

# Comparing OpenBRR, QSOS, and OMM Assessment Models

Etiel Petrinja, Alberto Sillitti, and Giancarlo Succi  
Free University of Bolzano, Italy

## Abstract

The objective of this study was to investigate the quality and usability of three Free/Libre Open Source Software assessment models: the Open Business Readiness Rating (OpenBRR), the Qualification and Selection of Open Source software (QSOS), and the QualiPSo OpenSource Maturity Model (OMM). The study identified the positive and negative aspects of each of them. The models were used to assess two Free/Libre Open Source Software projects: Firefox and Chrome (Chromium). The study is based on a set of controlled experiments in which the participants performed the assessment using only one model each. The model used and the Free/Libre Open Source Software project assessed were randomly assigned to the participants. The experiment was conducted in a controlled environment with defined tasks to be performed in a given time interval. The results revealed that the three models provided comparable assessments for the two assessed projects. The main conclusion was that all the three models contain some questions and proposed answers that are not clear to the assessors, therefore should be rewritten or explained better. The critical aspects of each model were: Functionality and Quality for OpenBRR; Adoption, Administration/Monitoring, Copyright owners, and Browser for QSOS ; and Quality of the Test Plan, and the Technical Environment for OMM. Participants perceived the quality and usability of the three models of comparable level.

**Keywords:** FLOSS Assessment Model, Quality Criteria, Software Quality, FLOSS Development Process

## 1 Introduction

The quality of Free/Libre Open Source Software (FLOSS) products is affected by many variables and it varies strongly in different products. Often, the adoption of a product is affected by the reputation of the producer rather than the real quality of the product itself. However, different indicators can provide hints on the quality of a FLOSS project, for example: the number of users, the longevity of the project, the documentation available on-line, etc. The list of possible indicators is limitless and besides the most well-known (number of product downloads, number of bugs reported, etc.) there are many others that can have different interpretations. Therefore, it is important to have a structured set of criteria to use to assess the quality of a FLOSS project. The most well-known set of criteria used to assess the quality of software development (usually Closed Source) is part of the CMMI model [1]. However, additional sets have been proposed in the last few years targeting FLOSS. Such models include:

- Open Source Maturity Model (OSMM) from Cap Gemini (2003) [3]
- Open Source Maturity Model (OSMM) from Navica (2004) [6]

- Methodology of Qualification and Selection of Open Source software (QSOS) (2004) [2]
- Open Business Readiness Rating (OpenBRR) (2005) [10]
- Open Business Quality Rating (Open BQR) (2007) [9]
- QualiPSo OpenSource Maturity Model (OMM) (2008) [7, 11]

The large plethora of available models witness the interest and the need of systematic approaches for the assessment of the quality of FLOSS projects.

The proposed assessment models provide a selected set of criteria with their interpretation and the description of how to use them. Besides the few mentioned criteria, there are several more indicators for the quality of the code, for the functionality, the usability, the testability, the documentation, the development process followed. Moreover, there are several ways to measure such characteristics. Therefore, it is essential to include in the assessment model a consistent subset of metrics that can be used for the assessment since not all of them can be used in all the cases. The proposers of a model have to take in consideration also different use cases for their model: a FLOSS developer, a FLOSS user, and/or a FLOSS integrator. All of them will probably have different expectations about the product and the development process. For these reasons, an assessment model must be flexible and be able to adapt to different use cases. An important aspect of the criteria included in the assessment model is the names of the criteria themselves and the wording of the related questions that are used to detail them.

Another important aspect in the evaluation of the quality of FLOSS is the development processes followed. Our opinion is that it is necessary to take in consideration both aspects of FLOSS: product and process. For example, the maintainability of the product is affected by both. FLOSS integrators may be interested in the documentation produced and if it is easy to use parts of it inside their other products. For such reasons, they will be interested in the process followed to develop the FLOSS product. In this case, the measurement of the FLOSS development process of the FLOSS project is important. Available FLOSS assessment models contain some aspects of the final product and some aspects of the development process. However, most of the models are focused on the assessment of the final product. Only the OMM model covers more in details the FLOSS development process, resembling to some extent the approach adopted in the CMMI. Nevertheless this difference of focus, we identified many commonalities between the analysed models, and we think that a comparison of three of them is reasonable.

We conducted this research comparing three similar models, partially to evaluate the OMM model that we developed but mainly to see how it is perceived by users in comparison with the other two models. Moreover, some of the results of this research related to OMM were useful for validation purposes of the model and its future improvement.

The research offers also a use case demonstration of the other FLOSS assessment models. We think that it is essential to verify how the proposed

models can be used concretely and what are the perceptions of people and their confidence in the results obtained by using different models.

