Available online at www.sciencedirect.com

ScienceDirect

IERI Procedia 4 (2013) 88 – 92

Procedia IERI

www.elsevier.com/locate/procedia

2013 International Conference on Electronic Engineering and Computer Science

A Review of Models for Evaluating Quality in Open Source Software

Adewole Adewumi*, Sanjay Misra and Nicholas Omoregbe

Department of Computer and Information Sciences, Covenant University, Ota, Nigeria

Abstract

Open source products/projects targeting the same or similar applications are common nowadays. This makes choosing a tricky task. Quality is one factor that can be considered when choosing among similar open source solutions. In order to measure quality in software, quality models can be used. Open source quality models emerged due to the inability of traditional quality models to measure unique features (such as community) of open source software. The aim of the paper therefore is to examine the characteristic features, unique strengths, and limitations of existing open source quality models. In addition, we compare the models based on some selected attributes.

© 2013 The Authors. Published by Elsevier B.V. Open access under CC BY-NC-ND license.
Selection and peer review under responsibility of Information Engineering Research Institute

Key words: FLOSS, Open source software, Quality models

1. Introduction

Open source software is all around us today. They range from operating systems (such as Linux, Solaris, FreeBSD) to middleware/database technologies (such as Apache web server/MySQL) and then to end-user products such as web browsers (e.g. Mozilla Firefox). The list is in exhaustive.

* Corresponding author. Tel.: +2348034623562;
E-mail address: wole.adewumi@covenantuniversity.edu.ng.

2212-6678 © 2013 The Authors. Published by Elsevier B.V. Open access under CC BY-NC-ND license.
Selection and peer review under responsibility of Information Engineering Research Institute
doi:10.1016/j.ieri.2013.11.014

Most open source software projects are regarded to be of high quality from the perspectives of designers, writers and even those who use the software (Raja and Barry, 2005). The Institute of Electrical and Electronics Engineers (IEEE) defines quality as, "the degree to which a system, component or process meets specified requirements" or "the degree to which a system, component or process meets customer or user needs or expectations" (Pressman, 2005). In order to measure quality in software, quality models can be used.

The quality model approach is one of two main approaches for understanding the quality of software products (Haaland et al., 2010). Some quality models focus around a set of attributes/metrics used to distinctively assess quality by making quality a quantifiable concept. Example is the McCall model (McCall et al., 1977), the Boehm model (Boehm et al., 1976) (Boehm, et al., 1978), and the ISO9126 product quality standard (ISO, 2001). Due to the fact that these quality models ignored some attributes (such as community) unique to open source software, new quality models began to emerge in 2003 (Haaland, et al., 2010) specific to open source software.

The aim of this paper therefore is to identify the characteristic features, unique strengths and limitations of existing open source software quality models. This can serve as a guide to those intending to use any of the models and also to lay a foundation for improvements on the models. The rest of this paper is organized as follows. Section 2 presents the research method that was applied. Section 3 presents the characteristic features, unique strengths and limitations of existing open source quality models. Section 4 is a comparative study of the models based on selected criteria. Section 5 discusses the comparison made and Section 6 concludes the paper.

2. Research Method

We identified from literature (Haaland et al., 2010) the actual year (being 2003) when the first open source quality model emerged, we then decided to check through literature starting from 2003 all through to 2012 in order to find the models that had been proposed. As a whole there were 6 relevant publications namely: (Duijnhouwer and Widdows, 2003) (Wasserman et al., 2006) (Origin, 2006) (Samoladas et al., 2008) (Petrinja et al., 2009) (Ortega et al., 2010).

After retrieving the relevant publications, we examined each in order to identify the features of the model; the strengths of the model as well as the limitations that could be improved on. This is presented in the section that follows.

3. Features, Strengths and Limitations of Existing Quality Models for Open Source Software

This section presents a summary of the various quality models in the order of year in which they were proposed. For each model we identified the features and strengths as well as the identifiable limitations to be improved on where applicable. Table 1 shows this.

Table 1 Summary of Features, Strengths and Limitations to Improve on

Model	Features and Strengths	Limitations to Improve on
CapGemini Open Source Maturity Model	Consists of product and application indicators Can be updated on a regular basis through feedback from customers	
QSOS	Consists of four iterative stages namely: definition, evaluation, selection and qualification Supported by a tool called O3S Allows objective and traceable evaluation of free and open source software	Recent documentation such as version 1.7 need to be translated from French to English for wider use
OpenBRR	Accelerates software evaluation process through a systematic approach Ensures better decisions and increase confidence in selected open source software Open and customisable and can be applied to any business situation	The original goal of offering a vendor-neutral federated clearinghouse of quantifiable data on open source software packages to help drive adoption and development is yet to be achieved. The model's website contains only a single page and has remained static with no links to any useful resource material.
SQO-OSS	Hierarchical model that evaluates source code and community processes Automatic calculation of metric values Correlation of metric values to a set of predefined quality profiles Limits user interaction thereby reducing subjectivity Provides an infrastructure for developing new metrics, plugging them in and running them on open source projects of any size	
OpenSource Maturity Model	Tree level scale Simplicity and availability of tools for evaluation process	Need for industrial validation of the model and gathering of necessary feedback
QualOSS	Robustness and evolvability are the two factors upon which the model is developed Reduces subjectivity in the quality measurement process by automating quality measurement	

4. Comparison of the Quality Models

In this section, we compare the models based on the following criteria:

- Availability of published results online: Has any evaluation of open source projects/products using the model been published on the Web?
- Origin of the model: Is the model based on another model?
- Availability of tool support: Whether or not an automated tool exists to aid evaluation process?

