On the Impact of Product Quality Attributes on Open Source Project Evolution

António César B. Gomes da Silva
Glauco de Figueiredo Carneiro
Salvador University (UNIFACS)
Salvador, Bahia, Brazil
antoniocesar01@gmail.com
glauco.carneiro@unifacs.br

Miguel Pessoa Monteiro
Universidade Nova de Lisboa (UNL)
NOVALINCS/UNL
Lisbon, Portugal
mtpm@fct.unl.pt

Fernando Brito e Abreu
Instituto Universitário de Lisboa (ISCTE-IUL)
ISTAR-IUL
Lisbon, Portugal
fba@iscte-iul.pt

Kattiana Constantino
Eduardo Figueiredo
Federal University of Minas Gerais (UFMG)
Minas Gerais, Brazil
kattiana@dcc.ufmg.br
figueiredo@dcc.ufmg.br

Abstract—**Context: Several Open Source Software (OSS) projects have adopted frequent releases as a strategy to deliver both new features and fixed bugs on time. This cycle begins with 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 a still open question is the set of criteria and priorities that influence the decisions of which issues should be analyzed, implemented/solved and delivered in next releases. In this paper, we present an exploratory study whose goal is to investigate the influence of target product quality attributes in software evolution practices of OSS projects. The goal is to search for evidence of relationships between these target attributes, priorities assigned to the registered issues and the ways they are delivered by product releases. To this end, we asked six participants of an exploratory study to identify these attributes through the data analysis of repositories of three well-known OSS projects: Libre Office, Eclipse and Mozilla Firefox. Evidence indicated by the participants suggest that OSS community developers use criteria/priorities driven by specific software product quality attributes, to plan and integrate software releases.**

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

I. Introduction

The attractiveness of Open Source Software (OSS) projects for both the users' and developers' communities has aroused the interest of Software Engineering researchers [1]. OSS development is a relevant context to study, but it differs from proprietary development in aspects such as development processes, team structure, and developer incentives [2]. Understanding the rationale behind successful OSS initiatives may allow teams of non-OSS projects to reuse best practices on their own projects [3][4].

According to previous work [5], there are more than 400 active OSS distributions, and each year 26 new projects on average 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 on larger periods (e.g. quarterly or yearly) is typically referred to as a traditional release cycle [6]. The adoption of frequent releases in OSS projects takes into account that these projects have usually users all over the world, who eagerly download each new version as soon as it is released, and test it as thoroughly as they can. The worldwide dispersion of users often leads to a continuous (24 hours a day) testing process [7]. The process of reporting and resolving issues for a system during its development and/or maintenance is usually supported by an Issue Tracking System(ITS). Typically, an ITS can record the issue type (e.g. defect, enhancement, patch, task), its state (e.g. new, assigned, resolved, closed), the date of submission and of each state change, who was its submitter, was was assigned to address it, any comments by others, and indications of severity and/or priority [8].

The community members usually vote for decisions regarding the demands registered in the ITS [9]. However, a still open question is the set of criteria and priorities that influence the decisions of which issues should be analyzed, implemented/solved and delivered in next releases, to the detriment of others. This question is a relevant aspect for the success of such projects. We conducted an exploratory study to investigate the influence of target product quality attributes in software release practices of OSS projects. The goal is to look for evidences on the relationships among these target attributes, the priorities assigned to the registered issues and how they are delivered by product releases.

The rest of this paper is organized as follows: section II presents some background and related work; section III outlines our exploratory study; section IV describes the collected data and its analysis; finally, section V discusses the results of the study, identifies their validity threats and scopes future research on this topic.

II. Background and Related Work

The ISO/IEC 25010 international standard for software product quality [10], that replaced the ”old” ISO/IEC 9126 standard [11], provides a quality model composed of several software quality characteristics, that are further broken down into many sub-characteristics, as represented in figure 1. For instance, one of the main characteristics is maintainability, which is further broken down into modularity, reusability, analysability, modifiability and testability. Reusability, for example, is related to Software Reuse, which is defined as the process of building or assembling software applications and systems from previously developed software [12]. Both ISO standard families are based on the model, that the quality of the process influences the internal quality of the software product, which in turn influences its external quality, and then its quality in use [13]. Studies have reported the relevance and influence that software quality attributes have in the software development cycle of a product [14][15]. This influence is also true for OSS projects. Two surveys in this direction were conducted by Henningsson and Wohlin [16]. 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 between many of the software quality attributes.

