# Evaluating Open Source Software Quality Models against ISO 25010

Adewole Adewumi, Nicholas Omoregbe
Department of Computer and Information Sciences
Covenant University
Ota, Nigeria
{wole.adewumi,
nicholas.omoregbe}@covenantuniversity.edu.ng

Sanjay Misra
Department of Computer Engineering
Atilim University
Ankara, Turkey
smisra@atilim.edu.tr

*Abstract*— **Quite a number of open source software quality models exist today. These models emerged as a result of the need to measure quality in open source software, which is quite unlike closed source, or proprietary software. ISO 9126 standard forms the basis from which most of these models derive. However, ISO 9126 standard has been replaced by ISO 25010. Therefore, as research endeavors progress towards evolving the "silver bullet" open source software quality model, it is the aim of this paper to evaluate existing open source software quality models against the ISO 25010 standard. The findings from this study reveal a candidate model (from among the existing models) that can be leveraged in deriving a generic open source software quality model.**

***Keywords—evaluation; open source software; quality models; ISO 25010; ISO 9126***

## I. Introduction

Prior to the emergence of open source software (OSS) quality models, the McCall, Dromey and ISO 9126 models were already in existence [1]. These models however did not factor in some quality attributes unique to OSS such as community – a body of users and developers formed around OSS who help to contribute to and popularize the OSS [2]. This gap was what led to the evolution of OSS quality models. The majority of the OSS quality models that exist today are derived from the ISO 9126 quality model [1] [3]. The International Organization for Standardization proposed it in 1991. It defines six quality characteristics namely: functionality, reliability, usability, efficiency, maintainability and portability. ISO 9126 views quality from three perspectives namely: internal, external and quality in use. The internal quality perspective focuses on static measures of intermediate products while the external quality perspective measures the behavior of the code when executed. 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 of use. It measures the extent to which users can achieve their goals in a particular environment, rather than measuring the properties of the software itself. The ISO 25010 model replaced ISO 9126 in 2010 [4].

ISO 25010 extends ISO 9126 to include computer systems, and quality in use from a system perspective [5]. The internal and external quality factors in ISO 9126 have been combined as the product quality factor in ISO 25010. Also, Security has been added as a characteristic rather than a sub-characteristic of functionality. This Security characteristic has as sub characteristics: confidentiality, integrity, non-repudiation, accountability, and authenticity. Compatibility (including interoperability and co-existence) has been added as a characteristic in ISO 25010 giving rise to eight characteristics in all as against six in ISO 9126. As research efforts continue towards evolving the "silver bullet" OSS quality model [2] it is important to check the compliance of the existing models with ISO 25010 so as to identify a candidate model that can be extended towards realizing a generic OSS quality model. The aim of this paper therefore is to evaluate existing OSS quality model against the ISO 25010 standard.

The rest of this paper is structured as follows: Section 2 reviews related works. Section 3 evaluates the existing OSS quality models by classifying them into first and second-generation models and evaluating each against ISO 25010. In Section 4 a comparative study is performed between all the existing and the ISO 25010 standard. The results are discussed in detail. Section 5 concludes the paper and mentions future work.

## II. Related Works

Since the advent of the first OSS quality model in 2003 [3], a number of other models have since been derived leading to an increasing collection of OSS quality models. Very few studies however, have focused on reviewing the models that currently exist so as to come up with more generic models that can stand the test of time in the strive towards the "silver bullet" quality model for OSS [2].

In [3] a review of OSS quality models was carried out with the intent to identify each model's characteristic features, unique strengths and limitations. This study serves as a guide to those intending to use any of the models for OSS quality evaluation. The study also lays a foundation that researchers can leverage in order to improve on the models. The comparative analysis carried out in the study was between the models and not a standard. It was done based on criteria defined by the paper's authors which include: availability of published results online; origin of the model; and availability of tool support. In addition, the model only considered six OSS quality models.

The study in [1] was an elaborate one. It took into consideration software quality models in general out of which OSS quality models are just a subset. The study was carried out in order to describe the main models and their strengths as well as to identify deficiencies. The main models were classified into two namely: basic quality models and tailored quality models. ISO 25010 was classified as part of the basic quality models. Comparison between ISO 25010 and the other basic quality models showed that the ISO model was more comprehensive in terms of the number of quality characteristics that it supported and could serve as a standard. The OSS quality models on the other hand were classified under the tailored quality models. However, no comparison was made between the basic quality model (ISO 25010) and the OSS quality models. This forms the motivation for the present study.

