The Influence of Software Product Quality Attributes on Open Source Projects: A Characterization Study

Antonio Cesar Brandao Silva[1], Kattiana Constantino[2], Glauco de Figueiredo Carneiro[1], Antonio Carlos M. de Paula[1], Eduardo Figueiredo[2], Miguel P. Monteiro[3] and Fernando Brito e Abreu[4]

[1] *Universidade Salvador (UNIFACS), Salvador, Bahia, Brasil*
[2] *Universidade Federal de Minas Gerais (UFMG), Belo Horizonte, Minas Gerais, Brasil*
[3] *Universidade Nova de Lisboa (UNL/NOVA LINKS), Lisboa, Portugal*
[4] *Instituto Universitario de Lisboa (ISCTE-IUL), Lisboa, Portugal*

Keywords: Software Releases, Open Source Software (OSS) Projects, Software Product Quality Attributes.

Abstract: Several Open Source Software (OSS) projects have adopted frequent releases as a strategy to deliver both new features and fix bugs on time. These entails express requests from the project's community, registered as issues in bug repositories by active users and developers. Each OSS project has its own priorities established by their respective communities. A still open question is to what extent these priorities influence selection of the issues that should be tackled first, implemented/solved and delivered in subsequent releases. In this paper, we present an exploratory study on the influence of target product quality attributes in software release practices of OSS projects. The goal is to search for evidence that clarify the relationships between target attributes, priorities assigned to the registered issues and the ways they are delivered by product releases. To this end, we asked a set of participants to identify these attributes through the data analysis of repositories of three well-known OSS projects: Libre Office, Eclipse and Mozilla Firefox. Evidence provided by the participants suggest that OSS community developers use criteria/priorities driven by specific software product quality attributes to plan and perform software releases.

1 INTRODUCTION

The attractiveness of Open Source Software (OSS) projects for users and developers communities has aroused the interest of Software Engineering researchers (Michlmayr et al., 2015) (Fitzgerald, 2006) (Gonzalez-Barahona et al., 2013) (Gonzalez-Barahona and Robles, 2013). OSS development differs from proprietary development in various aspects, such as development processes, team structure, and developer incentives (Jonsson et al., 2015). Understanding the rationale behind the success of OSS projects can help the staff and developers of non-OSS projects to draw lessons from reported best practices and apply them to their projects (Stol and Fitzgerald, 2015) (Rigby et al., 2012).

According to Adams and colleagues (Adams et al., 2015), there are more than 400 active OSS distributions, and each year an always greater number of projects are created. Due to competition and pressure of users and developers, OSS projects need to release new features and bug fixes within increasingly shorter time spans. Releasing software every few weeks is typically referred to as a frequent release cycle, while releasing quarterly or yearly is typically referred to as a traditional release (TR) cycle (Mantyla et al., 2013). The adoption of frequent releases in OSS projects takes into account that users of these projects are scattered all over the world and eagerly download each new version as soon as it is released and test it as thoroughly as they can. The global dispersion of users means that the code can be tested 24 hours a day (Thomas et al., 2009). To achieve this, OSS projects rely on volunteers to analyze, implement and test demands registered as bug issues to integrate them to the latest version (Adams et al., 2015). The process of reporting and resolving issues for a system during its development and/or maintenance is often handled through the use of an Issue Tracking System (ITS). For each issue, an ITS can typically record its type (e.g., defect, enhancement, patch, task), its state (e.g., new, assigned, resolved, closed) and date of submission. For each state change, an ITS usually allows recording the submit-

Silva, A., Constantino, K., Carneiro, G., Paula, A., Figueiredo, E., Monteiro, M. and Abreu, F.
The Influence of Software Product Quality Attributes on Open Source Projects: A Characterization Study.
DOI: 10.5220/0006265400290039
In *Proceedings of the 19th International Conference on Enterprise Information Systems (ICEIS 2017) - Volume 2*, pages 29-39
ISBN: 978-989-758-248-6

ter, any comments by others, and indications of severity and/or priority (Bijlsma et al., 2012).

Despite being supported by volunteers, relevant OSS projects have experts in their communities that have in-depth knowledge of the source code and components (Ye and Kishida, 2003). They are usually ranked according to their reputation in the project community as a result of their contributions throughout the releases (Oreg and Nov, 2008).

Several aspects may influence issue solving priority. Recurring tickets on ITS imply bug fixing (improving reliability) or adding new features (improving functionality) and those are the ones that community members usually vote for assigning a scheduling order (Crowston et al., 2003). However, other software product quality attributes may be critical for the project's success. It is therefore important to recall the ISO/IEC 25010 international standard for software product quality (ISO, 2011), that replaced the "old" ISO/IEC 9126 standard (ISO/IEC, 2001), by providing a quality model composed of several software quality characteristics, that are further broken down into many sub-characteristics, as represented in Figure 1.

In this paper, we describe an exploratory study to investigate the influence of target product quality attributes in software release practices of three OSS Projects (Libre Office, Eclipse and Mozilla Firefox). That is, we aim at obtaining evidence on how those quality attributes are related to the priorities assigned to registered issues and how they are delivered by product releases.

The paper is organized as follows. Next section presents the related work. Section 3 outlines the exploratory study. Section 4 presents the data collection and analysis. Section 5 presents the conclusion, threats to validity and scope for future research.

2 RELATED WORK

Some studies reported the importance and influence that software quality attributes have in the software development cycle of a product (Perepletchikov et al., 2005)(Offutt, 2002). This influence is also true for OSS projects. Two surveys in this direction were conducted by Henningsson and Wohlin (Henningsson and Wohlin, 2002). The first survey focused on the literature to capture the understanding of the quality attributes in the research community. The second one is an interview survey focused on the perception of the industry regarding the practice and understanding of the quality attributes in an industrial context. The authors concluded that it is clear, from both the literature and the industrial surveys, that relations do exist among many of the software quality attributes.