III. 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 [17]. 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 evidences in data provided by the repositories of the selected OSS projects. This section aims to address the following Research Questions:

RQ1- *Considering data available in public repositories, is it possible to identify at least three software product quality attributes of the OSS Projects Mozilla Firefox, Libre Office and Eclipse that are prioritized in software release practices?* The identification of software quality attributes in OSS projects is a key to understand their priorities throughout software releases. 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 evidence from the OSS project repositories support the indication of the tree prioritized software quality attributes?* These evidences are relevant lessons learned that can be referenced and followed by industry practitioners, as well as by other OSS projects.

To answer these questions, we defined the protocol of our study (III-C) and instantiated it in six study replications (IV-A to IV-F). This protocol is based on the analysis of data that have relationship with bug issues of three OSS projects (III-B). Prior to the tasks execution, one of the authors conducted a tutorial session (III-D) to help participants find out evidences through the use of repositories provided by the selected OSS projects.

A. Data Collection

Data collection. Data collection was carried out directly from answers provided to the questionnaire given to participants. To answer RQ1, we collected information regarding quality attributes and corresponding justifications. To answer RQ2, we collected evidence based on references provided by participants, e.g., print screens and URLs from which relevant information is accessible.

Passing score. After the tutorial, we analyzed data provided by the participants in accordance with the following two criteria. The *Criterion 1* was the 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. The *Criterion 2* was the indication of evidence from the repositories supporting the choice of the selected quality attribute.

Data analysis. Data from the questionnaires were analyzed in search of the primary strategies employed by participants to answer RQ1 and RQ2.

B. Target OSS Projects

This study relies on three highly regarded open source projects presented: *Mozilla Firefox*, *Eclipse*, and *LibreOffice*. We selected those projects due to several reasons. First, they adopted frequent releases implementation [18] [19]. Second, they have a very active developer community comprising 290 (*LibreOffice*), 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 were previously cited in the literature, which enhances their value as the basis for the present study.

C. The Study Protocol

Six participants took part in this study to answer research questions **RQ1** and **RQ2**. They were also asked to register the collected evidences from the analysis they performed and the time they were identified 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. To answer **RQ1**, we analyzed and compared the attributes indicated by each participant. Additionally, we analyzed the data provided by the participants to identify evidences to justify the attributes so far indicated to answer **RQ2**.

D. Participants Selection

The study involved six participants recruited from a MSc program in Computer Science. This number of participants offered a reasonable trade-off between the effort to plan and