## III. Evaluation of Open Source Software Quality Models

This section classifies the existing OSS quality models into first and second-generation models [2] and evaluates each model against ISO 25010.

### A. First Generation OSS Quality Models

*1) Open Source Maturity Model (OSMM) [6]:* It was designed as a tool that could be used to compare and decide on the most suitable open source product option for an organization based on the product's maturity. The model consists of product indicators as well as application indicators. The product indicators are categorized into four namely: Product, Integration, Use and Acceptance. The product category focuses on the product's inherent characteristics – age, selling points, developer community, human hierarchies and licensing. ISO 25010 does not include any of the characteristics in this category and so the category as a whole is excluded from the evaluation. The Integration category measures the product's modularity, adherence to standards as well as options to link the product to other products or infrastructure. Modularity is sub-characteristic of maintainability in ISO 25010 hence we mark maintainability for OSMM in Table I. Adherence to standards corresponds to compliance found under the compatibility characteristic of ISO 25010 hence compatibility is ticked in Table I. Options to link to other products or infrastructure correspond to interoperability, which is also found under the compatibility characteristic. The Use category informs on the ease with which a product can be deployed and the way in which the user is supported in the everyday use of the product. Ease of deployment corresponds to installability under the transferability characteristic in ISO 25010 and so this characteristic is marked in Table I. The way a user is supported in the everyday use of a product corresponds to helpfulness and technical accessibility under the operability characteristic of ISO 25010. Hence operability is marked in Table I. The Acceptance category tells about the market penetration of the product and the user base formed around the product. This category is out of the scope of ISO 25010 and is not considered in the evaluation. The application indicators on the other hand focus on environmental aspects that affect the product as well as the present demands of the user. The indicators include: usability, interfacing, performance, reliability, security, proven technology, vendor independence, platform independence, support, reporting, administration, advice, training, staffing and implementation. Usability is found under the quality in use category of ISO 25010 hence usability is marked in Table I. Interfacing corresponds to interoperability under the compatibility characteristic, which is already marked. Performance corresponds to performance efficiency in ISO 25010 and so it is marked in Table I. Reliability corresponds to the reliability characteristic in ISO 25010 and so it is marked in Table I. Security corresponds to security characteristic in ISO 25010 and so is marked in Table I. Proven technology corresponds to stability under the maintainability characteristic. Vendor independence and platform independence correspond to portability and adaptability under the transferability characteristic. Support corresponds to technical accessibility under operability characteristic. Training corresponds to learnability under operability characteristic Reporting, administration, advice, staffing and implementation are not in the scope of ISO 25010 and so they are excluded from the evaluation. Table I is a summary of the evaluation result.

From Table I it can be observed that OSMM addresses all the Product Quality characteristics of ISO 25010. However, it only measures usability under the Quality in Use category. A key strength of OSMM model is the fact that it can be updated on a regular basis making it flexible and adaptable to the dynamic open source domain. The update is carried out based on feedback from customers. The model however, only measures external quality. It does not address internal quality especially source code which is deemed important [7].

*2) QSOS [8]:* The purpose of this model is to qualify, select and compare free and open source in an objective, traceable and argued way. The model consists of four stages namely: Definition, Evaluation, Qualification and Selection. Definition establishes the frame of reference (which include: software families, types of licenses and types of communities) for the other steps. Evaluation involves collecting information from the open source community in order to build the identity card of the software and also build the evaluation sheet for scoring the software based on functional coverage, risks from user's perspective and risks from service provider's perspective. Qualification defines filters translating the needs and constraints related to the selection of free or open source software in a specific context (i.e. the user's context). Selection involves identifying software fulfilling user's requirements. Transition from one stage to the other is done iteratively. The model has a Free/Libre Open Source Software (FLOSS)-based license and is supported by a tool called open source selection software (O3S). With the model, objective and traceable evaluation of free and open source software can be performed. The model has however been criticized for providing a restrictive score range of between 0 and 2 during evaluation [9]. The criteria for evaluation include intrinsic durability, which is outside the scope of ISO 25010; industrialized solution, which is also outside the ISO 25010 scope; technical adaptability, which consists of modularity and by-products as sub-characteristics, and strategy, which is also outside the scope of ISO 25010. Modularity corresponds to modularity under maintainability characteristic, which is marked in Table I.