Quality is about perception and depending on the needs and how well the final system or construction can fulfill these needs, the grading of a particular quality attribute is influenced. "Grading" herein refers to the perceived importance of that attribute (Henningsson and Wohlin, 2002).

The Eclipse Foundation has also undertaken annual surveys to the Eclipse community to better understand things such as user perception regarding Eclipse functionalities or how users and developers get engaged into the OSS community.[1] [2] [3]

3 EXPLORATORY STUDY

In this section, we present an exploratory study to analyze how software quality attributes influence the prioritization of issues to be implemented/solved and delivered in future releases. Exploratory studies are intended to lay the groundwork for further empirical work (Wohlin et al., 2012). For this reason, there is no control group to compare to. The adopted strategy consisted in asking developers, outside the community of the selected projects, to search for evidence in data provided by the repositories of the selected OSS projects. This paper aims to address the following four Research Questions.

RQ1- *Considering data available in public repositories, which software product quality attributes are prioritized in software release practices by OSS Projects?* The identification of software quality attributes in OSS projects is a key to understand their priorities throughout their evolution. In practice, depending on the software quality attributes that are prioritized, specific issues are targeted and implemented in the next releases while others will be postponed or even discarded.

RQ2- *Which strategies can be used to identify the prioritized software quality attributes of OSS project repositories?* Unveiling strategies to find prioritized software quality attributes are useful to understand the rationale used by OSS projects. Successful identified practices can be referenced and followed by industry practitioners, as well as by other OSS projects.

RQ3- *What are the values assigned by developers to the fields "priority" and "severity" of issues registered in the projects' bug repositories related to both priority and non priority software product quality at-*

[1] https://goo.gl/LtzK5E

[2] https://goo.gl/2Fpm7c

[3] https://goo.gl/R3NHpw

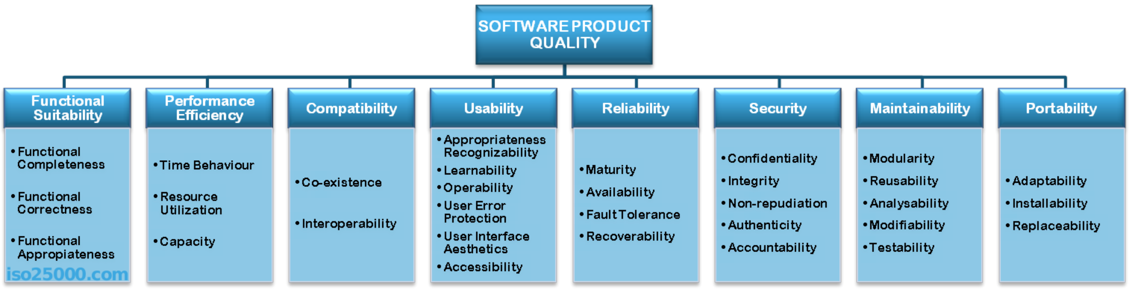


Figure 1: The ISO/IEC FCD 25010 Product Quality Standard (ISO, 2011).

tributes? This evidence can show how OSS software projects address other issues that are not directly related to the likely prioritized software quality attributes.

RQ4- *Are issues related to priority software product quality attributes delivered in releases just after they were registered?* The time elapsed to deliver an issue can be an evidence of its priority, according to the software quality attributes related to it.

To answer these questions, we defined a protocol (Subsection 3.3) and instantiated it in three phases. These phases are conveyed in Figure 2, where Steps 1 and 2 were related to the planning of the study and Steps 3 to 6 were repeated in Phases 1-3 of this exploratory study. This protocol focused on the analysis of data that have relationships with the bug issues of three OSS projects (Subsection 3.2). One of the authors conducted a tutorial session to present participants the tools and resources that could be used to perform the tasks.

3.1 Data Collection

Data Collection. Data collection was carried out directly from answers provided by the participants. To answer RQ1, we collected information related to quality attributes. To answer RQ2, we collected evidence based on references provided by participants, e.g., print screens and URLs from which relevant information is accessible. To answer RQ3 and RQ4, we analyzed the issues indicated by the participants and the authors.

Passing Score. After the tutorial, we analyzed data provided by the participants in accordance with the following two criteria. *Criterion 1* was indication of at least one quality attribute consistent with data from the analyzed repository. Therefore, inappropriate indications of quality attributes do not meet this criterion. *Criterion 2* was indication of evidence from the repositories supporting the choice of the selected quality attribute. Therefore, inappropriate indications of evidence do not meet this criterion.

3.2 Target OSS Projects

This study relies on three highly regarded open source projects: *Mozilla Firefox*, *Eclipse*, and *Libre Office*. We selected those projects mainly due to the three following reasons. First, they adopted frequent releases implementation (da Costa et al., 2014) (Gamalielsson and Lundell, 2014). Second, they have a very active developer community comprised of 290 (*Libre Office*), 1087 (*Mozilla Firefox*), and 113 (*Eclipse*) members, respectively. Third, all projects provide a vast repository of documentation, publicly available and readily accessible. Finally, these projects have been cited in the literature in different software engineering studies, which enhances their value as the basis for the present study (Gamalielsson and Lundell, 2014).

3.3 The Study Protocol

Seven participants took part in this study to answer research questions from RQ1 to RQ4. They were asked to register the collected evidence related to the analysis they performed in the questionnaire form. Moreover, the participants were asked to describe their strategies, as well as their experience while using the OSS project repositories to accomplish the tasks. The participants had available one week to performed the asked tasks.

The study involved seven participants recruited from a MSc and PhD programs in Computer Science from two different universities. This number of participants offered a reasonable trade-off between the effort to plan and execute the study and detailed qualitative analysis and the generalizability of results (Pfleeger, 1995) (Yin and Campbell, 2003). They were all volunteers and no compensation was provided for their participation in this study.

