CREAT. MATH. INFORM.
Volume **25** (2016), No. 1,
Pages 57 - 62

Online version at https://creative-mathematics.cunbm.utcluj.ro/
Print Edition: ISSN 1584 - 286X; Online Edition: ISSN 1843 - 441X
DOI: https://doi.org/10.37193/CMI.2016.01.07

A comparative study on existing software quality models

MARA DIANA HAJDU MACELARU

ABSTRACT. A comparison between main software quality models is done in this article. We analize the five quality models, we point out the good / bad issues in each model and we compare them based on the attributes/ characteristics, based on number of attributes as well as we define an algorithm to perform a comparison based on the importance of attributes in each model.

1. INTRODUCTION

In this paper we will analyze various software quality models, and their attributes / characteristics. The software quality models that will be analyzed and compared are: McCall's Quality Model, Boehm's Quality Model, Dromey's Quality Model, FURPS Quality Model and ISO 9126 Quality Model.

2. SOFTWARE QUALITY MODEL ANALYZED

McCall's Mod

Developed in 1977, the McCall quality model, defines quality from three perspectives: Product operation, Product revision and Product transition. In more details the McCall's Quality model consist of 11 quality factors to describe the external view of the software - meaning from the user point of view and 23 quality criteria to describe the internal view of the software quality meaning from developer's view.
Each perspective contains a set of quality factors:

- Product revision: Maintainability, Flexibility, Testability
- Product operations includes: Correctness, Reliability, Efficiency, Integrity, Usability
- Product transition includes: Portability, Reusability, Interoperability

The major contribution at the McCall model was the relations between quality characteristics and metrics.
The minus of this model is the accuracy in the measurement of the software quality due to the fact that the actual metric is computed by answering "yes" and "no" on the questions measuring a quality criteria.

Boehm's Model

Developed in 1978, this model attempts to qualitatively define the quality of software. The high-level characteristics address three main questions that a buyer of a software has:

- As-is utility: How well can I use?
- Maintainability: How easy is to understand, modify and re-test?
- Portability: Can I still use it if I do an environment change?

Received: 15.03.2015. In revised form: 28.09.2015. Accepted: 01.02.2016
2010 *Mathematics Subject Classification.* 68N30, 68N99.
Key words and phrases. *Software quality models attributes characteristics, software engineering metrics.*

The 7 quality factors that together represents the qualities expected from a system in case of Boehm's model are:

- in As-is utility we have the factors: Reliability, Efficiency, Human Engineering
- in Maintainability : Testability, Understandability, Modifiability
- Portability

Dromey's Model

Developed in 1995, Dromey's proposes a model that recognizes that quality evaluation differs for each product and that a more dynamic idea for modeling the process is needed.
The model has at a high-level view fours properties: **Correctness** that evaluates if some basic principles are violated, that has as quality attributes the Functionality and Reliability; **Internal** that evaluates how well a component has been deployed according to its use, that has as quality attributes: Maintainability, Efficiency, Reliability; **Contextual** deals with the external influence on the use of the software, and has as quality attributes: Maintainability, Reusability, Portability, Reliability; **Descriptive** measures the descriptiveness of a component, and has as quality attributes: Maintainability, Efficiency, Reliability, Usability.

Furps Model

Presented in 1992, the model is based on the following five characteristics: Functionality, Usability, Reliability, Performance, Supportability.
The model decomposes the characteristics into two main requirements:

- **Functional requirements** defined by input and expected output
- **Non-functional requirements** meaning Usability, Reliability, Performance and Supportability

FURPS has the disadvantage that it does not consider the portability aspect, which is an important criteria for application development.

ISO 9126

The ISO 9126 model was based on McCall's and Boehm's models. The model has two main parts:

- attributes of internal and external quality
- the quality in use attributes

The internal quality is the totality of characteristics of the software product from an internal view, while external quality is the totality of characteristics of the software product from an external view.
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.
The main characteristics from this model are: Functionality, Reliability, Usability, Efficiency, Maintainability, Portability.
The ISO 9126 quality model is the most useful since it has been build on an international consensus and agreement from all the country members of the ISO organization.