*3) Open Business Readiness Rating (Open BRR) [10]:* This model is intended to help IT managers assess which open source software would be most suitable for their needs. Open source users can also share their evaluation ratings with potential adopters, continuing the virtuous cycle and "architecture of participation" of open source. The model accelerates software assessment through a systematic approach. There are four phases involved. The first phase is a quick assessment to rule out software packages and create a shortlist of viable candidates. The second phase ranks the importance of categories or metrics. The third phase collects and processes data about an open source product while the last phase translates the data into the Business Readiness Rating. This ensures better decisions and increases confidence in the selected open source software. The model deals mostly with qualitative measures and so weights are assigned to each metric to be used for aggregating the metric scores and obtaining the score of each category. The assessment criteria for the model include: functionality, usability, quality, security, performance, scalability, architecture, support, documentation, adoption, community and professionalism. Functionality corresponds to functional suitability in ISO 25010 and is thus marked in Table I. Usability corresponds to usability a sub-characteristic of the Quality in Use model of ISO 25010 and is thus marked. Quality as a characteristic is outside the scope of ISO 25010 and is not included in the evaluation. Security corresponds to security in ISO 25010 and is so marked in Table I. Performance corresponds to performance efficiency in ISO 25010 and is so marked in Table I. Support corresponds to helpfulness and technical accessibility, which are sub-characteristics of the operability characteristic in ISO 25010, and is so marked in Table I. Scalability, architecture, documentation, adoption, community and professionalism are outside the scope of ISO 25010. They are not included in the evaluation.

*4) Sung et al. Model [11]:* This model was developed for use as a model for the selection of OSS for development purpose. It consists of four main characteristics (reusability, functionality, usability and portability) and ten sub-characteristics (suitability, security, understandability, learnability, operability adaptability, installability, co-existence, functional commonality and lawfulness). Reusability corresponds to reusability under maintainability characteristic in ISO 25010 and so is marked in Table I. Functionality corresponds to functional suitability characteristic in ISO 25010 and so it is marked in Table I. Usability corresponds to usability under the quality in use category and so is marked in Table I. Portability corresponds to portability under the transferability characteristic in ISO 25010 and so is marked in Table I. For the ten sub-characteristics, suitability again corresponds to functional suitability characteristic in ISO 25010 and is already marked. Security corresponds to security characteristic in ISO 25010 and so is marked in Table I. Understandability is outside the scope of ISO 25010 and so is not considered in the evaluation. Learnability corresponds to learnability a sub-characteristic of operability characteristic in ISO 25010 and so is marked in Table I. Operability corresponds to operability characteristic in ISO 25010, which is already marked. Adaptability corresponds to adaptability, which is a sub-characteristic of transferability, which is already marked in Table I. Installability corresponds to installability a sub-characteristic of transferability, which is already marked in Table I. Co-existence corresponds to co-existence, which is a sub-characteristic of compatibility in ISO 25010 and so is marked in Table I. Functional commonality as well as lawfulness is outside the scope of ISO 25010 and so they are excluded from the evaluation.

### *B. Second Generation OSS Quality Models*

*1) QualOSS [12*]: This model was designed to support the quality evaluation of FLOSS projects with a focus on evolvability and robustness. The product-related quality characteristics in the model include: maintainability, reliability, transferability, operability, performance, functional suitability, security, compatibility which are all characteristics in the ISO 25010 model and are thus marked in Table II. The community related quality characteristics include: maintenance capacity, sustainability, and process maturity. These are all out side the scope of ISO 25010 and so are not considered in the evaluation. Table II shows the result of the evaluation process. The model does not include any characteristic in the Quality in Use category.