We were interested in making observations based on a qualitative analysis of OSS projects repositories, regarding the possible influence of software product quality attributes on the scheduling of pending issues that may have affected the evolution of the an-

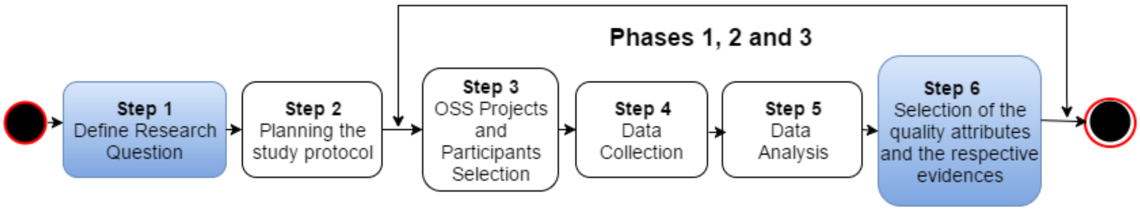
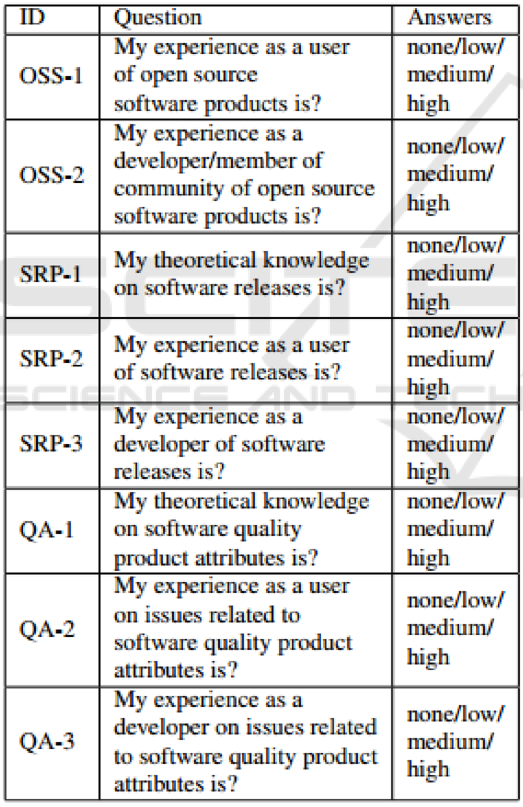

Phases 1, 2 and 3
Step 1 Define Research Question
Step 2 Planning the study protocol
Step 3 OSS Projects and Participants Selection
Step 4 Data Collection
Step 5 Data Analysis
Step 6 Selection of the quality attributes and the respective evidences

Figure 2: Overview of the exploratory study.

alyzed software product, rather than testing causality hypotheses using statistical inference. To be eligible for inclusion, participants filled out a pre-study questionnaire (figure 3) to describe their profile and experience in software release practices, OSS projects and software quality product attributes. In Table 1 the corresponding answers using the following scale: none(N)/low(L)/medium(M)/high(H).



ID	Question	Answers
OSS-1	My experience as a user of open source software products is?	none/low/ medium/ high
OSS-2	My experience as a developer/member of community of open source software products is?	none/low/ medium/ high
SRP-1	My theoretical knowledge on software releases is?	none/low/ medium/ high
SRP-2	My experience as a user of software releases is?	none/low/ medium/ high
SRP-3	My experience as a developer of software releases is?	none/low/ medium/ high
QA-1	My theoretical knowledge on software quality product attributes is?	none/low/ medium/ high
QA-2	My experience as a user on issues related to software quality product attributes is?	none/low/ medium/ high
QA-3	My experience as a developer on issues related to software quality product attributes is?	none/low/ medium/ high

Figure 3: Pre-study questionnaire.

The Tutorial Session. Prior to the tasks, the participants attended a tutorial session focusing on how to search for evidence in the *Open Stack IAAS Cloud Platform* project using data provided by tools such as: *Bugzilla* (issue tracking system), *Git* (configuration management system), *Gerrit* (code review tool for Git) and a *Wiki*. These tools are also used by the selected OSS projects for this study. We selected a different project in the tutorial session to avoid biasing in the study results. The first part of the tutorial focused on how to understand and search for data using the tool repositories for the *Open Stack* project. The second part focused on using the selected repositories to identify evidence that could establish relationships with the corresponding software product attributes prioritized by *Open Stack IAAS Cloud Platform*.

Table 1: Results of the Pre-activity Questionnaire.

ID	P1	P2	P3	P4	P5	P6	P7
OSS-1	N	M	L	H	M	L	H
OSS-2	N	N	N	N	N	N	M
SRP-1	M	L	L	H	M	L	L
SRP-2	M	L	L	H	M	M	H
SRP-3	M	M	L	H	M	N	M
QA-1	L	M	M	M	L	L	L
QA-2	L	M	M	L	N	M	M
QA-3	L	M	L	N	N	N	L

4 DATA ANALYSIS

This section presents the analysis of the answers provided by the participants to the aforementioned research questions.

4.1 First Phase

In the first phase of this experiment, we asked participants to indicate software product quality attributes that are best related to each project along with respective evidence that justify the indication. The goal was to answer research question **RQ1:** *Considering data available in public repositories, which software product quality attributes are prioritized in software release practices by the OSS Projects Mozilla Firefox, Libre Office and Eclipse?* and **RQ2:** *Which strategies can be used to identify the prioritized software quality attributes of the selected OSS project repositories?*

There was no predefined quantity of attributes to be indicated. Table 2 presents the quality attributes indicated by the participants for the three target projects. The following attributes, already mentioned in Section 2, were considered in the table: *Functional Suitability (FS)*, *Performance Efficiency (PE)*, *Compatibility (C)*, *Usability* (U), *Reliability (R)*, *Security*(S), *Maintainability (M)*, and *Portability (P)*.

