Review on Quality Models for Open Source Software and its reflection on Social Coding

Ragia AboElFadl, Akram Salah, Amr Kamel,
Computer Science Dept.
Cairo University
Giza, Cairo 12613, Egypt

***Abstract*— Social Coding Sites (SCSs) are social media services for sharing software development projects on the Web, many open source projects are currently being developed on SCSs. Assessing the quality is a crucial element for better selection of a specific project serving people requirements or needs. In this paper, we reviewed existing traditional models which evolved prior the evolution of open source software as well as open source quality models. We evaluated the selected models according to their reflection with respect to social coding project success factors: sociality, popularity, activity and supportability. Eight models were included in our research as we only selected models that introduces explicit metrics well defined for measuring, neither a process nor a generic methodology. Based on our selection criteria, a summary of the findings we obtained is that existing models doesn't fully consider or cover social factors for open source software evaluation hence there is a need for a model to measure the maturity / quality of open source projects from social factors perspective. We have also evaluated the existing models against a selected open source project hosted on social coding GitHub to assess each model applicability. Some of the measurements from the existing models were not applicable for evaluation.**

***Index Terms*—open source, quality models, social coding, OSS, maturity models.**

I. Introduction

Social Coding Sites (SCSs) are social media services for sharing software development projects on the web. Many open source software (OSS) are currently being developed on SCSs for all categories of software such as end-user applications, database and middleware technologies, web/mobile frameworks and libraries. One of the characteristics of SCSs is that they provide a platform on social networks that encourages collaboration between developers with the same interests and purpose.

Maturity is one of the key quality factors that people consider when choosing among similar open source projects. It can be affected by community or social factors like the number of users, the longevity of the project, etc. as well as the quality of the product in terms of code quality, coding standards and architecture and other factors. All of these factors make it hard and risky to assess and further to select the most suitable open source project to use in order to fulfill the needed requirements.

The aim of this paper is to review the existing open source quality models and evaluate their reflection on social coding. The paper is organized as follows: section 2 presents the previous reviews on existing models and identifies the characteristics of a successful open source project in social coding. Section 3 describes the research method for evaluating OSS models. Then section 4 demonstrates an overview on the existing models. And finally in section 5 discuss the evaluation method used in the study.

II. Literature Review

In this section, we first provide an overview on the common traditional quality models and their main characteristics, then we present existing reviews on open source maturity models, and finally we introduce other researches that identify what makes an open source project successful.

1. Traditional Quality Models:

Quality Models in software engineering domain started to evolve long time ago. McCall, Boehm, FURPS, Dromey, ISO 9126, ISO 25010 [1] introduced software quality models prior to the evolution of OSS that assess and evaluate the product quality characteristics. The common quality characteristics for these models are: portability, reliability, efficiency and usability [1]. Along with these quality models, another aspect of quality evolved for the software development process, the most common assessment methodology in the industry is the Capability Maturity Model for Integration (CMMI) [2]. CMMI is a set of related best practices derived from industry leaders that relate to product engineering and software development. It was developed for organizations to assess and enhance their internal processes.

2. Open Source Maturity Models:

Open source maturity models started to evolve in 2003. Most of open source models are derived from traditional quality models: IOS 9126 or CMMI [3].

We identified from literature few researches that have reviewed existing quality models. Adewole Adewumi [3] identified the weaknesses, strengths and limitations of: CapGemini Open Source Model, Qualification and Selection of Open Source Software Model (QSOS), Open Business Readiness Rating Model (OpenBRR), Software Quality Observatory for Open Source Software Model (SQO-OSS), Open Source Maturity Model OMM) and Quality of Open

IRJECE

This is an open access article under the CC BY-SA 4.0 license (https://creativecommons.org/licenses/by-sa/4.0/)

Source Software Model (QualOSS). A comparative analysis was done based on answering 3 questions as follows: "is there available published results for evaluating each model?", "is there an available tool to support the model?" and "is the model based on another model?".

Etiel Petrinja [4] conducted an experiment to compare QSOS, OpenBRR and OMM on 2 similar open source products. The purpose of the study was mainly to identify which model better characterizes specific aspects to open source projects. Based on the study, the author concluded that OMM model has the strongest characteristics for assessing an open source project.