*2) OMM [13]:* This model was developed as part of the deliverables of a larger project called QualiPSo [14] [15] [16]. QualiPSo – Quality Platform for Open Source Software – is one of the largest initiatives of the European Union. It defines an evaluation framework for the trustworthiness of FLOSS projects. The trustworthiness is defined in terms of product quality and considers as-is utility, exploitability in development, functionality, interoperability, reliability, performance, security, cost-effectiveness, customer satisfaction and developer quality. Utility and exploitability in development are out of the ISO 25010 scope and so they are not included in the evaluation. Functionality corresponds to functional suitability characteristic in ISO 25010 and is thus marked in Table II. Interoperability is a sub-characteristic of compatibility metric in ISO 25010 and is so marked in Table II. Reliability corresponds to reliability characteristic in ISO 25010 and is thus marked in Table II. Performance corresponds to performance efficiency characteristic in ISO 25010 and is thus marked in Table II. Security corresponds to security characteristic in ISO 25010 and is thus marked in Table II. Cost effectiveness, customer satisfaction and developer quality are out of the scope of ISO 25010.

*3) SQO-OSS [17][18]:* The model can be used for detailed quality evaluations of OSS thereby supporting decisions of whether to use or not to use. The model is hierarchical and evaluates source code and community processes, which are key quality factors of OSS. For the source code quality it measures maintainability, reliability and security. The sub-characteristics of maintainability are: analyzability, changeability, stability and testability. The sub characteristics of reliability are maturity and effectiveness. Security has no sub-characteristic. For the community quality, it measures mailing list quality, documentation quality and developer base quality. Maintainability corresponds to maintainability in ISO 25010 and so is marked in Table II. The sub-characteristics of maintainability in SQO-OSS are all contained in the ISO 25010 model. Reliability corresponds to reliability in ISO 25010 and is also marked in Table II. However, maturity a sub-characteristic is out of the scope of ISO 25010. Effective a sub-characteristic of reliability in SQO-OSS corresponds to Effectiveness in the Quality in Use model and is thus marked in Table II. Security corresponds to security characteristic in ISO 25010 and is thus marked in Table II. The community quality is outside the scope of ISO 25010 and so is excluded the evaluation.

*4) EFFORT [19]:* This is the quality model on which the EFFORT framework for evaluating the quality and functionality of OSS systems is built. The name EFFORT is an acronym for Evaluation Framework for Free/Open souRce projecTs. It supports the evaluation of product quality, community trustworthiness and product attractiveness. In order to measure product quality, the model leverages on the quality characteristics of ISO 9126. Community trustworthiness has developers, community activity, support tools, support services and documentation as sub-characteristics. Product attractiveness has functional adequacy, diffusion, cost effectiveness and legal reusability as sub-characteristics. The characteristics the product quality factor is based on ISO 9126 model therefore, functionality in ISO 9126 corresponds to functional suitability in ISO 25010 and is marked in Table II. Reliability in ISO 9126 corresponds to reliability in ISO 25010 and is marked in Table II. Usability in ISO 9126 corresponds to usability under the Quality in Use factor of ISO 25010 and so is marked in Table II. Efficiency in ISO 9126 corresponds to performance efficiency in ISO 25010 and is marked in Table II. Maintainability in ISO 9126 corresponds to maintainability in ISO 25010 and is marked in Table II. Portability in ISO 9126 corresponds to portability under the transferability characteristic in ISO 25010 and so is marked in Table II. Security was a sub-characteristic of functionality in ISO 9126 but is now a characteristic in ISO 25010. Due to this correspondence, security is marked in Table II. The strength of this model is that it considers the key quality aspects of OSS that include: the software itself, the community built round the software as well as its appeal to the user.

## IV. Comparative Study and Discussion

The key difference between the two classes of models described in the previous section is that the second-generation models provide more tools to aid the quality evaluation process compared to the first generation models. Table I shows a comparison of the first generation quality models to ISO 25010.

TABLE I. Comparison of First Generation OSS Quality Models with ISO 25010

| ISO 25010 | Quality Characteristics | OSMM | QSOS | Open BRR | Sung et al. |
|---|---|---|---|---|---|
| Product Quality | Funtional Suitability | x | | x | x |
| | Reliability | x | | | |
| | Performance efficiency | x | | x | |
| | Operability | x | | x | x |
| | Security | x | | x | x |
| | Compatibility | x | | | x |
| | Maintainability | x | x | x | x |
| | Tranferability | x | | | x |
| Quality in Use | Effectiveness | | | | |
| | Efficiency | | | | |
| | Satisfaction | | | | |
| | Safety | | | | |
| | Usability | x | | x | x |

