTRACT

To achieve a quality product it is necessary for the developer to understand how it can be achieved. Theoretical study is the preliminary phase of understanding how to perform any activity and knowing its pros and cons. For achieving a quality product this paper shows a quick and elaborative review for a quality product.

Categories and Subject Descriptors

- Compilation of the Software Quality Factors and corresponding Quality Criteria along with their description in order of popularity of the factors in the four Quality Models.
- Summary of the Occurrence of Quality Factors in the Quality Models.
- Frequency of Occurrence of Quality Criteria in Quality Models.

General Terms

Reliability, Experimentation, Performance, Design,.

Keywords

Quality, Factors, Criterion, Quality Models

1. INTRODUCTION

A quality product can be developed on a platform where some attributes are described related to quality. This quality attributes base on the factors and there corresponding criterion.

A quality model is a set of Factors. Every factor is a combination of different Criteria. To develop a good quality product, it is important to understand the relationship between factors, criterion and even the occurrence of a specific factor as it shows betterment towards the quality [2].

2. BACKGROUND

For achieving quality in products has always been an aim for the developers. For achieving this aim different constraint from different scientist, developers and workers have been defined in different eras. The quality of a software product begun in the late 70,s when a group of scientists gave the concept of a quality Model. In this concept they described their own factors an certain criterion against their factors.

2.1 History of Quality Models:

The idea of quality of software product was introduce by McCall *et al.* (1977). The idea soon expanded and several other authors explored in the refinement of the idea. A yearly order will be observed in the following discussion. We will restrict ourselves to the literature related to following four models as Software Quality Engineering Society recognizes only those[1], [2], [3].

- McCall's Quality Model.
- Boehm's Quality Model.

- Dromey's Quality Model.
- ISO/IEC 9126 Quality Model.

The first step of achieving quality is to understand in descriptive manner about factors and the criterion against those factors. A brief description is given as under:

3. SOFTWARE QUALITY FACTORS AND THEIR QUALITY CRITERIA ALONG WITH THEIR DESCRIPTION

3.1 Portability

Strategy of developing software for one platform with an intent to get easily adapted for other platforms (Transferability) Adoptability The capability of how to survive a change you had not been asked for

Co- Existence The capability of the software to co-exist with other independent software in a common environment sharing common resources.

Compliance The capability of the software to strictly maintain standards, conventions, or practices

Instability The capability of the software product to get installed in a specific environment

Replacability The capability of the software product to be used in place of another specified software product for the same purpose in the same environment [5], [3], [2].

3.2 Reliability

Ability of the software: not to go failed while running, and to work as per design (dependability)

Accuracy The condition or quality of the system of being correct and freedom from error or defect. Determines the degree of reliability

Consistency After every time and manner the correctness is the same. Error Tolerance It is a software that does not unduly penalize user errors [5], [3], [2].

3.3 Efficiency

A software should fulfil its purpose without waste of resources i.e. getting most of the utilized resources. It is usually gauged with respect to time and storage.

Execution Efficiency The ability of the software to optimally use the processor to get the intended task or purpose performed.

Storage efficiency It is the ability to store and manage data that consumes the least amount of space with little to no impact on performance [5], [3], [2].

3.4 Maintainability

It is the ease with which a product can be maintained in order to isolate and/or correct defects and/or their cause, to meet new requirement, and to coop with the changed environment.

Conciseness: shortness in size and comprehensiveness in scope of the Software (module).

Modularity: It is the degree to which a system's components may be separated and recombined. It helps apply Pareto Principle which states that 80% of effects come from 20% of areas.

Self Descriptiveness:

It provides simplicity by reducing users' memory load and improves his efficiency by letting them retain their capacity for their tasks instead of bothering with the system.

Simplicity: The property, condition, or quality of being simple [5], [3], [2].

3.5 Usability

The function performed by the program is also useful elsewhere, is robust against human error, or does not require excessive core memory.

Communicativeness The ability of the software to communicate its features to the user intuitively. This improves its usability.

Operability The capability of the software product to enable the user to operate and control it

Training The action of teaching users proper use of the software [5], [3], [2].

3.6 Functionality

At every stage, it must be assured that the software is fully functional in all of its areas of application.

Accuracy The condition or quality of the system of being correct and freedom from error or defect. Help gauge functionality

Compliance The capability of the software to strictly maintain standards, conventions, or practices

Interoperability Capability of the software product to interact with one or more specified systems

Security The capability of the software product to protect information and data from unauthorized access

Suitability The capability of the software to provide an appropriate set of functions to perform specified tasks and to meet user objectives [5], [3], [2].

