Quality and libre software: a theoretical and practical approach.

AUTHORS

Andrés Estevez Costas, aecostas (at) gmail (dot) com
Javier Estévez Valiñas, valdaris (at) gmail (dot) com
Pedro A. González Pérez, pedro.gonzalez (at) dicoruna (dot) es

José Adrián Lema Cernadas, endri.from.vicedo (at) gmail (dot) com
Andrés Maneiro, andres.maneiro (at) gmail (dot) co
Pablo J. Sangiao Roca, psanxiao (at) gmail (dot) com
Juan Ignacio Varela García, nachouve (at) gmail (dot) com
Roberto Vieito Raña, robertovr (at) gmail (dot) com

All: Master on Libre Software, Caixanova/Igalia/Universidad Rey Juan Carlos

ACKNOWLEDGEMENTS

This paper has been elaborated with the effort of all its authors, the financial support by Caixanova, the technical support by Igalia and Universidad Rey Juan Carlos and the valuable help of the lecturers Daniel Izquierdo, Israel Herraiz and Teófilo Romera.

ABSTRACT

In the last years, libre software projects have gained importance. Companies are now trying to understand the structure of those projects. Nowadays there is still a lack of confidence on libre software projects in some scenarios. This is because there are not many formal processes to assure their quality, even when these processes exist for commercial software. In this case, they do not just assess the quality of the product, but also measure the quality of its development process. And the development process is quite a particular one in libre software projects.

What follows is a general study about how libre software world[1] is currently related with software quality. After an initial contextualisation as for quality in general and software quality in particular, libre software related issues are taken into consideration.

OpenBRR (Open Business Readiness Rating) is a new model "to rate software in an open and standardized way"[2]. This model will be applied to a precise libre software domain: GNU/Linux popular distributions, in order to show how it fits libre software quality assessment issues and what are its strengths and weaknesses.

According to this experience, it looks that there is still room for improvements in the quality assessment models field, specially when it comes to libre software products.

INTRODUCTION

Nowadays quality is, more a requirement than just a marketing motto. In commercial world, fulfilling quality exigences is a 'sine qua non' condition to access a broad sector of possible clients.

Libre software needs quality metrics to make objective comparisons among available solutions besides commercial goals. But it presents specific features that cannot be measured by the metrics currently used in the software industry. It is necessary to use *ad hoc* metrics to evaluate not only features related to the projects, but also to their development processes. This is because libre software projects are mainly developed through collaboration of groups of people, that contribute sporadically, heavily depending on software (CVS, SourceForge, bug tracking systems, mailing lists, etc), asynchronously, geographically distributed, with frequent releases, quick consideration of feedback and with many actors in the distribution chain. Understanding (and assessing) these community dynamics should be of great importance for rigorous quality metrics models.

Within this context, libre software needs to reinforce its *marketing techniques* by adding serious quality measurement models to its strengths. Historically, product quality has been looked at closely in libre software, and sometimes identified (arguably) as one of its flaws[3]. But this promotion may be useless if the development process quality issues are ignored. To solve this problem, several methods have been developed to define assessment processes suitable for libre software. Best known of these are QUALOSS[4], SQO-OSS[5], QSOS[6], OSMM[7]. Some of them focus on aspects such as maturity, durability or strategies adopted by the organization around the Open Source project itself. Other methodologies add functional aspects to the assessment process. In other cases, the correspondent projects are still not developed enough or they lack clear documentation, this way difficulting their use.

Open Business Readiness Rating (OpenBRR) is another of these methodologies. Intended to become "a new standard model for rating open source software"[2], it has been conceived to integrate companies constraints (notably for tests and reliability), and focuses on the sharing of the results and the reduction of the Total Perceived Cost of Ownership (TPCO) for libre software[8]. It is rather well documented and easy to use. This makes it to be the best known and most used of all of the mentioned ones. And that is why it has been chosen for this work.

Methodologies like OpenBRR can be understood as quality measurement models and thus they can be used to compare several different projects. Given that it makes no sense comparing an operating system to a chat client since the projects to be compared must be of the same type. And, of course, assessing criteria must be the same for all of the compared softwares and for each target user.

To put into practice the current existing methodologies and testing their validity, the aforementioned quality model will be applied to four GNU/Linux distributions. These distributions are: Ubuntu, Fedora, openSUSE and PCLinuxOS. We have chosen these GNU/Linux distributions according to their position in the *top ten distributions ranking* of DistroWatch[9]. Once this is done, and taking advantage of the assessing experience, the methodology itself is to be analysed in order to search for its strengths and weaknesses, and to consequently propose a set of possible enhancements to it.