In the available literature, there are no comparisons of different use cases of available FLOSS assessment models. This research aims at (partially) filling this gap presenting some empirical data about the comparison of different models. We expect significant differences in the quality perception of models by users, however we must be aware that this difference depend also on the use cases adopted inside the experimentation. Some models are perceived better in the area of FLOSS communities, others are preferred by developers, and others by users.

This paper is structured as follows. After this first introductory section, we briefly present some related work. In the third section we describe our research design presenting in details the experimentation performed. The fourth section is the main part of the paper and presents research results. Afterwards, we present some threats to validity of the research conducted, and, finally, we present our conclusions.

## **2 Related work**

The OpenBRR and the QSOS models were partially validated by their developers and used in a small number of use cases with results available on their web portals [2, 10]. However, there are no empirical evaluations of the validity of the two models. Moreover, the number of use cases is limited and we noticed a quite steady number of reported FLOSS projects assessments on the web portals of the two models.

We conducted an initial validation of the OMM model as part of the QualiPSo project [8]. We involved all the partners of the project that are interested in the future use of OMM. The description of the initial validation process is available in reports of the project [8]. Inside those documents are available also results of the research presented in this paper with additional content that we were not able to present here due to space limitations. A key outcome of the research presented by authors lists the actions necessary to improve the OMM model. Such reports contains also an evaluation of critical elements identified mainly inside the OMM model but also inside the OpenBRR and the QSOS models.

In our knowledge, only a few researches have been published analysing and comparing available FLOSS assessment models. One of them was conducted by Deprez and Simons that compared the OpenBRR and the QSOS models [12]. They have done a rigorous comparison of both models and they identified advantages and disadvantages of both. The main difference between their approach and the one proposed in this paper is that we wanted to use the models with real FLOSS projects and try to find out what are the problems encountered by participants during the assessment process and collect their subjective perceptions related to the quality and usability of the models. Deprez and Simons proposed a detailed conceptual comparison of elements of the two models without conducting a real use case.

### 3 Research design

Our plan was to conduct a controlled experiment [5, 4]. We managed to satisfy many requirements for a controlled experiment as: randomization of participants, and testing specimens, and the set up of a controlled environment, the detailed planning of the experimentation process, and others. However, we were not able to involve different types of participants, for example professional programmers. This can be a problem for the generalization of results. Anyway, we took different actions to mitigate these and few other threats to validity of our research and we present key one in section 5. A detailed description of the scope of the research and the methodology used are presented in the following subsections.

#### 3.1 Scope

The research included three FLOSS assessment models: OpenBRR, QSOS, and OMM. For the research we used all questions of the three models unifying their presentation and structure. We did not use the on-line questionnaires provided by the two methodologies. We carefully, taking care not to lose any details of the two models, copied all questions in a uniform structure, with a comparable level of details. For the experimentation we choose two well known FLOSS projects: Firefox, and Chrome (Chromium). Google Chrome is not a FLOSS project but its development is tightly related to the Google Chromium FLOSS project. In this paper we will use generically the name Chrome. We assessed two FLOSS projects that are providing similar functionality, namely a web browser. Therefore, we were able to compare specific final FLOSS products and FLOSS development process characteristics.

#### 3.2 Methodology

We planned to conduct a well structured and executed experimentation process according to the operational definition of the experimentation process stated in one of the most frequently cited papers about this subject [5]:

Controlled experiment in software engineering is: “A randomized experiment or a quasi-experiment in which individuals or teams (the experimental units) conduct one or more software engineering tasks for the sake of comparing different populations, processes, methods, techniques, languages, or tools (the treatments)”.

From the definition we see that an important aspect of a controlled experiment is randomization of:

- individuals participating to the experiment,
- the tasks that they will have to perform, and
- all the treatments that are included in the experiment.

We randomized most of the experimentation components. After we decided to use three assessment models (OpenBRR, QSOS, and OMM) we distributed them randomly between all participants. The only constraint we imposed was that the number of users using each model was the same. For the experimentation we

decided to use two FLOSS projects: Firefox and Chrome. Partially we choose these two projects because during the previous two years most of our students were involved in university projects that used Firefox or Chrome as source of code, and other type of project data. We expected most of our participants will be therefore at least aware of the two projects. We distributed the two projects randomly between all participants, taking care to give the same number of each projects to participants using a specific model. In this way we managed to have random participants using the same number of models on both FLOSS projects.

During the project planing process we addressed the following five aspects of the experimentation process:

**1. Object of the study – What is studied?**

The object of the study was one of the three FLOSS assessment models. At the same time we were also interested in the components of the three models. Participants had to answer to all questions present in the model and they were also asked to express their opinion on each question.