3.7 Testability

The ability of a software or part its of to get tested to be assured of its being error-free and as per requirements

Instrumentation It refers to an ability of the software to monitor or measure the level of a product's performance, to diagnose errors and to write trace information.

Modularity It is the degree to which a system's components may be separated and recombined. It helps apply Pareto Principle which states that 80% of effects come from 20% of areas.

Self Descriptiveness It provides simplicity by reducing users' memory load and improves his efficiency by letting them retain their capacity for their tasks instead of bothering with the system.

Simplicity The property, condition, or quality of being simple [5], [3], [2].

3.8 Reusability

The ease of reusing the software in different contexts

Simplicity The property, condition, or quality of being simple

Machine and Software Independence

The autonomy of the hardware or software to be used independently in any system [5], [3], [2].

3.9 Correctness

To evaluate if basic principles have been followed

Completeness The requirements are properly fulfilled.

Consistency Agreement or logical coherence among modules

Traceability It refers to the completeness of the information about every step in a process of software development

3.10 Flexibility

The ease of making changes required by changes in the operating environment

Expandability Ability of a computer system to accommodate additions to its capacity or capabilities.

Generality Independence of software from unnatural restrictions and limitations

Self Descriptiveness It provides simplicity by reducing users' memory load and improves his efficiency by letting them retain their capacity for their tasks instead of bothering with the system [5], [3], [2].

3.11 Human Engineering

It is about robustness, integrity, accessibility, and communicativeness of the involved humans.

Accessibility The degree to which a system is available to as many users as required.

Communicativeness The ability of the software to communicate its features to the user intuitively. The involved human beings will find them at a higher level of ease.

Integrity To decrease cost, reduce risks, and ensure compliance, access beyond a defined level may not be available to users [5], [3], [2].

3.12 Integrity

Protection of the program from unauthorized access

Access Audit It is to establish how well a system performs in relation to exercise its control on access.

Access Control It refers to exerting control over the system and defines who can interact with the system along with the level of access [5], [3], [2].

3.13 Modifiability

The ease with which a software system or component can be modified to correct faults, improve performance or other attributes, or adapt to a changed environment.

Augmentability The capability of the software to get additional features accommodated to coop with the fresh requirements

Structurdness The state or condition of software of being well-thought and ingenious [5], [3], [2].

3.14 Undersatndability

The relevant purpose of the software product must be clearly documented keeping the respective user-context into account.

Conciseness Shortness in size and comprehensiveness in scope of the Software (module).

Consistency Agreement or logical coherence among modules

Legibility The ease with which parts of the software may be recognized and can be organized into a coherent pattern

Structurdness The state or condition of software of being well-thought and ingenious [5], [3], [2].

3.15 Interoperability

The efforts required to couple the system to another system

Modularity It is the degree to which a system's components may be separated and recombined. It helps apply Pareto Principle which states that 80% of effects come from 20% of areas.

Communality possession of common features or attributes[5], [3], [2].

3.2 Occurrence of Quality Factors in the Quality Models

The following table describes the frequency of factors and Criterion occurring against them [4],[7],[8],[9].

Table 1.Occurrence ofQuality Factors and their Criterion

| Factor | Occurrence of Factors | No. of Criteria |
|-------------------|-----------------------|-----------------|
| Portability | 4 | 5 |
| Reliability | 4 | 3 |
| Efficiency | 4 | 2 |
| Mainatnability | 3 | 4 |
| Usability | 3 | 3 |
| Functionality | 2 | 5 |
| Testability | 2 | 4 |
| Reusability | 2 | 2 |
| Understandability | 1 | 4 |
| Correctness | 1 | 3 |
| Flexibility | 1 | 3 |
| Human Engineering | 1 | 3 |
| Integrity | 1 | 2 |
| Interoperability | 1 | 2 |
| Modifiability | 1 | 2 |

3.2. The Most Popular Factors: Portability, Reliability and Efficiency

Table 3.2 summarizes the occurrence of factors in quality models. The two factors namely Portability and Reliability have been given by all four quality models [4],[5],[6],[9].