Table 2 shows this comparison.

Table 2. Comparison of the quality models based on availability of published results online, model origin and tool support

Criteria Model	Availability of Published Results Online	Model Origin	Tool Support
CapGemini Open Source Maturity Model	Yes	ISO/IEC 9126 quality model	No
QSOS	Yes	ISO/IEC 9126 quality model	Yes
OpenBRR	Yes	ISO/IEC 9126 quality model; CapGemini Open Source Maturity Model; Navica Open Source Maturity Model	No
SQO-OSS	Yes	ISO/IEC 9126	Yes
OMM	Yes	Capability Maturity Model	Yes
QualOSS	Yes	CapGemini Open Source Maturity Model, QSOS & Open BRR	Yes

5. Discussion

From the comparative study in Table 2, we see that the six open source software quality models considered have the following kinds of origin namely:

- Originate from purely traditional software quality models such as (ISO/IEC 9126). Three of the models belong to this category and they are: CapGemini Open Source Maturity Model, QSOS, and SQO-OSS model
- Originate from a mix of traditional software quality models and contemporary open source software quality models. One of the models belongs to this category namely: OpenBRR.
- Originate from purely contemporary open source software quality models. One of the models belongs to this category namely: QualOSS.
- Originate from the Capability Maturity Model. OpenSource Maturity Model (OMM) is the only one in this category

The origins of the models determine the kind of attributes they possess. Finally, four out of the six models have tool support.

6. Conclusion and Future Work

This paper presents a review of six existing open source software quality models. The review presented the characteristic features, unique strength(s) and limitations of the models. Recommendations were also made for attending to the limitations of the models where applicable. A comparative study was done on the quality models based on three selected criteria which include: whether or not any evaluation of open source projects/products using the model has been published on the Web, the origin of the model and whether or not it provides tool support for the evaluation process. We found that the origin of the models can be classified into four as discussed in the previous section. We also found out that all of the models have results of their evaluation published on the Web. In addition, we found that four out of the models provided tool support. It is believed that this work will serve as guide for open source software evaluators when they intend to choose a model with which to evaluate open source software options.

We plan in the future to extend this work by adding more models that may emerge. We also plan to compare the quality of the models (in terms of results obtained when used to evaluate real-world open source software/products).

References

[1] Origin A. Method for Qualification and Selection of Open Source Software (QSOS), version 1.6. Available at: http://www.qsos.org/download/qsos-1.6-en.pdf, 2006.
[2] Boehm BW, Brown JR, Lipow M. Quantitative evaluation of software quality. *Proceedings of the 2nd International Conference on Software Engineering,* 1976.
[3] Boehm BW, Brown JR, Kaspar H, Lipow M, McLeod G, Merritt M. Characteristics of Software Quality. Vol. 1. North-Holland Publishing Company, 1978.
[4] Duijnhouwer FW, Widdows C. Open Source Maturity Model. Capgemini Expert Letter, 2003.
[5] Fagerholm F. Measuring and tracking quality factors in Free and Open Source Software projects. Masters' Thesis, University of Helsinki, 2007.
[6] Haaland K, Groven AK, Regnesentral N, Glott R, Tannenberg A, FreeCode AS. Free/Libre Open Source Quality Models-a comparison between two approaches. *4th FLOS International Workshop on Free/Libre/Open Source Software*, pp. 1-17. 2010.
[7] ISO, International Organization for Standardization. ISO 9126-1:2001, Software engineering - Product quality, Part 1: Quality model, 2001.
[8] McCall JA, Richards PK, Walters GF. Factors in Software Quality. General Electric, National Tech. Information Service, 1977.
[9] Ortega F, Coca P, Cortazar, DI. QualOSS: Quality of Open Source Software. 2010.
[10] Petrinja E, Nambakam R, Sillitti A. Introducing the OpenSource Maturity Model. In Proceedings of the 2009 ICSE Workshop on Emerging Trends in Free/Libre/Open Source Software Research and Development, pp. 37-41. IEEE Computer Society, 2009.
[11] Petrinja E, Sillitti A, Succi G. Comparing OpenBRR, QSOS, and OMM Assessment Models. Open Source Software: New Horizons, (2010): 224-238.
[12] Pressman RS. Software Engineering: a Practitioner's Approach. McGraw-Hill; 2005.
[13] Raja U, Barry E. Investigating Quality in Large-Scale Open Source Software. Fifth Workshop on Open Source Software Engineering (pp. 43-46). St Louis: ACM; 2005.
[14] Samoladas I, Gousios G, Spinellis D, Stamelos I. The SQO-OSS quality model: measurement based open source software evaluation. Open source development, communities and quality (2008): 237-248.
[15] Wasserman AI, Pal M, Chan C. Business Readiness Rating for Open Source. Proceedings of the EFOSS Workshop, Como, Italy. 2006.