**2. Purpose – What is the intention?**

The purpose of the experiment was to predict the usability and precision of metrics inside the three models and the perception of the quality of the whole model. Based on the results, we wanted to know also which model better characterizes specific aspects of FLOSS.

**3. Quality focus – Which effect is studied?**

Our quality focus was the completeness of specific parts of the model and the precision of results for specific parts of the models. How detailed the model is in specific areas of FLOSS and whether the answers from different participants were similar or they diverged.

**4. Perspective – Whose view?**

Participants were students. We expected that they are mostly FLOSS users and a smaller percentage of FLOSS developers.

**5. Context – Where is the study conducted?**

The experimentation environment was a university laboratory.

An important aspect of the experiment conducted was the environment where the experiment was conducted. We included in the experiment mostly free willing participants from our university. We managed to involve 26 participants coming from the last year of the software engineering Bachelor and Master programs at our university. The experiments were always conducted in the same laboratory room where participants were able to use a computer connected to the web. Participants were separated and each was using his own computer. They were not allowed to communicate during the experimentation process.

The experiment had three phases:

1. First participants received an initial questionnaire where they were asked to report contextual data (age, experience in programming, experience with the assessed project, experience with the assessment model, and others information). This phase lasted 20 minutes.
2. For the second phase we distributed the printed version of questions relative to the assessment model they had to use. We gave them two hours (120 minutes) to assess the FLOSS project that was assigned to

them. They were able to browse web pages of the assessed project, search source code repositories, mailing lists, bug/issue management systems, and other web available sources to answer to questions that are part of the assessment model they used. We did not restricted their web access, they were allowed to search anywhere for information.

3. After the two hours, we asked them to finish the assessment process and we distributed a final questionnaire in which we asked them to describe their opinions of the quality of the model, the clarity of questions, the coverage of FLOSS aspects, and others. We gave them some possible answers to questions and allowed them to add also additional answers. The third phase lasted also 20 minutes.

Afterwards we collected results of all three phases and the experiment was concluded.

## 4 Results

We present first the contextual data about participants, then we present results of each assessment models used, their comparison, and at the end of this section we present opinions of the participants on the use of the three assessment models.

### Contextual data

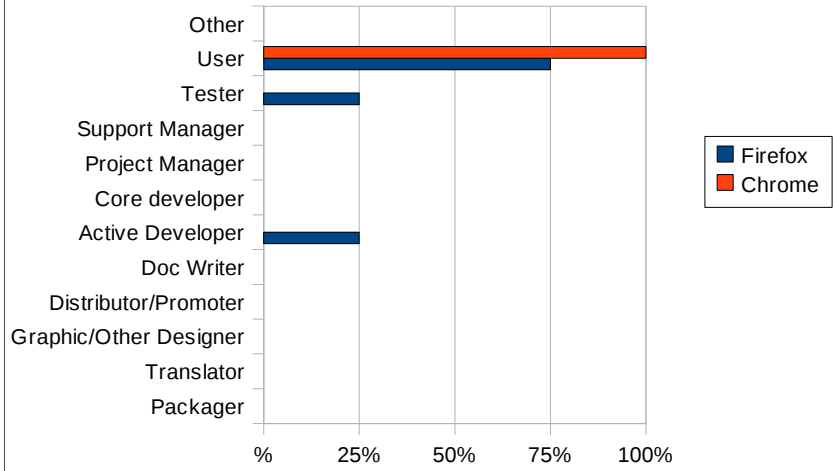
We collected many contextual data during the first phase of the experiment. We present here only a few aspects we consider important for understanding the results of the experiment:

- the role that best describes the current position in the assessed FLOSS project,
- the number of FLOSS projects the participant is or were involved in, and
- their experience in the assesed project related tasks.

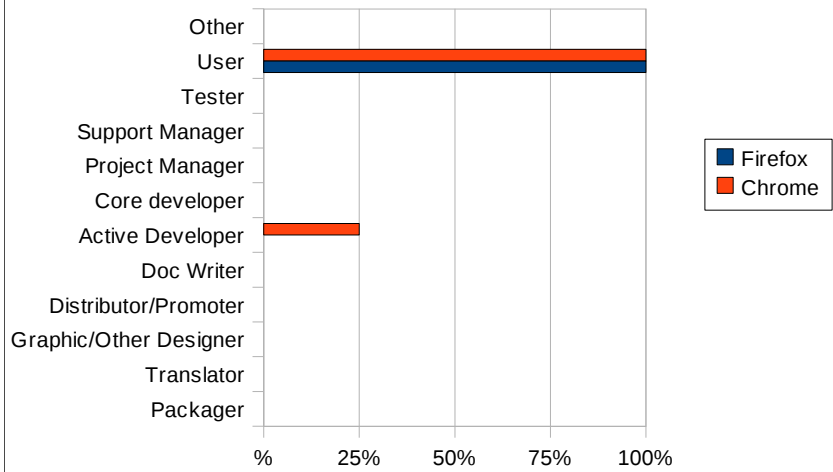
Some of the questions are personal, others are FLOSS specific, and some asks participants if they were already in contact with the assessed FLOSS project or the assessment model. If they were involved in the assessed project their assessment can influence the assessment process and can explain a better compilation of the assessment questionnaire.