According to the data shown in Table 2, the quality attributes indicated by the participants are presented as follows to answer **RQ1**. Figure 4 conveys the quantity of attributes indicated by each participant of this study. 5 (71%) participants indicated for Eclipse the attributes *Compatibility* and *Maintainability*; *Portability* was indicated by 4(57%) participants. In the case of Mozilla FireFox, *Maintainability* and *Portability* were both indicated by 4(57%) of the participants, whereas *Performance Efficiency*, *Reliability* and *Security* were also indicated for this project for 3(42%) participants. FireFox had indications of all quality attributes conveyed in Figure 1. Finally, Libre Office had *Usability* as the quality attribute with 6 indications. The possible reason for this result is the concern of the developer's community with a friendly and intuitive product, as well as the competition with other similar products. *Functional Suitability* and *Maintainability* were mentioned by 4 participants. Libre Office had less uniformity among the indicated attributes. For example, *Reliability* and *Security* were not mentioned and *Performance Efficiency* and *Portability* received just 1 vote each

After the indication of the attributes, the participants looked for evidence to answer **RQ2**. The study protocol did not establish a minimum number of issues, so participants were free to present as many issues as they saw fit [4].

Table 3 conveys the distribution of issues indicated by participants with respective associated software product quality attributes for the target projects.

To point out the selected quality attributes, participants 1, 3, 5, 6 and 7 adopted the strategy to filter issues by their *severity level* together with their respective *bug priority*. These participants justified the use of this strategy considering that projects adopting the frequent release approach need to foster the selection of failures and new features that may have a significant impact on the software product. Participant 4 adopted another strategy. He searched for relevant quality attributes by analyzing each software project profile and scrutinizing data in the software project portals and wikis. However, this strategy revealed as not effective, considering that only selecting quality attributes through portals and wikis, does not necessarily reflect the relationship of these attributes to the release schedules of those projects. Finally, participant 2 performed a quantitative analysis of the issues found in the projects' bug tracking systems, considering the response/solution time for the main components of the projects based on the quality attributes chosen by the participants.

4.2 Second Phase

The second phase aimed at identifying the attributes of issues that have severity assigned as *blocker* or *critical* to compare these attributes with those indicated by the participants in the first phase. The reason to search for issues with these types of severity is to focus on issues considered relevant in the context of the analyzed projects. During this search, we targeted the same versions/releases of the projects indicated by the participants. This second phase comprised the following researching question: **RQ3:** *What are the values assigned by developers to the fields "priority" and "severity" of issues registered in the projects' bug repositories related to both priority and non priority software product quality attributes?*

4.2.1 Eclipse

The search for the issues of Eclipse considered versions 3.3.2, 3.4, 4.4, 4.5 and 4.6 as well as severity *blocker*. As a whole, it returned 86 issues. The issues were analyzed one by one by the researcher to be characterized one or more assigned quality to which it is defined.

Figure 5 presents the results of this characterization of issues for the Eclipse project. Of the 44 issues returned in the searches, 66% were about *Compatibility* issues, either with one of the previous versions, or with some of the coexisting packages. Eclipse is a large project with several coexisting packages and many problems have been reported related to this issue. 21% of the issues raised relate to *Functional Suitability* and 9% to *Portability* , mainly on installation. Just 4% relate to *Performance Efficiency*. Other attributes were not identified for any issue. By coincidence, all issues had the same level of priority.

4.2.2 Mozilla Firefox

For Mozilla FireFox, the following versions were considered: 23 Branch, 42 Branch, 43 Branch, 45 Branch, 48 Branch, and 49 Branch. In addition, the severities considered were *blocker* and *critical*. In these configurations, in total, 28 issues were returned and an individual analysis to identify which attribute of quality is more closely represented was carried out. Figure 6 presents the results. The first perception is that several attributes were identified in this sample, as presented in table 3 in the participants' perception

[4] These issues are available at https://goo.gl/LdkXwJ

Table 2: Product Quality Attributes Indicated by the Participants.

	Eclipse								Mozilla Firefox								Libre Office							
	FS	PE	C	U	R	S	M	P	FS	PE	C	U	R	S	M	P	FS	PE	C	U	R	S	M	P
P1		1			1			1		1						1	1			1				
P2	1		1	1			1	1		1		1		1	1	1	1	1		1			1	
P3		1	1	1	1		1	1	1	1	1	1	1	1		1	1		1	1				
P4			1				1						1		1					1			1	
P5			1				1						1		1					1			1	
P6							1								1								1	
P7	1		1	1				1			1			1		1	1		1	1				1
Total	2	2	**5**	3	2	0	**5**	**4**	1	3	2	2	3	3	**4**	**4**	**4**	1	2	**6**	0	0	**4**	1

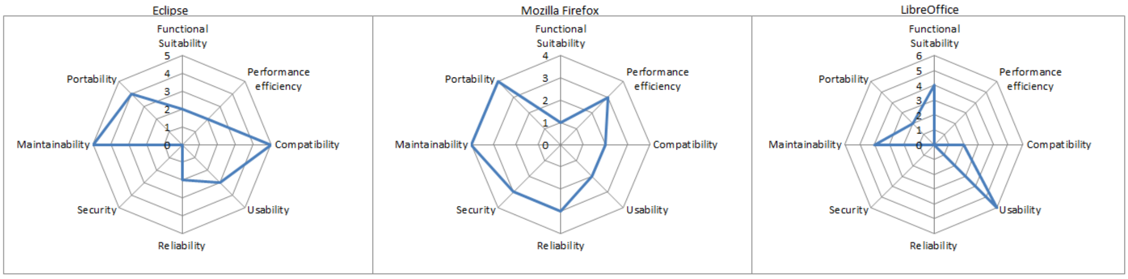


Figure 4: Software product quality attributes prioritized on releases according to participants.

Table 3: Issues indicated by Participants in Phase 1.