3. Software quality model comparative study

Each of the software quality models contains a set of characteristics. We can do a first comparison based on the software quality characteristics to highlight the common characteristics as well as less common ones. In Figure1 a comparison of the quality factors/characteristics from each quality model is presented.

We can notice that there is one characteristic common to all five models which is Reliability. Other characteristics that are found in four of the five models are: Efficiency, Usability, Portability. Less common is Human Engineering, Understandability, Modifiability, Integrity, Performance and Supportability.

Factors/Characteristics	McCall	Boehm	Furps	Dromey	ISO 9126
Maintainability	x			x	x
Flexibility	x				
Testability	x	x			
Correctness	x				
Reliability	x	x	x	x	x
Efficiency	x	x		x	x
Integrity	x				
Usability	x		x	x	x
Portability	x	x		x	x
Reusability	x			x	
Interoperability	x				
Human Engineering		x			
Understanability		x			
Modifiability		x			
Functionality			x	x	x
Performance			x		
Supportability			x		

Figure 1. A comparison between the quality models based on their quality attributes

A comparison based on the year it was developed and the number of the main characteristics that define the model can be seen above:

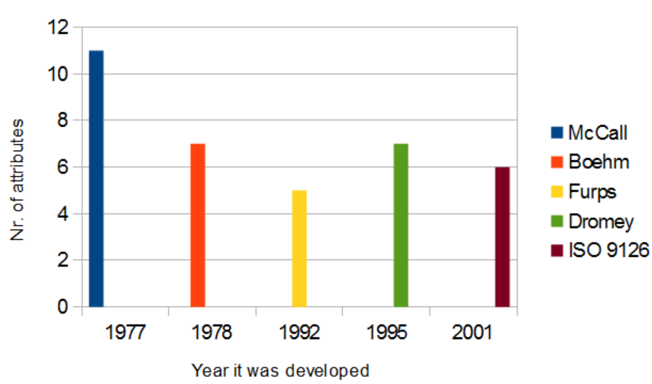

Figure 2. A comparison between the quality models based on the year they were developed and number of quality attributes

Based on the way each quality model is designed we can try to compute the characteristics importance in each model. A way to compute is to consider equal the importance of each level as well as each characteristics.

For McCall model we have three perspective that we will consider to have the same importance each one. For example if we sum the importance to be 100% we can say that each perspective is of importance of $\sim$ 33,33% for this model.

The product revision has three main characteristics : Maintainability, Flexibility, Testability. If we consider to have all of them equal importance for the model we can say that each of this three characteristics have an importance of 11,11% for the McCall model. The product operations includes has 5 characteristics: Correctness, Reliability, Efficiency, Integrity, Usability. Considering the logic from above each of this characteristics will have 6.66% importance for the model. We do similar calculation for product transition that includes: Portability, Reusability, Interoperability and we get that each of this characteristics have 11,11% importance.

For Boehm's model we apply the same algorithm and we get the results:

- Reliability, Efficiency, Human Engineering has 11,11% importance for this model each of them
- Testability, Understandability, Modifiability has 11,11% importance for this model each of them
- Portability has 33,33% importance

For the Dromey's model we sum the characteristics from each of the four properties as we have common attributes. Having four properties will mean that we have 25% importance for each property in part. First property has 2 quality attributes means each will have 12.5% importance, second has 3 quality attributes meaning $\sim$8.33% importance for each attribute in part, third and fourth has 4 properties meaning $\sim$6.66% importance. For common attributes we sum the importance as seen in Table 1.
Using same algorithm for all the models we get the following table with the characteristics importance based on the quality model:

Factors/Characteristics	McCall	Boehm	Furps	Dromey	ISO 9126
Maintainability	11.11%			20.83%	16.66%
Flexibility	11.11%				
Testability	11.11%	11.11%			
Correctness	6.66%				
Reliability	6.66%	11.11%	12.50%	33.33%	16.66%
Efficiency	6.66%	11.11%		14.58%	16.66%
Integrity	6.66%				
Usability	6.66%		12.50%	6.25%	16.66%
Portability	11.11%	33.33%		6.25%	16.66%
Reusability	11.11%			6.25%	
Interoperability	11.11%				
Human Engineering		11.11%			
Understanability		11.11%			
Modifiability		11.11%			
Functionality			50.00%	12.50%	16.66%
Performance			12.50%		
Supportability			12.50%		