Contextual question: **Role that best describes the position of the participant in the assessed FLOSS project**

**Which role below best describes your current position in the FLOSS project? (OpenBRR)**



**Which role below best describes your current position in the FLOSS project? (QSOS)**



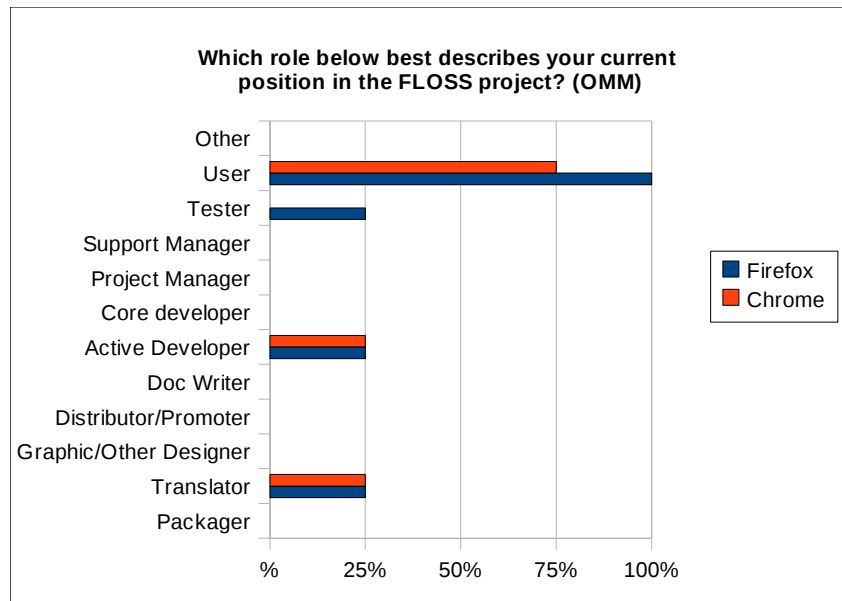


Figure 1: Role of participants in the assessed FLOSS project

From the three charts (Figure 1) we can see that almost all the participants have declared to have already used Firefox or Chrome browsers (nearly 100% of participants for both browsers; two participants did not choose any answer to this question). We do not know if they use them regularly or they have tried to use them just few times. Anyway, they are aware of the product and what it is used for. We can also see from the charts that few participants have contributed to the two FLOSS projects; they have declared to be testers, translators, or even active developers inside the Firefox or Chrome projects. We can see from the three charts that the number of developers is homogeneously distributed in all three assessment models groups. From the third chart we can see that we have an equal number of FLOSS developers involved in both Firefox and Chrome projects using the OMM methodology. Another peculiarity of the group using OMM is that we have additionally also a small number of translators participating to the experiment.

Contextual question: **Number of FLOSS projects you are/were involved in?**

Table 1: Number of FLOSS projects participants were involved in

	OpenBRR	
	Firefox	Chrome
<b>Number of FLOSS projects you are / were involved in?</b>		
Mean	0,67	0,75
Standard deviation	0,94	0,83



		QSOS	
		Firefox	Chrome
<b>Number of FLOSS projects you are / were involved in?</b>			
Mean		1	0,5
Standard deviation		1,22	0,87
		OMM	
		Firefox	Chrome
<b>Number of FLOSS projects you are / were involved in?</b>			
Mean		0,5	1,33
Standard deviation		0,5	1,25

Some of the participants of the experimentation were already involved in FLOSS projects (Table 1). We asked them in how many projects they have actively participated. We obtained comparable answers from them; some of them were never actively involved in any FLOSS project, others have participated in one or more (one student declared to participate in 3 FLOSS projects). From the three tables we can see that the mean is around one project and that it varies just slightly. We can notice that participants of the QSOS and OMM groups were involved in few more projects than the participants of the OpenBRR group. Anyway the difference between participants is small and it does not influence the results of the experiment.