PQA	Eclipse	Mozilla FireFox	Libre Office
Functional Suitability	1	0	5
Performance Efficiency	3	2	1
Compatibility	2	1	1
Usability	0	0	4
Reliability	1	2	0
Security	0	0	0
Maintainability	2	2	0
Portability	6	3	0

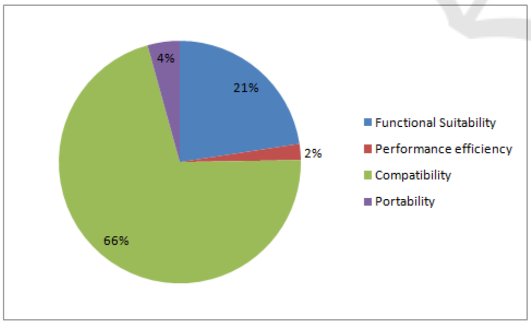


Figure 5: Product Quality Attributes of the Eclipse Issues.

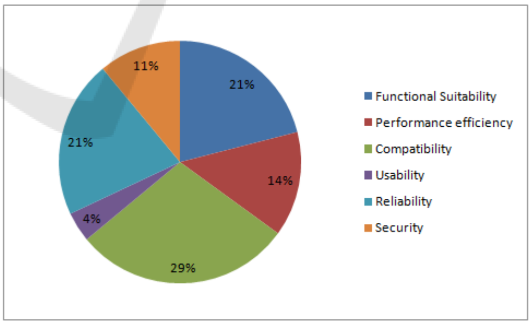


Figure 6: Product Quality Attributes of the FireFox Issues.

when they also mention more uniformly all the quality attributes for FireFox. The attributes mentioned were *Compatibility* 29%, *Functional Suitability* and *Reliability* with 21%, *Performance Efficiency* with 14%, *Security* with 11% and *Usability* with 4%.

4.2.3 Libre Office

Finally, for Libre Office, the following versions were considered: 3.4.3, 4.0.0.3, 4.1.0.2, 4.2.0.3, 4.2.4.2, 5.0.0.5 and 5.0.3.2. The levels of severity considered were *blocker* and *critical*, from which 11 issues were derived. The issues were analyzed individually. Figure 7 presents the results of this characterization for Libre Office. Of the 11 issues returned, 42% were about *Functionality* , 33% relate to *Performance Efficiency*, 17% to *Compatibility* and 8% relate to *Portability* . Other attributes were not identified for any issue.

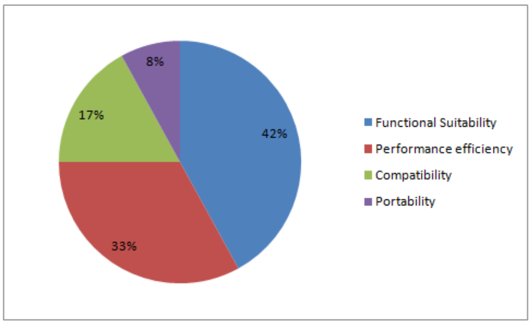


Figure 7: Product Quality Attributes of the Libre Office Issues.

4.3 Third Phase

The third Phase was to search for the issues repository, to check those that were not indicated by participants, but had a high level of severity in the three projects and related to the quality attributes selected by the participants pointed in Figure 4. We also checked issues implemented/resolved in releases near or far from their date of registration. This way, it is possible to verify the consistency of the selected attributes and allows to verify how they were prioritized in the release planning.

The adopted strategy was to initially define a filter that could select the issues of higher priority and severity in the Bugzilla and that were already finalized or resolved.

The following step was to read the description of the issues. If any evidence was found, which pointed to a given quality attribute, that issue would be selected and all its flow would be analyzed so as to confirm it. As in step 1, it was taken into account whether the issues were implemented/ resolved in releases that were close to their registration date. We wanted to detect the priority relationships in the release planning depending on the quality attribute indicated.

Finally, with this phase we derived research question **RQ4**, as stated in Section 3.

4.3.1 Eclipse

For Eclipse, the following filter was defined: *(Status: Resolved, Closed; Resolution: Fixed; Priority: P1; Severity: Blocker, Critical; Classification: Eclipse)*. The focus was on the following quality attributes: *(Functional Suitability (FS), Performance Efficiency, Compatibility, Usability, Reliability, Maintainability and Portability)*.

This query returned 492 issues, which exposed 24 issues that had some connections to the selected quality attributes. One example that stood out was issue 1310554 [5], which includes the following description *"[GTK3] Problem with table/tree editing"*. It was registered on the 29th of June 2014.

This description enabled us to make a connection with the product's functionality. However when analysing the bug report for more detailed information, we established a connection with *Compatibility*, because the user pointed out that when using a version of GTK (the multi-platform toolkit for creating graphical user interfaces) some commands would behave abnormally and that with a newer version of GTK no command would be working *"The editing works correctly with GTK2. GTK3 <= 3.8 - it works, but a context menu doesn't work. Right-click in an editing field, leave edit mode and actions (Cut, Copy, Paste ...) don't work. GTK3 >= 3.10 - you can't enter editing mode."*.

Finally, Figure 8 shows the distribution quality attributes. Of the 24 issues returned, 20.8% were about problems relating to *Functional Suitability (FS)*, 12.5% were about *Performance Efficiency*, 25% about *Compatibility*, 16.70% about *Reliability*, 8.30% about *Maintainability* and 16.70% about *Portability*.

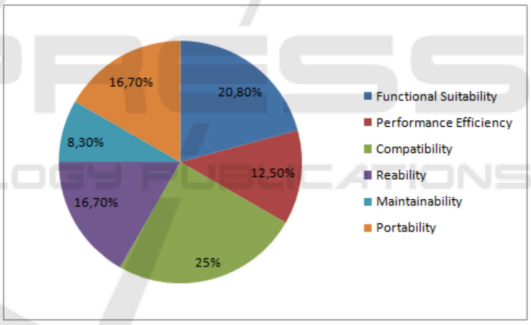


Figure 8: Findings from Eclipse Issues.