3.2.1 Portability

Portability which is actually the strategy of developing software for one platform with an intent to get easily adapted for other platforms (i.e. Transferability). The ever changing environment makes us consider portability as the essential factor as absence of the quality takes off the potential to coop with the dynamic world. The all five quality criteria associated

with the factor emphasize on coping with the changing conditions and getting the software adjusted accordingly. This emphasis can be witnessed by exploring each of the five criteria covered under this umbrella namely: Adoptability which is the capability of how to survive a change you had not been asked for; Co- Existence as the capability of the software to co-exist with other independent software in a common environment sharing common resources; the capability of the software to strictly maintain standards, conventions, or practices as Compliance;Instability which is the capability of the software product to get installed in a specific environment; and last but not least Replacability as the capability of the software product to be used in place of another specified software product for the same purpose in the same environment [5], [3], [2].

3.2.2 Reliability

Reliability which is the ability of the software not to go failed while running, and to work as per design. It actually determines how dependable the software is? The ultimate quality of the software is directly proportional to its reliability. This fact lets the factor secure its place in the top slot along with portability. Three criteria included under the heading are accuracy, consistency, and Error Tolerance. Obviously all three criteria contribute in reliability. The first criterion requires that the system is correct and free from error or defect and thus determines the degree of reliability. The second criterion checks after every time and/or manner the correctness remains same. The third and last criterion requires that the software does not unduly penalize user errors.

3.2.3 Efficiency

An efficient software fulfils its purpose without waste of resources i.e. getting most of the utilized resources. Efficiency is usually gauged with respect to time and storage. The first criterion of the factor namely Execution Efficiency is the ability of the software to optimally use the processor to get the intended task or purpose performed. The other aspect of the efficiency is covered with the other criterion of the factor i.e. Storage Efficiency. The most valuable resources are time and storage thus efficiency should be gauged accordingly. Storage efficiency is the ability to store and manage data that consumes the least amount of space with little to no impact on performance.

3.3 The Least Popular Factors

There are seven factors which occurred only once. Those are Correctness, Flexibility, Integrity, Interportability, Human Engineering, Modifiability, and Understandability. The first four belong to the McCall's model while remaining belong to the Boehm's model. McCall and Boehm are the pioneers in defining quality models. Both presented their models in late 70's whereas the two other models namely Dromey's and ISO/IEC 9126 were presented with a time lag of over a quarter century. Although the revision of the factors have started as early as Boehm's model was introduced just after McCall's models was introduced when the number of factors was shrunk from 11 to 7 with commonality of only four factors, the factors in a sense got standardized by the introduction of Dromey's and ISO/IEC 9126 models where number of factors stabilized around 7 with commonality of 6 factors. The least popular factors listed at the beginning of the section are all belong to the two earlier models. Stabilization of the factor during the quarter century wide period of maturity made these factor disappeared.

3.4 The Relation between Criterion and Factors

The relation between various criteria listed against various quality factors given by the four quality models is attempted to be established here. For the purpose all factors are tabulated against all criteria. The tabulation provides information regarding the relation by putting tick marks appropriately. Moreover, frequency of occurrence of the criteria against quality factor are calculated. Similarly, occurrence of quality factors against criteria is also counted. The Table 2 provides this information. The table is split into three groups with respect to frequency of occurrence of certain criteria against quality factor. The first part groups the criteria which occurred most frequently (i.e. thrice). Similarly group of doubly occurring criteria and singly occurring criteria were made. Out thirty four criteria, four fall in the first group of most frequently occurred criteria whereas the second group of doubly occurring criteria has five criteria. The least occurring criteria has rest of the twenty five criteria [2],[3],[7],[8].

The analysis of the table is based on these groups. The groups are summarized in Table 2.

Table 2. Relation between Criterion and Factors.

| Frequency of Occurrence | Criteria | No. of Criteria |
|-------------------------|--|--|
| 3 | Consistency | Correctness Reliability Understandability |
| | Modularity | Interoperability Maintainability Testability |
| | Self Descriptiveness | Flexibility Maintainability Testability |
| | Simplicity | Maintainability Reusability Testability |
| 2 | Accuracy, Communicativeness, Compliance, Conciseness, Structurdness | 5 |
| 1 | Access audit, Access Control, Accessibility, Adoptability, Augmentability, Co- Existence, Commonality, Completeness, Error Tolerance, Execution Efficiency, Expandability, Generality, Hardware and Software Independence, Instability, Instrumentation, Integrity, interoperability, Legibility, Operability, Replacability, Security, Storage efficiency, Sutability, Traceability, Training | 25 |