Contextual question: **Your experience in this project related tasks?**

Table 2: Experience of participants in the assessed FLOSS project

		OpenBRR	
		Firefox	Chrome
<b>Your experience in this project related tasks (months)?</b>			
Mean		1,67	2
Standard deviation		2,36	1,63
		QSOS	
		Firefox	Chrome
<b>Your experience in this project related tasks (months)?</b>			
Mean		2,25	2
Standard deviation		2,49	2,12
		OMM	
		Firefox	Chrome
<b>Your experience in this project related tasks (months)?</b>			
Mean		2,25	4
Standard deviation		2,28	3,74

We wanted to know also if participants have actively contributed to the assessed FLOSS project. From the three tables (Table 2) we can see that participants were only marginally involved in the two assessed FLOSS projects. The variability

between the three groups is small, therefore we can be confident that this aspect does not influence the results of the experimentation process.

### Results of assessing two FLOSS projects using three assessment models

We provide here our interpretation for the results obtained by using each of the three methodologies; additionally, we present an overview of similar characteristics measured by the three methodologies. We calculated mean values and standard deviations for assessments done by participants for each quality characteristic. We expect that a low value of the standard deviation means a similar assessment result obtained by different participants. This can confirm that the questions were clear, the people were able to find appropriate information on the web, and the threshold values were defined appropriately.

Table 3: Use of OpenBRR

	OpenBRR			
	Firefox		Chrome	
	Mean	Standard deviation	Mean	Standard deviation
<b>Functionality</b>	3,7	0,5	2,3	0,7
<b>Usability</b>	4,2	0,4	4,1	0,6
<b>Quality</b>	2,7	0,3	3,5	0,3
<b>Security</b>	2,7	0,7	2,3	0,7
<b>Performance</b>	3,0	0,0	3,1	0,5
<b>Scalability</b>	4,0	1,0	3,8	0,0
<b>Architecture</b>	4,3	0,5	2,3	0,2
<b>Support</b>	4,8	0,2	4,7	1,3
<b>Documentation</b>	4,2	1,2	3,1	0,5
<b>Adoption</b>	3,8	0,3	3,2	1,5
<b>Community</b>	3,8	0,8	4,3	0,8
<b>Professionalism</b>	2,7	0,2	2,7	0,6

From the results presented in Table 3 we can notice that the Firefox project obtained better grades than the Chrome project; we can see this from most of the assessed criteria. The larger differences are on the Functionality and on the Architecture. Only two criteria obtained a higher grade for the Chrome project: Quality, and Community. The important information for us is the value of the standard deviation for different criteria. We can not identify a criteria that has a high standard deviation for both projects, therefore we can not be sure of the bad quality of a specific criteria. The criteria that were not assessed homogeneously for one or the other project (the standard deviation value is relatively large) were: Scalability, Documentation, Support, and Adoption.

Table 4: Use of QSOS

Generic Section		QSOS				
		Firefox		Chrome		
		Mean	Standard deviation	Mean	Standard deviation	
<b>Intrinsic durability</b>  <b>Industrialized solution</b>  <b>Exploitability</b>  <b>Technical adaptability</b>  <b>Strategy</b>  <b>Browser features</b> <b>Accessibility features</b> <b>Web technology support</b> <b>JavaScript support</b> <b>Protocol support</b> <b>Image format support</b>	<b>Maturity</b>	2,5	0,5	2,4	0,1	
		<b>Adoption</b>	2,9	0,8	2,5	0,4
		<b>Development leadership</b>	2,8	0,5	2,8	0,2
		<b>Activity</b>	2,9	0,6	2,7	0,2
	<b>Independence of developments</b>	<b>Services</b>	2,8	0,1	2,5	0,5
		<b>Documentation</b>	3,0	0,1	2,3	0,5
		<b>Quality Assurance</b>	2,9	0,2	2,5	0,4
		<b>Packaging</b>	3,0	0,4	2,2	0,5
		<b>Ease of use, ergonomics</b>	3,0	0,5	3,0	0,0
	<b>Administration / Monitoring</b>	<b>Administration / Monitoring</b>	1,0	0,8	2,3	0,5
		<b>Modularity</b>	2,9	0,1	2,0	0,0
	<b>Code modification</b>	<b>Code modification</b>	2,8	0,3	3,0	0,0
		<b>Code extension</b>	2,4	0,6	2,3	0,5
	<b>License</b>	<b>License</b>	1,9	0,1	2,2	0,2
		<b>Copyright owners</b>	1,4	1,0	1,7	0,5
		<b>Modification of source code</b>	2,3	0,5	2,7	0,5
		<b>Roadmap</b>	2,9	0,6	3,0	0,0
		<b>Sponsor</b>	2,9	0,1	2,3	0,5
		<b>Strategic independence</b>	2,8	0,5	2,0	0,0
		<b>Browser features</b>	2,5	0,9	2,7	0,1
	<b>Accessibility features</b>	2,4	0,6	2,4	0,3	
	<b>Web technology support</b>	2,9	0,4	2,2	0,1	
	<b>JavaScript support</b>	2,0	0,0	2,5	0,5	
	<b>Protocol support</b>	2,3	0,5	2,7	0,2	
	<b>Image format support</b>	2,5	0,4	2,7	0,4	