Table 1. Importance weight of each characteristics based on the quality model to which it refers

We also can draw a chart to highlight better the attributes importance based on the algorithm described above:

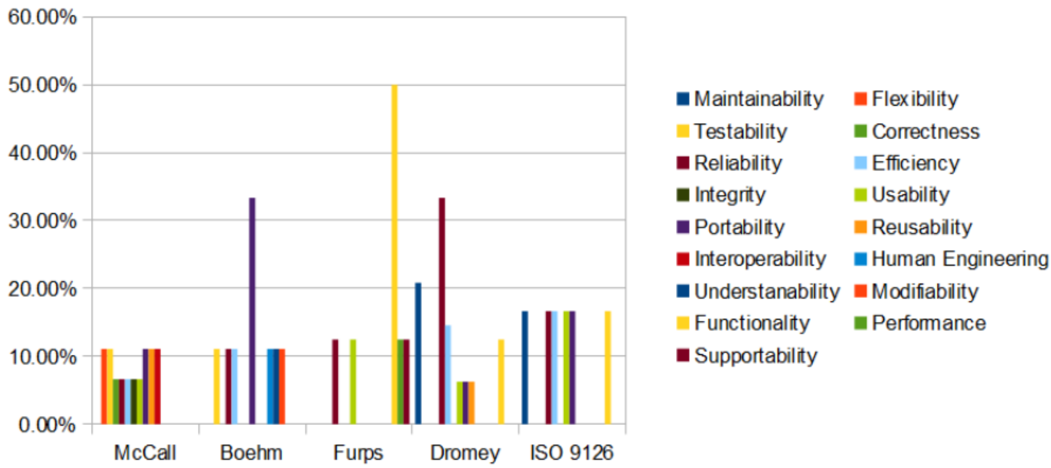

Figure 3. Chart representing the importance weight of each characteristics based on the quality model to which it refers

As we can notice the importance for attributes seems to be more balanced for McCall and ISO 9126 models. We also can notice characteristics, based on the described algorithm, that seems to be more important depending on the model: Portability for Boehm's model, Functionality for Furps model and Reliability for Dromey's model.

4. Conclusion

The importance of creating a software quality model that reflects better the quality of the products remains an aim in our days. As software development has evolved, the need of having a software quality model applicable in practice still remains an important topic in the research area.

The current article provides a comparative study of the five most important software quality models that were developed until now. Further work can include comparison of other software quality models with realistic data.

References

[1] Awan, S., Malik, F. and Javed, A., *An Efficient and Objective Generalized Comparison technique for Software Quality Models*, Int. J. Modern Education and Computer Science, **7** (2015), No. 12, 57–64

[2] Berander, P. and Damm, Lars-Ola, *Software quality attributes and trade-offs*, Blekinge Institute of Technology, 2005

[3] Deepshikha, J., *Analysis of Software Quality Models for Organizations*, Int. J. Latest Trends in Computing (E-ISSN: 2045-5364) **1** (2010), No. 2, 19–23

[4] Dromey, R. G., *A model for software quality*, IEEE Transactions on Software Engineering **21**, 1995

[5] Singh, I., *Different Software Quality Model*, Int. J. Recent and Innovation Trends in Computing and Communication, Volume 1, Issue 5, pp. 438–442

[6] Wagner, S., *Software Product Quality Control*, 2013, XII, 210pp., ISBN:978-3-642-38570-4

[7] Zeiss, B., Vega, D., Schieferdecker, I., Neukirchen, H. and Grabowski, J., *Applying the ISO 9126 Quality Model to Test Specifications*, *Software Engineering 2007 (SE 2007)*, Lecture Notes in Informatics (LNI) 105. Copyright Gesellschaft für Informatik, Köllen Verlag, Bonn, 2007, 231–242

[8] `http://sqa.net/iso9126.html`

Department of Mathematics and Computer Science
North University Center at Baia Mare
Technical University of Cluj-Napoca
Victoriei 76, 430122 Baia Mare, Romania
Email address: macelarumara@yahoo.com