From Table I it can be observed that maintainability characteristic of ISO 25010 is common to the first generation OSS quality models. First thing to note is that OSMM implements all the Product Quality characteristics of ISO 25010. This makes it the most comprehensive model in the first generation OSS quality models category Product-Quality-wise. The Sung et al. Model, OpenBRR and QSOS possessing seven, six and one characteristic(s) respectively follow closely behind OSMM. Another point to note is that the first generation OSS quality models with the exception of QSOS evaluate usability - a subset of the Quality in Use characteristics - of OSS. In addition, none of the first generation models measure effectiveness, efficiency, satisfaction and safety. Table II shows the comparison of the second-generation quality models to ISO 25010.

TABLE II. COMPARISON OF SECOND GENERATION OSS QUALITY MODELS WITH ISO 25010

| ISO 25010 | Quality Characteristics | Qual OSS | OMM | SQO-OSS | EFFORT model |
|---|---|---|---|---|---|
| Product Quality | Funtional Suitability | x | x |  | x |
|  | Reliability | x | x | x | x |
|  | Performance efficiency | x | x |  | x |
|  | Operability | x |  |  |  |
|  | Security | x | x | x |  |
|  | Compatibility | x | x |  |  |
|  | Maintainability | x |  | x | x |
|  | Tranferability | x |  |  | x |
| Quality in Use | Effectiveness |  |  | x |  |
|  | Efficiency |  |  |  |  |
|  | Satisfaction |  |  |  |  |
|  | Safety |  |  |  |  |
|  | Usability |  |  |  | x |

From Table II it can be observed that reliability characteristic of ISO 25010 is common to the second-generation OSS quality models. QualOSS measures all the characteristics given in ISO 25010 Product Quality and none of those for Quality in Use. EFFORT model measures six quality attributes while OMM measures five. OMM measures all attributes in Product Quality category except for operability, maintainability and transferability. It measures none of the Quality in Use characteristics. EFFORT model measures five characteristics of ISO 25010 Product Quality except for Operability, Security, and Compatibility. It also measures usability of OSS – which is a subset of Quality in Use. Among the second-generation OSS quality models, EFFORT model is balanced since it takes Product Quality and Quality of Use into consideration.

It should be noted that all the models considered in both tables also evaluate quality of OSS by considering the OSS community. However, the ISO standard does not make any provision for this. Going from this point therefore, EFFORT model can serve as a basis for developing a generic OSS quality model because it is a second-generation OSS quality model providing better tool support and methodology for evaluation; it considers three key perspectives of OSS quality namely: the product, its community and users' perception about it; the model has also been applied to evaluate OSS in the customer relationship management (CRM) domain [7] as well as in the enterprise resource planning (ERP) domain [19].

## V. CONCLUSION AND FUTURE WORK

This paper set out to conduct a comparative study between OSS quality models and the ISO 25010 model. The motivation stemmed from the gap identified in literature. Previous studies did not address this issue. In order to carry out a detail study, the existing OSS models were split into two generations. It was observed that the models in the second-generation provided better tool support to aid the evaluation process compared to the first generation models. The comparative study revealed that OSMM was the most comprehensive model in the first generation category satisfying all eight factors under Product Quality and usability under the Quality in Use factors. EFFORT model was preferred above the QualOSS model in the second category because it considered both the Product Quality and Quality in Use aspects. EFFORT model was also preferred above OSMM because it is a second-generation OSS quality model; and has been applied to evaluate OSS in the CRM and ERP domains. The conclusion from the study therefore is that EFFORT model can be extended in order to derive a generic OSS quality model. For future work, the EFFORT quality model will be extended in order to derive a generic OSS quality model that is fully based on the ISO 25010 standard.

## REFERENCES

[1] J.P. Miguel, D. Mauricio, and G. Rodríguez, A Review of Software Quality Models for the Evaluation of Software Products, International Journal of Software Engineering & Applications, 5(6), 2014, pp. 31-53.

[2] K. Haaland, A.K. Groven, N. Regnesentral, R. Glott, A. Tannenberg, and A.S. FreeCode, Free/Libre Open Source Quality Models-A comparison between two approaches, In 4th FLOS International Workshop on Free/Libre/Open Source Software, 2010, pp. 1-17.