The QSOS assessment methodology has a different number of thresholds for assessing specific criteria than the other two methodologies (Table 4). QSOS has just three different thresholds. This aspect changes the range of values of the standard deviation. Also smaller values of standard deviations compared with the other two methodologies represent considerable deviations of assessment values given by users. We can see from the table that the highest standard deviation values are in the following characteristics: Adoption, Administration/Monitoring, Copyright owners, and Browser features. Also by using the QSOS methodology we can see that the Firefox project graded slightly better than the Chrome Project; however, the differences between the two are smaller than in the case of the assessment using the OpenBRR model. From the table we see also that the QSOS methodology has four different granularity levels for the summary of results. This is different from the other two methodologies. We decided to present results for the third level of granularity that has a similar number and type of characteristics as the chosen level of granularity of the OpenBRR and OMM models.

Table 5: Use of OMM

	OMM			
	Firefox		Chrome	
	Mean	Standard deviation	Mean	Standard deviation
<b>Product Documentation (PDOC)</b>	3,9	0,0	3,5	0,1
<b>Popularity of the SW Product (REP)</b>	2,7	0,1	2,4	0,4
<b>Use of established and Widespread Standards (STD)</b>	3,8	0,3	3,1	0,7
<b>Availability and Use of a Roadmap (RDMP)</b>	3,1	0,5	2,3	0,4
<b>Quality of the Test Plan (QTP)</b>	4,2	0,6	3,3	0,8
<b>Relationship between Stakeholders (STK)</b>	4,1	0,1	2,9	0,1
<b>Licenses (LCS)</b>	3,4	0,2	2,7	0,5
<b>Technical Environment (ENV)</b>	3,9	0,7	2,8	0,8
<b>Number of Commits and Bug Reports (DFCT)</b>	3,5	0,1	3,5	0,0
<b>Maintainability and Stability (MST)</b>	3,8	0,4	2,7	0,4
<b>Contributions to the FLOSS project from SW Companies (CONT)</b>	3,2	0,1	2,7	0,3
<b>Results of Assessment of the Product by 3<sup>rd</sup> Party Companies (RASM)</b>	1,8	0,3	2,1	0,1

From the mean values for different trustworthy elements (TWE is one of the characteristics measured inside the OMM model) composing the OMM model we can see in Table 5 that the Firefox project in comparison with the Chrome project obtained better grades. The difference of the quality of the two projects is even sharper than it appeared with the use of the other two methodologies. From our point of view, this difference between the two projects reasonable since the Firefox project exists a longer period and it has a larger community. There is only one TWE that is larger for the Chrome project than for the Firefox project and is: RASM. The level of RASM for Chrome is not high (2,1) but it is even lower for the Firefox project (1,8), showing that the Firefox product is not tested and/or presented appropriately on the project's website. The values of standard deviations calculated for OMM are lower than for the other two methodologies; keeping in consideration the 5 grades threshold levels. The higher standard deviations resulted for the Quality of the Test Plan (QTP) (Firefox project 0,6 and Chrome project 0,8) and the Technical Environment (ENV) (Firefox project 0,7 and Chrome project 0,8). The use of the OMM methodology on both projects showed higher standard deviation values for QTP and ENV. We analysed more in details the two TWEs and identified the questions that obtained largely heterogeneous answers by different participants. We plan to propose some changes related to those TWEs in the newer version of OMM.

### **Participants assessment of the three models**

In this section, we present answers to only two questions out of ten that we asked. We present only the answers that are important to better understand the use of the three methodologies.