GOALS

As stated before, the main purpose of this work is to analyze the relation between quality and libre software projects. OpenBRR provides a model for this purpose and several metrics and it will be applied in a real environment. We will assign a value for each of the selected GNU/Linux distributions according to this model. During the process we will also analyze the OpenBRR model. Nevertheless, the target is not only to evaluate the model itself, we will also try to

suggest how it could be improved.

As a consequence, the main targets of this work are the following:

- Application of a particular libre software quality metrics model to the specific domain of GNU/Linux Distributions
- Evaluate what is the highest quality GNU/Linux distribution as for a given user's profile.
- Discovering and analyzing the strengths and weaknesses of the metrics model used.
- Providing a list of suggestions and recommendations to enhance the capabilities and usability of the model.

METHODOLOGY

Regarding GNU/Linux distributions analysis, as stated before, it relies upon the use of OpenBRR methodology. This one provides a spreadsheet that contains a wide set of libre software related parameters to be analyzed and assessed. The weights of these parameters depend upon the profile of the appointed target user. Thus, the first task to undertake is the weights assignment to the different parameters according to the needs and requirements of a given type of user.

We have applied the OpenBRR methodology taking the point of view of an average user because she is the most common one. We consider this typical user to be made to Windows platform with her main requirements being the next ones:

- If something breaks, she will not use the distribution anymore, because she cannot solve problems by herself. Although she can visit forums or mailing lists in order to solve her problem, she will not likely do so. Hence, functionality and quality are more important categories than support and documentation.

- She wants to carry out any action in an easy, quick and intuitive way. Automatic configuration and a simple installation are necessary for this user to obtain a positive experience. Performance and usability are important categories too.

- The visibility of the application is very important because this kind of user tends to prefer vanilla versions of programs or, simply, the most popular ones. As a consequence, adoption category must be well considered.

- Compatibility with most popular file formats (multimedia codecs like mp3, office suite formats, and others) is required by default. This point is considered in the functionality category.

- An average user generally does not worry about security. The idea that GNU/Linux is more secure than Windows is very extended. And that is exactly the reason why many people move to GNU/Linux. Hence, there is a false sensation of 'total secureness' among those users who have not a solid knowledge on this issue. Due to this reason, we assign a low weight to this category. Even so, we do not weight it to zero because, despite the previously mentioned idea, security is actually an important fact in order to get programs correctly working.

Next task consisted of deciding what assessment categories are the most relevant to this users and, in consequence, assigning the correspondent set of weights. This was done by subdividing the group of authors into five sub-groups, each of which drafted a previous weighting proposal. These proposals were discussed by the whole group and, after a short conciliation process, a common proposal was agreed. Given that all of the initial proposals were rather similar in contents and weights, it was agreed that the final common one was made by assigning to each category the mean weight to each category.

The result of this work is that displayed in the following table:

ASSESSMENT CATEGORY	WEIGHT (%)
Functionality	22
Usability	22
Quality	11
Security	4
Performance	15
Scalability	0
Architecture	0
Support	3
Documentation	7
Adoption	16
Community	0
Professionalism	0

The categories depicted in the table are those proposed by OpenBRR. Thus the work of the group consisted of just weighting them according to the targeted profile. The set of agreed criteria to do so was the following:

- Functionality, Usability: all author groups assigned weights of 20% or bigger, as they understood that these were the categories being more important for the average user.
- Quality, Performance, Adoption: all groups marked these three categories as the second level set in importance.
- Security, support and documentation: they were marked as a minor set of categories.
- Scalability, Architecture, Community, Professionalism: no group assigned any weights to these four categories.

Regarding functionality and usability assessment of the GNU/Linux distributions, a list of tasks was agreed that have been carried out by the authors. These tasks try to cover the most usual features that these distributions offer to average users. And they are, once weighted, those introduced at OpenBRR spreadsheet to evaluate those aspects of the distributions.

These tasks were divided into the next groups:

- Installation. We measure the installation and basic configuration, setting up the language to Spanish or Galician[10], take note of any problem and installing from live-cd if possible.
- Office tools. We test the compatibility with open formats (odt) and the more common proprietary ones (Microsoft Office) because they are those that the average user needs. Also, we check if the localization includes grammar and ortographic tools.
- Multimedia. The goal of this section is to check if different video (divx, mpg, dvd) and audio (ogg, mp3, wav, wma) formats can be played. If some of them can not be played, it is desirable that an automatic tool searches for and downloads the adequate codec. On the other hand the study tests if multimedia keyboards work fine too.