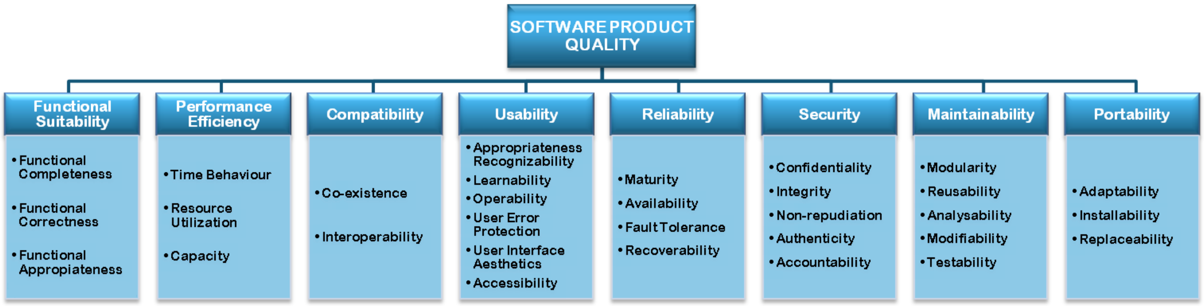

Fig. 1: The ISO/IEC FCD 25010 product quality standard

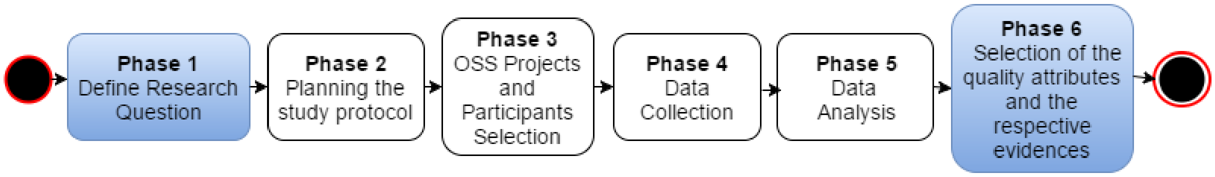
Phase 1
Define Research Question
Phase 2
Planning the study protocol
Phase 3
OSS Projects and Participants Selection
Phase 4
Data Collection
Phase 5
Data Analysis
Phase 6
Selection of the quality attributes and the respective evidences

Fig. 2: Overview of the exploratory study

execute the study and detailed qualitative analysis and the generalizability of the results [20] [21]. They were all volunteers and no compensation was provided for their participation in this study.

We were interested in making observations based on a detailed, qualitative analysis of OSS projects repositories, regarding the influence of software product quality attributes on the scheduling of pending issues, rather than testing causality hypotheses using statistical inference. To be eligible for inclusion, participants filled out a pre-study questionnaire (Table I) to describe their profile and experience in software release practices, OSS projects and software quality product attributes. In Table II, we present the answers of each question presented in Table I, using the following ordinal scale: none/low/medium/high.

TABLE I: Pre-study questionnaire

ID	Question	Answers Options
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

TABLE II: Results of the pre-study questionnaire

ID	P1	P2	P3	P4	P5	P6
OSS-1	None	Medium	Low	High	Medium	Low
OSS-2	None	None	None	None	None	None
SRP-1	Medium	Low	Low	High	Medium	Low
SRP-2	Medium	Low	Low	High	Medium	Medium
SRP-3	Medium	Medium	Low	High	Medium	None
QA-1	Low	Medium	Medium	Medium	Low	Low
QA-2	Low	Medium	Medium	Low	None	Medium
QA-3	Low	Medium	Low	None	None	None

The Tutorial Session. Prior to the tasks, the participants attended a tutorial session focusing on how to search for evidences in the *Open Stack IAAS Cloud Platform* project using the data repositories of several tools used during its development: *Bugzilla* (issue tracking system), *Git* (configuration management system), *Gerrit* (code review tool for Git) and a *Wiki*. These tools were also used in the OSS projects selected 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 within the aforementioned tool repositories for the Open Stack project. The second part focused on using the selected repositories to identify evidences that could establish relationship with corresponding software product attributes prioritized by the Open Stack IAAS Cloud Platform.

One of the first indicators of the connection between the bugs and release planning was found in the interaction between users to request a given fix in the next patch. [1] These bugs also help to identify instances of reuse, through access to the *Gerrit* link provided by one user. Through it, we were able to identify the creation of a method override. Use of *Gerrit* also meets the modifiability requirement, as it shows the code before and after the bug fix. [2]

IV. Data Analysis

This section presents the analysis of data provided by the participants to answer the two research questions stated above.

A. *Participant 1*

Data provided by Participant 1 were discarded because criteria 1 and 2 were not met. This decision can be partly

[1] https://bugzilla.redhat.com/show_bug.cgi?id=1283721

[2] https://review.openstack.org/#/c/346796/8/puppet/services/ceph-mon.yaml

explained by the profile of participant 1 presented in Table II. He/she was not familiar with software quality product attributes, neither from the theoretical, nor from the practical perspective. Probably for this reason, this participant could not provide consistent evidences regarding quality attributes, taken from the projects' data repositories.