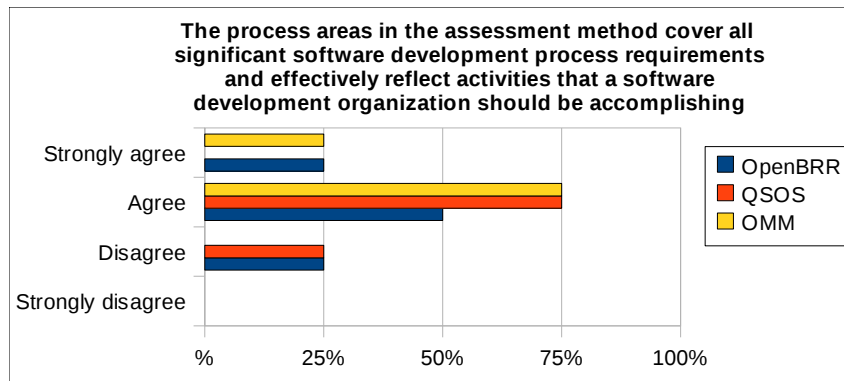


Figure 2: Completeness of coverage of FLOSS process areas

As evident from Figure 2, participants perceived OMM as flexible to a large extent; 12% strongly agree that the model is flexible and a large 88% perceived it as flexible. We think that this result is a good indication of the modularity of the proposed model. Other two models obtained similar, just slightly lower values.

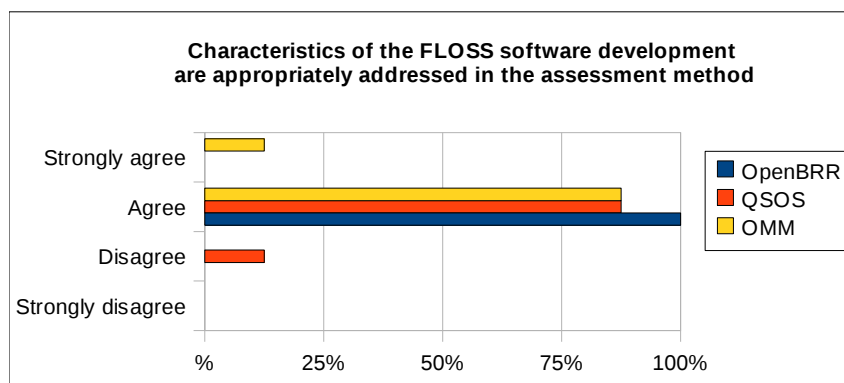


Figure 3: The model addresses appropriately FLOSS characteristics

The OMM model scored best on the question about the quality of coverage of characteristics that were actually included in the model (Figure 3). A large majority of users of OMM agreed with the sentence and none disagreed with it. The other two methodologies scored also good as is evident from the chart, only the QSOS model obtained a 10% of negative answers.

## 5 Threats to Validity

The randomization of used assessment models, of assessed FLOSS projects, and participants was a key requirement for the experiment and a possible threat to validity of the results obtained. Therefore, we planned this aspect in advance and try to mitigate its negative elements as already presented in the third section.

An aspect that we want to improve is the number of participants of experiments coming from the industry. We plan to conduct additional controlled experiments

with individuals coming from industry as programmers, and (integration) software projects managers. This is an important aspect for us to be able to see the usability of the three models in general for all potential users. A large percentage of experiments conducted with students can prevent the generalization of results. However, in the current experimentation process we wanted mainly to identify some problems of the three models and specially the OMM model in order to be able to improve it in the future. In the following iteration we want to identify the quality of the OMM model and the other two models in general and propose OMM to a larger user base.

Characteristics assessed by the three methodologies differ slightly, therefore not all of the characteristics can be compared. Small differences between some parts of the three models presented also a possible threat to validity of our conclusions. We managed to mitigate this threat by unifying the look and feel of the three models at the same time however not losing any content part of the three assessment models. Since we were not able to remove all differences, we preferred to present in this document separately results obtained for all three models and provide an interpretation and comparison just for a limited subset of comparable characteristics.

## **6 Discussion and Conclusions**

An important inconsistency in the used models was the naming of the characteristics assessed by each model. Although the three models contain characteristics that have sometimes the same name (for example license in OMM and QSOS) it is difficult or in some cases impossible to compare them because the internal questions are often (partially) different. For the conclusion we will present the comparison of one characteristic that is present in all three models (Documentation). The mean values obtained for the Firefox project using the three methodologies were: 4,2 (OpenBRR), 3,0 (QSOS), 3,9 (OMM). Taking in consideration that 3 is the highest value for QSOS we see that for this characteristic the three models obtained a similar value. The three values measure for the Chrome project were: 3,1 (OpenBRR), 2,3 (QSOS), and 3,5 (OMM). This three values are also quite similar (value 2 in QSOS is the intermediate value). Also the differences of the values for documentation of the Firefox and the Chrome projects are quite constant using all three models. For comparable characteristics we can see that the three models provide similar evaluation results for the two assessed projects.