4.3.2 Mozilla Firefox

For Mozilla Firefox, the following filter was defined to query the bugs: *(Status: Resolved, Closed; Resolution: Fixed; Priority: P1; Severity: Blocker, Critical; Product: Firefox)*. The focus was on the following quality attributes: *(Functional Suitability (FS), Performance Efficiency, Compatibility, Usability, Reliability, Security, Maintainability and Portability)*

This query returned 83 issues. Analysis of the list led to 9 issues that had a relation with the selected quality attributes. One example is issue 1310554[6], which has the following description "*updateChildIndex in PlacesSyncUtils is extremely inefficient*" and

[5] https://goo.gl/47eHGz
[6] https://goo.gl/wnca9g

was registered on the 16th of October 2016, allowing the identification of issues related to the product's efficiency.

When analyzing the bug report to obtain more details, we found a connection with *Efficiency*, since the user pointed out that having a few markers inside a folder causes Firefox to make a call for each marker, each call lasting between 5 and 10 seconds *"If you have 10000 bookmarks in a single folder, applying those records takes many hours, during which Firefox is completely unresponsive. Instrumenting PlacesSyncUtils shows this is called 10000 times, with each call taking around 5-10 seconds."*.

The dialog between users reveals that on the 24th of October 2016, a modification was carried out so as to simplify the way bookmarks are synchronized and that this update would be available in the current and subsequent versions.

Figure 9 shows the distribution between attributes. Of the 9 issues returned, 22.20% were about *Performance Efficiency (PE)* problems, 22.20% *Usability*, 22.20% *Security* and 33.40% relate to *Maintainability*.

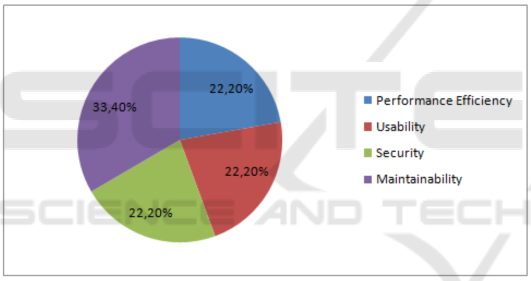


Figure 9: Findings from Mozilla Firefox Issues.

4.3.3 Libre Office

For Libre Office, the following bug filter was defined *(Status:Resolved, Closed; Resolution:Fixed; Priority: Highest; Severity:Blocker, Critical)*. Focusing on the following quality attributes: *(Functional Suitability (FS), Compatibility, Portability, Usability and Maintainability)*

As a result of this query, we obtained 207 bugs. The corresponding descriptions were read with a focus on the quality attributes suggested by the participants. If the text had a section that could be associated with the qualities attributes indicated, this issue would be analyzed more closely. For example in issue 84752[7] says: *"Document With Form Controls Unusable Speed"*. This section calls into attention the terms "unusable" and "speed", which suggest a relation among the attribute *Usability*. When we accessed the issue to obtain more information, we learned that when the user made the upgrade, the execution of some controls became so much slower as to render the system almost unusable. Another example is the following issue description: *"We have upgraded from 4.2 to 4.3 and found that documents with form controls are too slow to use."*. This issue exposes a connection with the *Usability* attribute.

Table 4: Summary of the delivered issues in phase 1.

		Quality Atribute	ID	Release Interval
Phase 1	Eclipse	Portability	290182	1(4.6.1 - 4.6.2)
		Portability	187062	6(4.4.0 - 4.6.0)
		Portability	498196	1(4.6 - 4.6.1)
		Compatibility	228109	2(3.3.2 - 3.4)
		Compatibility	222885	5(3.4 - 3.5M2)
		Portability	234307	2(3.4- 3.4RC2)
		Portability	232641	2(3.4 - 3.4RC2)
		Functional Suitability	250946	5(3.4 - 3.5M2)
		Performance Efficiency	234718	2(3.4 - 3.4RC2)
	Firefox	Portability	1242901	9(43 - 52)
		Performance Efficiency	1260850	6(45 - 51)
		Portability	1287823	4(48 - 52)
		Maintanability	1308840	3(49 - 52)
		Maintanability	959567	1(42 - 43)
	Libre Office	Functional Suitability	74104	1(4.2.0 - 4.2.1)
		Functional Suitability	78801	2(4.2.4 - 4.2.6)
		Usability	95709	13(5.0.3 - 5.2.0)
		Usability	41169	34(3.4.3 - 4.2.0)
		Functional Suitability	62038	19(4.0.0 - 4.2.5)
		Usability	66924	1(4.1.0 - 4.1.1)
		Usability	96742	0(5.2.0 - 5.2.0)

Finally, when analyzing the issue list we detected 18 issues that had a relation with the selected quality attributes. Figure 10 depicts how the distribution between attributes. From the selected issues 41.17% were about *Usability*, 29.41% *Functional Suitability (FS)* problems, 11.76% *Compatibility*, 11.76% *Portability* and 5.88% relate to *Maintainability*.

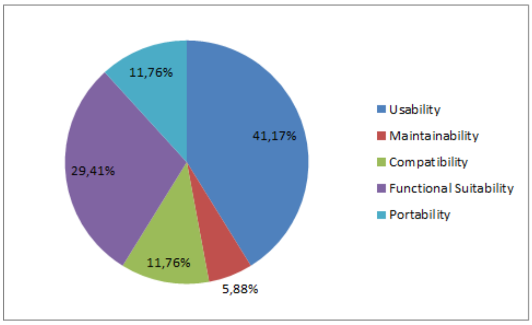


Figure 10: Findings from Libre Office Issues.

4.3.4 Delivered Issues in Releases

Finally, the analysis based on the issues indicated in phase 1 and 3 was carried out. In this analysis, it was

[7] https://goo.gl/4Eks9m

Table 5: Summary of the delivered issues in phase 3.