Jose P. Miguel [1] classified the models into either basic or tailored. Open source maturity models are classified as tailored given that they originate from the traditional basic models. The author then compared the basic models which mainly focus on the products quality characteristics. The comparison was done against the most common quality attributes: accuracy, adaptability, analyzability, attractiveness, changeability, correctness, efficiency, flexibility, functionality, human engineering, installability, integrity, interoperability, maintainability, maturity, modifiability, operability, performance, portability, reliability, reusability, stability, suitability, supportability, testability, transferability, understandability and usability.

Adewole Adewumi [5] introduced a systematic review on the existing open source quality assessment models. The research did not only include models, but also included analytical work and data mining related publications that evaluate OSS. The author classified the existing models into 5 categories namely: single-attribute, rounded category, community-only attribute, non-community attribute and non-quality in use models. He then applied a comparative analysis between the selected OSS quality models and the ISO 25010 model. The comparison included community related characteristics: maintenance capacity, sustainability and process maturity, and further extended to how they compare with the OSS quality models.

3. Open Source Projects Success Factors:

Recent studies introduced other measurements for identifying the success factors of an open source project on social coding. SCSs are social media services for sharing software development projects on the web. Many open source projects are currently being developed on SCSs. One of the characteristics of SCSs is that they provide a platform on social networks that encourages collaboration between developers with the same interests and purpose.

Collaborative Innovation Networks (COINs) [6] in recent years have proved their ability to produce high quality software. GitHub [7] is one of the leading examples on COINs. It is a web-based community hosting service for software development projects. According to the statistics published by GitHub [8] in 2017, the community hosts 24M distinct users, 1.5M organizations and 67M projects.

Researchers analyzed GitHub data for defining factors of a successful open source project. Jarczyk [9] presented 2 measures of quality for GitHub OSS projects: one is based on project popularity and the other is identified as project supportability which represents how fast the project's team solves issues reported by users.

Yuya Yoshikawa [10] identified the characteristics of a successful open source project where success was defined by 3 main indices: activity, popularity and sociality. The activity is calculated based on the update frequency of content developed by each project. The popularity and sociality are evaluated by the number of bookmarks and the number of updates from the external developers to quantify these indices. The following metrics were used:

Number of commits (Commit) represents the number of times codes and documents are added to the repository of the project. Projects with higher Commit are more likely to develop quickly and effectively. The author used it for measuring the activity of a project.

Number of stars (Star) represents the number of developers who marked the project with a star. Stars are generally used to monitor the behavior of a project or watch the project later. The author used it to measure the popularity of a project.

Number of pull requests (PullReq) represents the number of times the external developers requested that their codes be accepted. Since a higher PullReq indicates that the project is frequently discussed with external developers, the author used it to indicate the sociality of a project.

Despite the fact that traditional quality models focus extensively on evaluating the quality characteristics of a product, they unfortunately do not take into consideration quality attributes related to OSS. On another note, open source quality models which originally evolved to assess OSS by introducing new quality attributes, none of these models evaluate important criteria, in terms of every quality characteristic under product quality, quality in use, and community related quality characteristics [5]. According to Muhammad A. Babar, there is still little or no adoption of quality assessment models in practice despite that many of them exists [11].

Social factors are important to assess the quality of open source projects, however they are not given enough attention in neither open source maturity models nor traditional quality models. Hence, these models need to be evaluated from social factors perspective.

III. Research Method

This section outlines the research method used to include or exclude any given model in the study, as well as the research question.

- Reviewed literature starting from 2003 till present in order to identify the models that had been proposed since the first open source quality model emerged in 2003 as pointed out by Adewumi [3].

IRJECE

This is an open access article under the CC BY-SA 4.0 license (https://creativecommons.org/licenses/by-sa/4.0/)

- Include only publications on models that have specific explicit elements or factors to evaluate the quality of OSS. However, publications introducing generic methodology or process for assessing quality were excluded such as QSOS[12] which was excluded since its represents a generic methodology.
- Evaluate the selected models from the social factors perspective.

Following this methodology, only 8 models matched the predetermined inclusion criteria and hence included in our assessment.

.

IV. MATURITY MODELS FOR OPEN SOURCE

In this section, we explain the included models. For each model, we give an overview on how it evaluates the quality of OSS. The reflection of the model on social coding success factors is discussed in the next section.

1. Open Source Maturity Model (OSMM) from CapGemini :

CapGemini [1] is based on the maturity of the product and is set according to maturity indicators. These indicators are grouped in product and application indicators. For the final evaluation each of the sub indicators is given a value between 1 and 5 giving a total score. Indicators are product, integration, use, acceptance and application indicators are: usability, interfacing, performance, reliability, security, support, administration, reporting, training. The model didn't define explicit quality measures for evaluating the quality characteristics.