Nevertheless we were mostly interested in the specific problems related to the OMM model, we identified also key problematic components of the OpenBRR and QSOS models. Based on the results of our experimentation process, we identify several problems related to each of the three assessment models. The Functionality and the Quality characteristics were assessed divergently by different participants using the OpenBRR model. The diverging values for the QSOS model were obtained for: Adoption, Administration/Monitoring, Copyright owners, and Browser characteristics. In the case of the OMM model the problematic characteristics were: Quality of the Test Plan (QTP), and the Technical Environment (ENV). Based on the final questionnaire filled during the

experimentation process, we found out that most often the problems related to specific questions are caused by a not clear formulation of the question and in some cases the not clear understanding of the threshold value (available value for the answer) used by each model.

Based on the results of the experimentation process conducted we saw that OMM obtained at least as good results as OpenBRR and QSOS models. In few aspects (Figure 2 and Figure 3) it was perceived better than the other two models. With the experimentation we found out some elements of the model that have to be improved:

- the identification of misleading questions inside the QTP and ENV Trustworthy elements,
- simplification of the overall complexity of some questions,
- simplification of thresholds values (current answers are complex and to extensive),
- automation of the scoring mechanism for calculating Practices, Goals, and TWEs scores (in some case we identified errors in the calculated results in all three models); and
- creation of an easy to understand description of how to use the OMM model.

These action items will help us modify appropriately the OMM model in order to improve its quality and usability.

## 7 Acknowledgement

The research was conducted in the scope of the QualiPSo project (FP-IST-034763). We are grateful to all QualiPSo partners and other participants to experimentation processes conducted.

## 8 References

1. E. Amoroso, J. Watson, M. Marietta, J. Weiss, "A process-oriented methodology for assessing and improving software trustworthiness"; Proceedings of the 2nd ACM Conference on Computer and communications security; ACM Press, 1994, pages: 39-50.
2. Atos Origin, "Method for Qualification and Selection of Open Source Software (QSOS)", <http://www.qsos.org> (Last visited: December, 2009).
3. F.-W. Duijnhouwer, C. Widdows, "Capgemini Expert Letter Open Source Maturity Model", Capgemini, 2003.
4. Hannay, J. E., Hansen, O., By Kampenes, V., Karahasanovic, A., Liborg, N., and C. Rekdal, A. A Survey of Controlled Experiments in Software Engineering. IEEE Trans. Softw. Eng. 31, 9 (Sep. 2005), 733-753. DOI= <http://dx.doi.org/10.1109/TSE.2005.97>.
5. Kitchenham, B. A., Pfleeger, S. L., Pickard, L. M., Jones, P. W., Hoaglin, D. C., Emam, K. E., and Rosenberg, J. Preliminary guidelines for empirical research in software engineering. IEEE Trans. Softw. Eng. 28, 8 (Aug. 2002), 721-734. DOI= <http://dx.doi.org/10.1109/TSE.2002.1027796>.

6. Navica Inc.: "The Open Source Maturity Model is a vital tool for planning open source success", <http://www.navicasoft.com/pages/osmm.htm> (Last visit: December 2009).
7. Petrinja, E., Nambakam, R., Sillitti, A.: Introducing the OpenSource Maturity Model. Workshop on Emerging Trends in Free/Libre/Open Source Software Research and Development collocated with 31st International Conference on Software Engineering (ICSE 2009), Vancouver, Canada.
8. Qualipso Consortium: QualiPSo - Quality Platform for Open Source Software. <http://www.qualipso.org/index.php>, Last visit December 2007.
9. Taibi, D., Lavazza, L., Morasca, S.: OpenBQR: A framework for the assessment of OSS, Open Source Software 2007, Limerick (June 2007).
10. A. Wasserman, M. Pal, C. Chan, "Business Readiness Rating Project", "BRR Whitepaper 2005 RFC 1".  
[http://www.openbr.org/wiki/images/d/da/BRR\\_whitepaper\\_2005RFC1.pdf](http://www.openbr.org/wiki/images/d/da/BRR_whitepaper_2005RFC1.pdf) (Last visited: December, 2009) .
11. Wittmann, M., Nambakam, R., Ruffati, G., Oltolina, S., Petrinja, E., Ortega, F.: "Deliverable A6.D1.6.3: CMM-like model for OSS",  
<http://www.qualipso.org/sites/default/files/A6.D1.6.3CMM-LIKEMODELFOROSS.pdf> (last visited December 2009).
12. Deprez, J.-C., and Alexandre, S.: Comparing Assessment Methodologies for Free/Open Source Software: OpenBRR and QSOS, Book chapter in Lecture Notes in Computer Science, Ed. Springer Berlin, June 2008, pg. 189-203.