[3] A. Adewumi, S. Misra, and N. Omoregbe, "A Review of Models for Evaluating Quality in Open Source Software," 2013 International Conference on Electronic Engineering and Computer Science, IERI Procedia, vol. 4, 2013, pp. 88 – 92.

[4] ISO/IEC 25010, "Systems and Software Engineering—Systems and Software Product Quality Requirements and Evaluation (SQuaRE)—System and Software Quality Models," International Organization for Standardization, Geneva, 2010.

[5] ISO/IEC 25010 "Systems and Software Engineering—Systems and Software Product Quality Requirements and Evaluation (SQuaRE)—System and Software Quality Models," International Organization for Standardization, Geneva, 2010.

[6] F. Duijnhouwer, and C. Widdows, "Open Source Maturity Model, Capgemini Expert Letter", 2003, Available at: http://jose-manuel.me/thesis/references/GB_Expert_Letter_Open_Source_Maturity_Model_1.5.3.pdf Date accessed: 5th January, 2015

[7] L. Aversano, and M. Tortorella, Applying EFFORT for evaluating CRM open source systems, In Product-Focused Software Process Improvement, Springer Berlin Heidelberg, 2011, pp. 202-216.

[8] Atos, Method for Qualification and Selection of Open Source Software (QSOS) version 1.6, 2006, Available at: http://master.libresoft.es/sites/default/files/Materiales_MSWL_2010_2011/Project%20Evaluation/materiales/qsos-1.6-en.pdf Date accessed: 5th January, 2015

[9] J.C. Deprez, and S. Alexandre, Comparing assessment methodologies for free/open source software: OpenBRR and QSOS, in: 9th International Conference on Product-Focused Software Process Improvement (PROFES'08), Springer Verlag, 2008, pp. 189–203.

[10] A.I. Wasserman, M. Pal, and C. Chan, Business Readiness Rating for Open Source, Proceedings of the EFOSS Workshop, Como, Italy, 2006, Available at: http://docencia.etsit.urjc.es/moodle/file.php/125/OpenBRR_Whitepaper.pdf Date accessed: 5th January, 2015

[11] W.J. Sung, J.H. Kim, and S.Y. Rhew, A Quality Model for Open Source Software Selection, In Sixth International Conference on Advanced Language Processing and Web Information Technology (ALPIT 2007), 2007, pp. 515-519.

[12] M. Soto, and M. Ciolkowski, The QualOSS open source assessment model measuring the performance of open source communities. In Proceedings of the 3rd ESEM, 2009, pp. 498–501.

[13] E. Petrinja, R. Nambakam, and A. Sillitti, Introducing the Open Source Maturity Model. In Proceedings of the 2009 ICSE Workshop on Emerging Trends in Free/Libre/Open Source Software Research and Development 2009, pp. 37-41.

[14] V. Del Bianco, L. Lavazza, S. Morasca, and D. Taibi, and D. Tosi, The QualiSPo approach to OSS product quality evaluation, in: 3rd International Workshop on Emerging Trends in Free/Libre/Open Source Software Research and Development, ACM, New York, 2010a, pp. 23–28.

[15] V. Del Bianco, L. Lavazza, S. Morasca, and D. Taibi, and D. Tosi, An investigation of the users' perception of OSS quality, in: 6th International Conference on Open Source Systems, Springer Verlag, 2010b, pp. 15–28.

[16] V. Del Bianco, L. Lavazza, S. Morasca, and D. Taibi, A Survey on Open Source Software Trustworthiness, IEEE Software 28 (5), 2011, pp. 67–75.

[17] I. Samoladas, G. Gousios, D. Spinellis, and I. Stamelos, The SQO-OSS quality model: measurement based open source software evaluation. In Open source development, communities and quality, Springer, 2008, pp. 237-248. .

[18] D. Spinellis, G. Gousios, V. Karakoidas, P. Louridas, P.J. Adams, I. Samoladas, I., and I. Stamelos, Evaluating the quality of open source software. Electronic Notes in Theoretical Computer Science, 233, 2009, pp. 5-28.

[19] L. Aversano, and M. Tortorella, Quality Evaluation of FLOSS Projects: Application to ERP Systems, Information and Software Technology, 55(7), 2013, pp. 1260-1276.