2. Open Source Maturity Model (OSMM) from Navica:

Navica OSMM [13] assess mainly maturity based on these 6 categories (Product software, Support, Documentation, Training, Product integrations and Professional services), each one having a specific weight. Each category is subjectively evaluated to a given score from 1 to 5 and the summation of these scores reflects the maturity of the OSS.

Navica OSMM didn't define explicit definition for each category, its subjective to the evaluator to define the key important element that resides in each category.

3. Software Quality Observatory for Open software (SQO-OSS):

SQO-OSS [14] is a hierarchical model that evaluates the source code and the community process allowing automatic calculation of metrics, it considers only the community factors that can be automatically measured. The author introduced a direct metric for each quality characteristic in the model, such that maturity is measured by Number of open critical bugs in last 6 months, Community quality was measured by: mailing list quality, Documentation quality and developer base quality.

4. Open Business Readiness Rating (OpenBRR):

OpenBRR [15] is a Rating framework influenced by the Cap-Gemini and ISO 9126 Models, first step is to have a list of similar products to be evaluated in place, then every component is evaluated with respect to a set of indicator Selected, it consists of 12 categories: functionality, usability, quality, security, performance, scalability, architecture, support, documentation, adoption, community, professionalism, each category can be measured by set of metrics, the author recommend as a best practice to normalize these metrics to have a common ground when evaluating them, proving a final score computed from 1 to 5, metrics can be qualitative so it may require subjective evaluation. We assessed the model based on the introduced best practices factors in the publication.

5. Open Business Quality Rating (Open BQR):

Open BQR [16] is derived from QSOS and BRR, it mainly a process that has 3 phases, the first phase is a quick assessment filter to Analyze the external qualities, analyze of internal quality, product support over time, existence of required functionality, second phase is data collection and processing, finally is the data translation.

6. A study on maturity model of open source software community to estimate the quality of products:

The model [17] introduced in the research is mainly derived from the CMMI, the author interpreted that good community gives indication of a good quality product, the model consists of 5 levels of maturity:

I. Initial: the project is just started. Community is not formed at this stage.

II. Managed: Development is carried out by a project. At this stage product are released.

III. Defined: a community defines development processes and rules to manage their projects and products. According to the definitions, projects are carried out. The projects continuously release products.

IV. Quantitavily managed: a community quantitatively measures the status of projects and products. For example, the community determines the release of products by checking the quality of them.

V. Optimizing: a community quantitatively measures the status of projects and it continuously improves the process.

The model only assesses the community characteristics, having a successful community doesn't guarantee having a successful project, having a mature process in development doesn't guarantee that this is the best project for selection among similar ones. Although the model only assesses the community characteristics yet it didn't consider the sociality, popularity, supportability and activity. This means that this model partially covers all community factors, however it is lacking defining what to measure and how to measure them.

7. Quality of Open Source Software Model (QualOSS):

QualOSS [18] mainly assesses 8 product quality characteristics and 3 community quality characteristics.

IRJECE

This is an open access article under the CC BY-SA 4.0 license (https://creativecommons.org/licenses/by-sa/4.0/)

Maintain-ability, reliability, portability, operability, performance, functional suitability, security and compatibility as quality characteristics, maintenance capacity, sustainability and process maturity as community quality related characteristics. QualOSS defined explicit measures for community related characteristics, activity factor is covered in the maintenance capacity element as maintenance capacity focuses mainly on the degree to which the contributors increase by time.

8. Open Source Maturity Model (OMM):

OMM [19] is mainly derived from CMMI, it was designed on the Goal Question Metric (GQM) approach (Basili, 1992) it defines key elements that allow users to trust the quality of OSS, it mainly comprises trustworthy elements Based on these TWEs the model define goals that have to be achieved. OMM is based on these 12 TWE: Product Documentation (PDOC), Popularity of the SW Product (REP), Use of Established and Widespread Standards (STD), Availability and Use of a (product) Road map (RDMP), Quality of Test Plan (QTP), Relationship between Stakeholders (Users, Developers etc) (STK), Licenses (LCS), Technical Environment (Tools, OS, Programming Language, Dev Environment.) (ENV), Number of Commits and Bug Reports (DFCT), Maintainability and Stability (MST), Contribution to FLOSS Product from SW Companies (CONT), Results of Assessment of the Product by Third Party Companies (RASM). The measure to quantify the TWE elements in the model are not standard, people can perceive these TWE by different measures.