4. Rationale of the Relation Between Quality Criterion and Quality Factors

Table 3. Criterion and their relative Factors.

| Criterion | Factors |
|----------------------|-------------------|
| Consistency | Correctness |
| | Reliability |
| | Understandability |
| Modularity | Interoperability |
| | Maintainability |
| | Testability |
| Self Descriptiveness | Flexibility |
| | Maintainability |
| | Testability |
| Simplicity | Maintainability |

| Criterion | Factors |
|------------------------|-------------------|
| | Reusability |
| | Testability |
| Accuracy | Functionality |
| | Reliability |
| Communicativeness | Human Engineering |
| | Usability |
| Compliance | Functionality |
| | Portability |
| Conciseness | Maintainability |
| | Understandability |
| Structurdness | Modifiability |
| | Understandability |
| Access Audit | Integrity |
| Access Control | Integrity |
| Accessibility | Human Engineering |
| Adaptability | Portability |
| Augment ability | Modifiability |
| Co-Existence | Portability |
| Commonality | Interoperability |
| Completeness | Correctness |
| Error Tolerance | Reliability |
| Execution Efficiency | Efficiency |
| Expandability | Flexibility |
| Generality | Flexibility |
| HW and SW Independence | Reusability |
| Install ability | Portability |
| Instrumentation | Testability |
| Integrity | Human Engineering |
| Interportability | Functionality |
| Legibility | Understandability |
| Operability | Usability |
| Replaceability | Portability |
| Security | Functionality |
| Storage Efficiency | Efficiency |
| Suitability | Functionality |
| Traceability | Correctness |
| Training | Usability |

4.2 Rationale of the relation

1. Consistency as the quality criterion reflects consistent behaviour of the software. Here it means that after every time and manner the correctness/ reliability/ understandability remains at the same level. These quality factors directly affect the behaviour of the software in the changed/ modified circumstances and we still expect same level of performance. Thus the consistency is desired.
2. Modularity is the degree to which a system's components may be separated and recombined. It drastically improves maintainability, testability and interportability of software as it becomes far easier to manage the final product if it is suitably moduled.
3. Self Descriptiveness provides simplicity by reducing users' memory load and improves his efficiency by letting them retain their capacity for their tasks instead of bothering with the system. Presence of this criterion influences flexibility, maintainability, and testability as things go intuitively understood.
4. The property, condition, or quality of being simple. If simplicity has been induced into the software it will be

easier to get maintained and tested. Moreover reusing the software in different contexts will also be easier.