- Web browser. We test how the default browser handles flash and multimedia formats (real video).
- Device recognition. We test the compatibility with the following devices: USB storage, digital cameras, printers, scanners, external monitors and function keys in laptops.
- Network configuration. We check how easy is the set up of a local network (ethernet and wi-fi).
- File recognition. On this section we test if the most popular compressed formats (zip, rar and 7z) are supported by default.
- Software update. We check if the distribution has a updating and upgrading management program.

RESULTS

The results obtained after assessing all of the four distributions are those displayed in the table below.

Distribution	Functionality	Usability	Quality	Security	Performance	Support	Documentation	Adoption	Total
OpenSUSE	0.44	1.1	0.39	0.20	0.45	0.15	0.35	0.80	**3.88**
Ubuntu	0.66	1.1	0.26	0.20	0.75	0.12	0.35	0.80	**4.24**
PCLinuxOS*	0.22	1.07	0.13	0.05	0.75	0.09	0.21	0.67	**3.19**
Fedora	0.22	1.1	0.24	0.17	0.75	0.15	0.21	0.80	**3.24**

*In this case, several metrics are missing due to the impossibility to access its bug tracking report system.

These results fit with the listing of the most popular distributions shown at Distrowatch[11] at the time of this work, showing an adequate correspondence between OpenBRR methodologies and the most common perception about what are the most idoneous distributions for general purpose uses.

As showed above, and once discarded the outsider scores from PCLinuxOS, most categories receive very similar scores for all of the distributions. Maximal differences appear to be obtained at 'Functionality'.

This seems to be a consequence of two main reasons:

- The proposed distribution of weights does especially reward this category, as explained above
- The use of different desktop environments by each of the distributions produces sensible differences in the functionalities offered by them

Moreover, the model itself seems to be basically addressed to the assessment of the fitness of applications to their target needs. This fact does also bias the results towards functionalities.

CONCLUSIONS

According to the results obtained, the figures of the study for each distribution vary into a limited range. PCLinuxOS can be considered as the outsider in the study because of the impossibility to get some of the information required for the metrics of the OpenBRR model. This model does not establish how this case should be marked.

Once all the work is done, it is time to evaluate how suitable is the methodology used to meassure the quality in the

given selected domain. This point is segmented into sections: On the one hand it shows some weaknesses found in openBRR; on the other hand it proposes a list of suggestions to improve the fitness of OpenBRR to the more common quality meassurements processes and metrics.

The major weaknesses found during the work with OpenBRR were the following:

1. OpenBRR weights assigning process seems to be too dependent on subjective criteria, so that comparing results between different software analysis is not accurate enough to extract any objective conclusion. The lesser the personal options, the more comparable the results.
2. As a consequence, currently OpenBRR is not an adequate tool to support a quality certification, unless weighting criteria are fixed for a set of given possible software users/needs.
3. OpenBRR assessing categories and sub-categories may still be re-studied so that they fulfill a set of requirements as wide as possible, and so that they measure other relevant issues or metrics.
4. It would be desirable that the model made use of a more finished tool.
5. The OpenBRR website may not fullfil the expectatives of some IT managers who want to evaluate if they are going to use OpenBRR as their software metric. Lacks of clearly and up to date structured information on the use, objectives and benefits of the project, and it seems that the OpenBRR brand image is still not well set.
6. The current repository of studied examples is insufficient even to compare a few number of them per software category, which produces a severe lack of utility and practical use of OpenBRR.
7. Some internal OpenBRR metrics (like number of bugs, number of unique developers) offer a quantitative assessment of the project without taking into consideration its size.

Some of the results of the present study seem to show that there is still margin for improvement in different aspects. The following is a set of suggestions that have arisen during the whole process of doing this work:

1. Masks of weights could be helpful to address problems derived from the subjective weights assigning phase. The assignment of weights on these masks could be obtained as an average from the opinions of a wide range of professionals and users from each domain and software category. Obtaining this way one mask for each category which is shared allowing the comparison of results from different sources.

2. The tool provided by OpenBRR may be more powerful than the current set of spreadsheets. A web application, including the previously defined and calculated weights masks, could be more suitable for the needs of a model like this.