V. Discussion

In this section, we discuss the selected models from perspective of social coding project success factors, namely: sociality, activity, popularity and supportability. We summarize the results in "Table 1".

1. Sociality:

Qual-OSS addressed sociality indirectly. The model defined the sustainability element as having external contributors and internal contributors. This relates to sociality factor as it considers involvement of external contributors. However, the relationship is not straightforward as the objective of sustainability and sociality is different although they tend to use the same measures. The other selected models did not define elements that tackled sociality.

2. Popularity:

OMM defined Software Product Popularity (REP) as an essential element for measuring maturity of an OSS. The other models considered popularity as growth of a software in terms of developers and users. This lacks measure of trust and user satisfaction. According to this observation, we decided that none of the models except OMM covers popularity, as trust and user satisfaction are vital aspects of it.

3. Activity:

OMM cover activity factor through Number of Commits and Bug Reports (DFCT) element that resembles software growth. OpenBRR also covers the activity factor in the community category defined in the model by assessing how active the users are. SQO-OSS defined the developer base quality element as measuring the growth of active users and rate of developer intake/turnover. This element partiality relates to the Activity factor. As growth of active users implicitly means software growth. However, registered users with rare contributions should not be considered equal to registered users with frequent contributions. We considered SQO-OSS partially covering Activity. The other models did not define elements that consider Activity.

4. Supportability:

Cap-Gemini OSMM, Navica OSMM and OpenBRR cover supportability through support category in their models. Open BQR cover the assessment of supportability factor in the product support over time element. OMM partially covers supportability factor by defining STK element as the relationship between stakeholders and users. This element may be interpreted as a reflection of supportability. SQO-OSS and QualOSS models did not address supportability.

5. Model Evaluation in social coding environment:

To assess model relevance in social coding environment, we evaluated projects hosted on GitHub to better understand the models applicability on open source projects hosted on social coding environment. Aurelia framework[20] was selected as an example of a free OSS project hosted on social coding. We evaluated the project from the perspective of the included models. For Cap-Gemini, we were not able to evaluate the administration, training, staffing, selling point and human hierarchy factors in the social coding context as these factors are related to commercial OSS. When evaluating the project against Navica OSMM, we were not able to evaluate the training element as nowadays people mainly rely on either documentation and tutorials, or collaborative efforts of individuals in the same network. For the professional services element, Navica has no clear definition on how we can measure this element. Hence, we could not relate it to any of the existing measurements on GitHub. For SQO-OSS, we were not able to measure the mailing list for Aurelia framework as the related technology is now somehow irrelevant to social coding and other similar modern technologies have recently evolved for example: watchers. As for OpenBRR, we were not able to measure professionalism element. Professionalism in openBRR refers to the level of professionalism in the development process and the model suggested measuring it by assessing the project driver as well as the difficulty to contribute and collaborate with core developer team. Hence, these measure are not applicable in social coding environment. The elements of Qual-OSS and OpenBQR models are

IRJECE

This is an open access article under the CC BY-SA 4.0 license (https://creativecommons.org/licenses/by-sa/4.0/)

applicable on social coding for Aurelia framework, however the elements are not enough for evaluation and that was evident in the evaluation conducted with respect to social factors. For OMM, the RDMP and QTP were not applicable due to the fact that social coding open source projects do not have a rigid test plan or defined roadmap in place. Finally, the Kuwata's model was not applicable for assessment on Aurelia framework due to its generic context.

TABLE I. SOCIAL CODING FACTORS COVERAGE IN OPEN SOURCE QUALITY MODELS

Model	Sociality	Popularity	Activity	Supportability	Elements NA for Social Coding
Cap-Gemini	NC	NC	NC	C	Administration, training, staffing, selling point, human hierarchy
Navica	NC	NC	NC	C	Training and professional services
SQO-OSS	NC	NC	PC	NC	Mailing list
OpenBRR	NC	NC	C	C	Professionalism
OpenBQR	NC	NC	NC	C	-
Kuwata, 2014	PC	PC	PC	PC	NA
QualOSS	PC	NC	PC	NC	-
OMM	NC	C	C	PC	RDMP, QTP, RASM

a. NC refers to: not covered, PC refers to: partially covered and C refers to: covered

VI. CONCLUSION:

In this study we identified publications including models to assess the quality of open source projects. we identified from literature what makes an open source project hosted on social coding sites successful. we evaluated the models that are selected according to the social factors: sociality, popularity, activity and supportability. We concluded that social factors are not fully covered in the selected models, hence a need for new model to consider social factors for assessing open source project quality.