5. Accuracy is the condition or quality of the system of being correct and freedom from error or defect. It determines the degree of reliability and affects the functionality of the software.
6. Communicativeness is the ability of the software to communicate its features to the user intuitively. It thus improves its usability and enhances robustness, integrity, accessibility, and communicativeness of the involved humans.
7. Compliance is the capability of the software to strictly maintain standards, conventions, or practices. By doing so functionality and portability get improved as standardization makes it acceptable everywhere.
8. Conciseness means shortness in size and comprehensiveness in scope of the Software. A compact and concise software will not only be easy to comprehend but will also be easier to maintain because unnecessary details will have been removed.
9. The state or condition of software of being well-thought and ingenious is known as structuredness. A well structured software can be modified easily. Its understandability will no doubt be improved.
10. Access audit is to establish how well a system performs in relation to exercise its control on access. So it decreases cost, reduces risks, and ensures compliance, access beyond a defined level may not be available to users and improves integrity.
11. Access control refers to exerting control over the system and defines who can interact with the system along with the level of access. So it decreases cost, reduces risks, and ensures compliance, access beyond a defined level may not be available to users and improves integrity.
12. Accessibility is the degree to which a system is available to as many users as required. It thus improves its usability and enhances robustness, integrity, accessibility, and communicativeness of the involved humans.
13. Adaptability is the capability of how to survive a change you had not been asked for. By this portability gets improved as adaptability makes it adjusted everywhere.
14. Augment ability is the capability of the software to get additional features accommodated to coop with the fresh requirements. So the system can be modified to get adjusted to changed environment.
15. The capability of the software to co-exist with other independent software in a common environment sharing common resources obviously makes it portable.
16. Commonality is the possession of common features or attributes. This reduces the efforts required to couple the system to another system.
17. Completeness means that the requirements are properly fulfilled and correctness is the degree at which basic principles have been followed. This forms the relation.
18. Error tolerant software is one that does not unduly penalize user errors. This property improves the ability of the software: not to go failed while running, and to work as per design and thus makes the software dependable.
19. Execution efficiency is the ability of the software to optimally use the processor to get the intended task or purpose performed. It makes the software fulfil its purpose without waste of resources i.e. software becomes efficient.
20. Expandability is the ability of a computer system to accommodate additions to its capacity or capabilities. It makes the system flexible by providing ease in making changes required by changes in the operating environment.
21. Independence of software from unnatural restrictions and limitations induce generality into the system. It makes the system flexible by providing ease in making changes required by changes in the operating environment.
22. The autonomy of the hardware or software to be used independently in any system. It makes the system reusable by providing ease of reusing the software in different contexts.
23. If a software product could easily get installed in a specific environment i.e. software developed for one platform could get easily adapted for other platforms thus it gets portable.
24. Instrumentation refers to the ability of the software to monitor or measure the level of a product's performance, to diagnose errors and to write trace information. This property makes the software or part itself to get tested.
25. Integrity of the software does not grant access beyond a defined level to the involved humans
26. Interportable software requires little efforts to couple the system to another system. But at every stage, it must be assured that the software is fully functional in all of its areas of application otherwise interoperability will be of no use.
27. Legibility is the level of ease with which parts of the software may be recognized and can be organized into a coherent pattern. For this, relevant purpose of the software product must be clearly documented keeping the respective user-context into account which improves understandability.
28. Operability is the capability of the software product to enable the user to operate and control it. This makes the function performed by the program useful elsewhere as well.
29. Replaceability is the capability of the software product to be used in place of another specified software product for the same purpose in the same environment. It makes the system adaptable to other platforms.
30. Security is the capability of the software product to protect information and data from unauthorized access. Secured software maintains its functionality without compromise
31. Storage efficiency is the ability to store and manage data that consumes the least amount of space with little to no impact on performance. The software thus becomes efficient in general.
32. Suitability is the capability of the software to provide an appropriate set of functions to perform specified tasks and to meet user objectives. It thus guarantees full functionality of the software.

33. Traceability refers to the completeness of the information about every step in a process of software development. It improves correctness.
34. Training is the action of teaching users proper use of the software. The system stays usable.

5. CONCLUSION

Quality depends on the design of the product. For designing a product, it is important for the developer to have primary knowledge of Factors that affect the quality of a product. And the criteria which are the building blocks of a factor. This paper covers the study and comparison of quality models as it tells factors, criteria and their occurrence along with their description under one roof. Along with this it shows the right path to the developer for developing a quality product.

6. REFERENCES

- [1] Software Engineering- Product Quality, Part 3, 4, Version 1.0 (ISO/IEC TR 9126), url: http://www.iso.org/iso/home/store/catalogue_tc/, visited: 7th July, 2012.
- [2] Al-Qutaish, R. E., "Quality Models in Software Engineering Literature: An Analytical and Comparative Study," *Journal of American Science*, 2010. Vol. 6, no. 3, pp. 166-175.
- [3] Kitchenham, B., Pfleeger, S. L. "Software Quality: the Elusive Target," 1996. *IEEE Software*, vol. 13, no. , pp. 12-21.
- [4] Wirth, N. "A Brief History of Software Engineering," *IEEE Annals of the History of Computing*, 2008. Vol. 30 no.3, pp. 32-39.
- [5] Boehm, B. W.; Brown, J. R.; Kaspar, H.; Lipow, M.; McLeod, G.; and Merritt, M., "Characteristics of Software Quality," *North Holland Publishing, Amsterdam, The Netherlands, 1978, vol., no., pp.*
- [6] Deissenboeck, F.; Juergens, E.; Lochmann, K.; and Wagner, S. 2009. **Software quality models: purposes, usage scenarios and requirements.** In *Proceedings of the 7th ICSE conference on Software quality* (Munich, Germany, 2009).
- [7] Dromey, R. G., "A Model for Software Product Quality," *IEEE Transactions on Software Engineering*, 1995, vol. 21, no., pp 146-162..
- [8] History of Software Engineering. In *Wikipedia*. Retrieved June 27 2012, from http://en.wikipedia.org/wiki/History_of_software_engineering.
- [9] Software Engineering- Product Quality, Part 3, 4, Version 1.0 (ISO/IEC TR 9126), url: http://www.iso.org/iso/home/store/catalogue_tc/, visited: 7th July, 2012.