		Quality Atribute	ID	Release Interval
Phase 3	Eclipse	Compatibility	308364	5(1.0-1.0M5)
		Compatibility	363334	6(4.2-4.2M6)
		Compatibility	74341	2(3.0-3.1)
		Compatibility	438505	5(4.4-4.5RC2)
		Compatibility	28727	4(2.1-2.1M4)
		Functional Suitability	479435	12(4.5-4.6M4)
		Functional Suitability	294650	20(3.5.1-3.7M7)
		Functional Suitability	64540	1(3.0-3.0RC1)
		Functional Suitability	63296	1(3.0-3.0RC1)
		Functional Suitability	14291	5(2.0-2.0M5)
		Reability	335263	1(3.6.1-3.6.2)
		Reability	97148	2(3.1-3.1RC2)
		Reability	16208	6(2.0-2.0M6)
		Portability	184433	1(3.3-3.3RC1)
		Performance Efficiency	129979	6(3.2-3.2M6)
		Performance Efficiency	52982	7(3.0-3.0M7)
		Performance Efficiency	27264	4(2.1-2.1M4)
		Maintainability	270744	7(3.5-3.5M7)
		Maintainability	179210	2(3.3-3.3RC2)
	Firefox	Maintainability	1261651	0(48-48)
		Maintainability	1211016	0(42 - 42)
		Performance Efficiency	1310554	0(51-51)
		Performance Efficiency	1303611	1(51-52)
		Security	1215885	0(44-44)
	Libre Office	Usability	84752	3(4.3.0-4.3.3)
		Usability	72647	0(4.2.0-4.2.0)
		Usability	73464	1(4.2.0-4.2.1)
		Usability	89873	0(4.5.0-4.5.0)
		Usability	47368	6(3.5.1-3.6.0)
		Usability	73243	7(4.2.0-4.3.0)
		Usability	72741	13(4.1.0-4.2.5)
		Functional Suitability	43868	0(3.4.5-3.4.5)
		Functional Suitability	55493	4(3.6.0-3.6.4)
		Functional Suitability	76949	1(4.2.3-4.2.4)
		Functional Suitability	36982	6(3.4.0-3.4.6)
		Compatibility	38452	7(3.4.0-3.5.0)
		Compatibility	64490	30(3.5.4-4.4)
		Portability	69517	0(4.2.0-4.2.0)
		Portability	92041	0(5.0.0-5.0.0)

taken into account if the issues were implemented/ resolved in the following releases relative to their registration date. This way you could find some statement of priority in the release planning according to the indicated quality attribute. The strategy adopted at this stage was to find through the information provided in the issues the version that was found the error / fit and also in which version this was made available. With this information we access the project portal to find information about project releases and identify how many versions have been deployed until the issue correction is made available. This factor we are calling the interval between releases.

Table 4 presents a summary of the findings in phase 1 issues. Table 5 presents the findings of the issues in phase 3, which show the attributes of quality, issues id, and interval between the version that was made available and the corresponding fix made[8].

4.4 Comparison of the Results

Figures 11, 12 and 13 presents the issue counts found in the three phases, for the three projects under study.

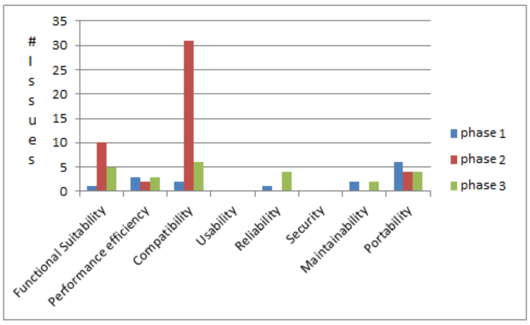


Figure 11: Software product quality attributes prioritized in Phase 1, 2 and 3 in Eclipse.

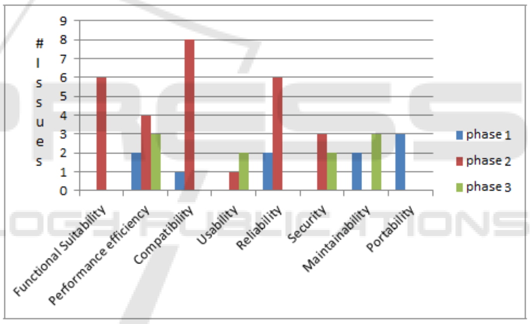


Figure 12: Software product quality attributes prioritized in Phase 1, 2 and 3 in Mozilla Firefox.

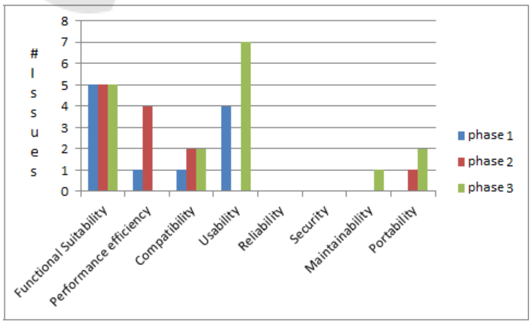


Figure 13: Software product quality attributes prioritized in Phase 1, 2 and 3 in Libre Office.

For Eclipse, *Compatibility* was the quality attribute that came most to the fore. Especially in phase 2 of the study, a possible justification is due

[8] These issues are available at https://goo.gl/LdkXwJ

to being a tool that integrates several other tools, and runs in multiple versions of several operating systems. Eclipse and the projects created around it must be compatible and able to adapt to heterogeneous infrastructure (hardware, operating systems, system versions, etc). In addition, *Performance Efficiency (PE)* presented similar results in the 3 phases of the study.

In Mozilla Firefox, the attributes with most issues raised were *Compatibility* and *Performance Efficiency*. However, while *Compatibility* had a significant number of issues in phase 2, *Performance Efficiency* presented more homogeneous results in the three phases. This may be justified by the fact that Firefox is a Web Browser and must be compatible with different technologies used in portals and be able to be installed on various operating systems (*Compatibility*). All this must take place as transparently to the user as possible, without problems of slowness or inoperability arising. The performance of the program is fundamental, because the competition in the market is fierce as regards performance (*Performance Efficiency*).