3. A tool like the one described in the previous two statements, would allow an efficient gathering, integration, processing and latter publication of data. The availability of such a data repository would increase the accuracy of comparisons held on libre software products. This feature would also add the possibility of comparing one single product against the data in the repository. Having this structured data repository is essential to achieve the main goals of the OpenBRR model.

4. All the data gathered with a tool like the one described above would allow to define a set of categories and sub-categories for each software domain. These predefined sheets would help to achieve uniformity in measurements in each domain and to reduce noise introduced by the subjectivity of the current method, which presents the same set of categories no matter the software domain.

5. Internal OpenBRR metrics may be re-thought so that they can get adapted to the different sizes of studied projects within each category. These metrics should take into account parameters defining the size of the project such as SLOC, number of mailing lists, number of subscribers in these mailing lists or number of committers.

6. The bigger the repository of comparable examples of studied software, the more useful the model would become in order to analyse what is the most adequate application for a given set of users and needs. As previously pointed out, this would be possible if metrics and weighting criteria are standardised and fixed for each category of software applications.

7. A more thoroughly designed website would transmit a better image of the tool which would also result in a more positive perception of the OpenBRR methodology. As an example, at the time of doing this work the web page and the documentation were clearly out of date.

Another suggestion that is not critical at all, is that the '0-5' range of global marks seems to be a quite restricted one to reflect relevant assessing differences between different software tools of the same families. It must be taken into consideration that within this work Linux distribution have been analysed, not a given family of software tools. If this was the case, functionality assessment would produce more similar scores for the different tools, as they all would have been designed to fulfill a similar set of requirements. And, as seen, functionality is precisely the category that has produced most different scores at our work. A possible solution to this problem would be that of adopting a '0-100' range for the global score, as it would be more easily understood at the time that it would better reflect marking differences.

In opposition to what precedes, OpenBRR makes a rather fine work when assessing software applications compliance to individual users needs. This is so because, even containing some quantitative metrics, it does mainly measure software against a set of subjective criteria. This way being a good tool to assess what is known as 'perceived quality' of software [Gousios et alt., 2007][12]. However it is not that good to measure other quality categories as 'product' or 'processes', both of which are better assessed by means of objective and quantitatively measurable criteria.

Actually, one of the main characteristics of libre software is the fact that almost all the data generated during the development process are publicly available. This allows software engineering research community to use data mining technologies to quantify different aspects of the libre software development process.

OpenBRR seems to be a good first approach to quality assessment. However it should definitely be improved by reactivating its community and using data gathering techniques to maintain a repository of analyzed projects which anybody could query and which gives comparable results. Taking such approach further would be of great help for the widely adoption of libre software in the industrial market. This target is being faced by several other initiatives at this moment in search for new libre software quality assessment models.

1 Through out this paper we will use the term “libre software” to refer to any code that conforms either to the definition of “free software” (according to the Free Software Foundation) or “open source software” (according to the Open Source Initiative).

2 From OpenBRR web site: http://www.openbrr.org/

3 Among other documents, see:

- http://www.cyrius.com/publications/michlmayr_hunt_probert-quality_practices_problems.pdf : "[...] problems related to quality, such as the volatility of volunteer contributions, and the inability of developers to keep up with bug reports."
- http://www.cyrius.com/publications/michlmayr_hill-reliance.html : "[...] most Free Software development is performed by volunteers who cannot be relied upon for consistent levels of work in the same way that non-volunteers in the most commercial projects can"

4 http://www.qualoss.org/

5 http://www.sqo-oss.eu/

6 http://www.qsos.org/

7 http://www.oss-watch.ac.uk/resources/osmm.xml

8 http://www.openbrr.org/como-workshop/papers/GermanRoblesGonzales-Barahona_EFOSS06.pdf

9 http://www.distrowatch.com/ The study has been carried out with four chosen GNU/Linux distributions out of the Top Ten Distributions at Distrowatch, avoiding selecting two from the same family, such as Debian and Ubuntu.

10 These languages were chosen as examples to measure the possibilities of localization of the different distributions, beyond their original language.

11 http://distrowatch.com/dwres.php?resource=major

12 Georgios Gousios et Alt: "Software Quality Assessment of Open Source Software". In: Proceedings of the PanHelenic Conference on Informatics, Patras, Greece (2007) Document available at http://pci2007.upatras.gr/proceedings/PCI2007_volA/A_303-315_Gousios.pdf