REFERENCES

[1] J. P. Miguel, D. Mauricio, and G. Rodríguez, "A Review of Software Quality Models for the Evaluation of Software Products," *Int. J. Softw. Eng. Appl.*, vol. 5, no. 6, pp. 31–53, 2014.

[2] E. Amoroso, C. Taylor, J. Watson, and J. Weiss, "A Process-Oriented Methodology for Assessing and Improving Software Trustworthiness 2 Definition of Software Trustworthiness Trust Principles," *Proc. 2nd ACM Conf. Comput. Commun. Secur.*, pp. 39–50, 1994.

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

[4] J. Flaherty and E. Petrinja, "Comparing OpenBRR,QSOS and OMM Assessment Models," *Coach. (Third Ed.*, pp. 59–82, 2010.

[5] A. Adewumi, S. Misra, N. Omoregbe, B. Crawford, and R. Soto, "A systematic literature review of open source software quality assessment models," *Springerplus*, vol. 5, no. 1, p. 1936, 2016.

[6] L. Dabbish, C. Stuart, J. Tsay, and J. Herbsleb, "Social coding in GitHub: transparency and collaboration in an open software repository," *Proc. ACM 2012 Conf. Comput. Support. Coop. Work*, pp. 1277–1286, 2012.

[7] "GitHub," 2017. [Online]. Available: https://github.com.

[8] "GitHub statistics," 2017. [Online]. Available: https://octoverse.github.com.

[9] O. Jarczyk, B. Gruszka, S. Jaroszewicz, L. Bukowski, and A. Wierzbicki, "GitHub Projects. Quality Analysis of Open-Source Software," in *Social Informatics: 6th International Conference, SocInfo 2014, Barcelona, Spain, November 11-13, 2014. Proceedings*, L. M. Aiello and D. McFarland, Eds. Cham: Springer International Publishing, 2014, pp. 80–94.

[10] Y. Yoshikawa, T. Iwata, and H. Sawada, "Collaboration on Social Media: Analyzing Successful Projects on Social Coding," *arXiv Prepr. arXiv1408.6012*, p. 10, 2014.

[11] K.-J. K. Stol, M. A. Babar, and M. Ali Babar, "Challenges in using open source software in product development: a review of the literature," *Proc. 3rd Int. Work. Emerg. Trends Free. Source Softw. Res. Dev.*, pp. 17–22, 2010.

[12] "Method for Qualification and Selection of Open Source Software (QSOS)." [Online]. Available: www.qsos.org.

[13] James A J Wilson, "Open Source Maturity Model," 2010. [Online]. Available: http://oss-watch.ac.uk/resources/archived/osmm.

[14] I. Samoladas and G. Gousios, "The SQO-OSS quality model: measurement based open source software evaluation," *Open source Dev. …*, vol. 275, pp. 237–248, 2008.

[15] P. O. Standard, F. Assessment, and O. S. Software, "Business Readiness Rating for Open Source," *Access*, pp. 1–22, 2005.

[16] D. Taibi, L. Lavazza, and S. Morasca, "OpenBQR: A framework for the assessment of OSS," *IFIP Int. Fed. Inf. Process.*, vol. 234, pp. 173–186, 2007.

[17] Y. Kuwata, K. Takeda, and H. Miura, "A study on maturity model of open source software community to estimate the

IRJECE

This is an open access article under the CC BY-SA 4.0 license (https://creativecommons.org/licenses/by-sa/4.0/)

quality of products," *Procedia Comput. Sci.*, vol. 35, no. C, pp. 1711–1717, 2014.

[18] M. Soto and M. Ciolkowski, "The QualOSS open source assessment model measuring the performance of open source communities," *Empir. Softw. Eng. Meas. 2009. ESEM 2009. 3rd Int. Symp.*, pp. 498–501, 2009.

[19] E. Petrinja, R. Nambakam, and A. Sillitti, "Introducing the opensource maturity model," *Proc. 2009 ICSE Work. Emerg. Trends Free. Source Softw. Res. Dev. FLOSS 2009*, pp. 37–41, 2009.

[20] "Aurelia Web Framework." [Online]. Available: https://github.com/aurelia/framework.

.

IRJECE

This is an open access article under the CC BY-SA 4.0 license (https://creativecommons.org/licenses/by-sa/4.0/)