For Libre Office, *Functionality* was the attribute indicated most often. It is present in all three phases. *Usability* also comes to the fore, although no issues were found in the second phase. One thing that may explain the relative prominence of these attributes is that it is strongly user-oriented, as well as being an office suite, which demands characteristics intrinsic to its application domain. Therefore, it needs to provide a significant number of functionalities, in ways meet the (demanding) expectations of its users. In addition, it must be easily approached by non-technical users and have an attractive and modern graphical interface, so that its use becomes intuitive.

5 CONCLUSION

This paper presented an exploratory study to investigate the influence of target product quality attributes in software release practices of OSS Projects. We targeted on four research questions to search for evidence that show the relationships among the target attributes, the priorities assigned to the registered issues and how they are delivered by product releases. With this initiative we aim at understand how OSS projects perform to align their goals with their users needs.

In this way, it was possible to verify that this prioritization is strongly influenced by the profile and objective of the project, however it was not possible to conclude if a certain quality attribute has priority in the planning of releases of these projects.

5.1 Threats to Validity

In this study three limitations were founded. The first can be assigned to the reduced number of users that participate in the study. The results of the sample may have been influenced because only seven users participated. A tutorial with a test project was presented, with a view to reduce this risk, in the hope of filling possible gaps that could occur in the data taken from the public repositories. The second threat was a possible misinterpretation of the requested activity. In order to minimize this threat, the authors were available to answer any doubts that may be necessary during the execution of this activity. The last threat concerns to the identification of product quality attributes and issues from Eclipse, Mozilla Firefox and, Libre Office projects. To mitigate this risk, all issues found were discussed to make more reliable quality attributes characterized for the selected projects.

5.2 Work in Progress

As ongoing work, we are know replicating this study with the support of machine learning techniques. The goal is to collect and analyze a larger amount of data to confirm the software quality attributes considered as priority for each project.

REFERENCES

Adams, B., Kavanagh, R., Hassan, A. E., and German, D. M. (2015). An empirical study of integration activities in distributions of open source software. *Empirical Software Engineering*, pages 1–42.

Bijlsma, D., Ferreira, M. A., Luijten, B., and Visser, J. (2012). Faster issue resolution with higher technical quality of software. *Software quality journal*, 20(2):265–285.

Crowston, K., Annabi, H., and Howison, J. (2003). Defining open source software project success. *ICIS 2003 Proceedings*, page 28.

da Costa, D. A., Abebe, S. L., McIntosh, S., Kulesza, U., and Hassan, A. E. (2014). An empirical study of delays in the integration of addressed issues. In *ICSME*, pages 281–290.

Fitzgerald, B. (2006). The transformation of open source software. *Mis Quarterly*, pages 587–598.

Gamalielsson, J. and Lundell, B. (2014). Sustainability of open source software communities beyond a fork: How and why has the libreoffice project evolved? *Journal of Systems and Software*, 89:128–145.

Gonzalez-Barahona, J. M., Izquierdo-Cortazar, D., Maffulli, S., and Robles, G. (2013). Understanding how companies interact with free software communities. *IEEE software*, (5):38–45.

Gonzalez-Barahona, J. M. and Robles, G. (2013). Trends in free, libre, open source software communities: From volunteers to companies. *it–Information Technology it–Information Technology*, 55(5):173–180.

Henningsson, K. and Wohlin, C. (2002). Understanding the relations between software quality attributes-a survey approach. In *Proceedings 12th International Conference for Software Quality*. Citeseer.

ISO, I. (2011). Iec 25010: 2011,“. *Systems and Software Engineering—Systems and Software Quality Requirements and Evaluation (SQuaRE)—System and Software Quality Models*.

ISO/IEC (2001). *ISO/IEC 9126. Software engineering – Product quality*. ISO/IEC.

Jonsson, L., Borg, M., Broman, D., Sandahl, K., Eldh, S., and Runeson, P. (2015). Automated bug assignment: Ensemble-based machine learning in large scale industrial contexts. *Empirical Software Engineering*, pages 1–46.

Mantyla, M. V., Khomh, F., Adams, B., Engstrom, E., and Petersen, K. (2013). On rapid releases and software testing. In *Software Maintenance (ICSM), 2013 29th IEEE International Conference on*, pages 20–29. IEEE.

Michlmayr, M., Fitzgerald, B., and Stol, K.-J. (2015). Why and how should open source projects adopt time-based releases? *Software, IEEE*, 32(2):55–63.

Offutt, J. (2002). Quality attributes of web software applications. *IEEE software*, 19(2):25.

Oreg, S. and Nov, O. (2008). Exploring motivations for contributing to open source initiatives: The roles of contribution context and personal values. *Computers in human behavior*, 24(5):2055–2073.

Perepletchikov, M., Ryan, C., and Tari, Z. (2005). The impact of software development strategies on project and structural software attributes in soa. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 442–451. Springer.

Pfleeger, S. L. (1995). Experimental design and analysis in software engineering. *Annals of Software Engineering*, 1(1):219–253.

Rigby, P. C., Cleary, B., Painchaud, F., Storey, M.-A., and German, D. M. (2012). Contemporary peer review in action: Lessons from open source development. *Software, IEEE*, 29(6):56–61.

Stol, K.-J. and Fitzgerald, B. (2015). Inner source–adopting open source development practices in organizations: A tutorial. *IEEE Software*, (4):60–67.

Thomas, L., Schach, S. R., Heller, G. Z., and Offutt, J. (2009). Impact of release intervals on empirical research into software evolution, with application to the maintainability of linux. *Software, IET*, 3(1):58–66.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., and Wesslén, A. (2012). *Experimentation in software engineering*. Springer Science & Business Media.

Ye, Y. and Kishida, K. (2003). Toward an understanding of the motivation of open source software developers. In *Software Engineering, 2003. Proceedings. 25th International Conference on*, pages 419–429. IEEE.

Yin, R. K. and Campbell, D. (2003). Case study research: Design and methods. applied social science research methods series, vol. 5.