B. Participant 2

Participant 2 indicated *Portability* and *Efficiency* as quality attributes for *Mozilla Firefox*. According to Participant 2, the indication of *Portability* was related to the need of the *"the Web browser to be compatible with different technologies and also run in different operating systems"*. The manifesto of *Mozilla Firefox* states that the effectiveness of the Internet as a public resource depends upon interoperability (protocols, data formats, content), innovation and decentralized participation worldwide. [3] And the project seems to follow this principle. Regarding *Efficiency*, this participant found comments registered in Bugzilla issues, comparing *Mozilla Firefox* with *Chrome*, one of its main competitors. The following one, produced by one of the developers, is a good corroborating example: *"significant problem: Firefox is taking an extremely long time to load compared to Chrome because our load time often includes the restoration of a large number of tabs"*. [4]

This participant also identified the following attributes for the *Eclipse* project: *Portability*, *Reliability* and *Efficiency*. The following sentence justified the selection of these three attributes: *"it is an open source tool that provides integration with various other tools. It runs on various versions of operating systems (Portability). Reliability is also a concern of Eclipse community during its operation. To deal with this issue, tests are planned and executed to find failures in an attempt to fix them in case they occur"*. Finally, the participant justifies the choice of *Efficiency* with this sentence: *"Response time and resource consumption are also important factors to Efficiency and they are prioritized in corresponding issues that address this subject"*. An example of concern with *Portability* is illustrated in the following issue excerpt, where one developer writes to another developer *"take a look at this problem, which is specific to Linux"*. [5]

The same participant identified the attributes *Functionality* and *Usability* for the *LibreOffice* project. The reason for the indication of these attributes was that *"it is an office suite software, which requires characteristics intrinsic to its application domain and strongly geared to the user needs, i.e. it must provide features that meet the expectations of its users, must be easily understood (easy to use) and have an attractive and modern graphical user interface, contributing to an intuitive and friendly use"*. The participant also identified that issues related to these attributes were marked in the project bug repository as *high priority* for *LibreOffice*. For example, the following issue reports an error related to find and replace functionality. During the developer's interaction it was mentioned that the problem occurred in a specific version: *"this still affects 4.2.0.4, which is now an official release and will be available in LibreOffice 4.2.1"*.[6]

C. Participant 3

Participant 3 proposed *Security*, *Usability* and *Portability* as attributes for the *Mozilla Firefox* project. The participant observed that *"Firefox is a web browser available for many platforms, including Windows, OS X, Linux, Android and iOS and compatible with the up-to-date Web technologies"*. The participant presented a list conveying the distribution of issues registered in the bug repositories through their several possible status (assigned, closed, new, reopened, resolved, unconfirmed and verified). Considering the status and average bugs solutions of the *Mozilla Firefox* project, the participant concluded that expert users usually mark issues related to *Usability* (Tabbed Browser Toolbar and Customization), *Portability* (Extension compatibility), and *Security* (session restore, disability access) as high priority in the project.

For the *Eclipse* project, *Functionality*, *Compatibility* and *Maintainability* were proposed as quality criteria. According to the participant *"Eclipse is a complete multi-platform environment, compatible with different programming languages (C, Java, PHP, etc.) and operating systems (Windows, Linux, iOS, etc.)"*. As evidence, the participant presented a table containing overall statistics on the status and priorities assigned to the solution of *Eclipse* project issues. This table reveals that the highest priorities are given to components of the SWT UI and Core. This prioritizing also takes into account the percentage of issues related to these components.

For the *LibreOffice* project, the selected attributes were *Functionality*, *Usability* and *Maintainability*. The participant mentioned that *"Libre Office is a powerful office suite with a clean interface and rich in productivity tools. It includes several applications, such as Writer (Word processing), Calc (spreadsheet), Impress (presentations), Draw (vector graphics and flowcharts), Base (database) and Math (edit formula). It runs on Linux, iOS, Android and Windows"*. Similarly to the *Eclipse* project, the participant also presented a table containing overall statistics on the status and priorities assigned to the solution of the *LibreOffice* issues. Upon this table, the participant highlighted that *"it was clear that according to data from the last 500 recorded bugs, the terms most frequently used among the highest priority include: crash, merge, update, document, displayed, Calc, Wizard"*.

D. Participant 4

Portability, Usability and *Efficiency* were proposed as quality attributes for *Mozilla Firefox*. The participant justified that the community is concerned with the above quality criteria - *"Firefox for desktop - Reliable, flexible, quick"; "Firefox for iOS - Fast, intelligent, Yours"; "Committed to you, your privacy and an open Web"; "you can modify the Firefox according to your needs"; "Save your time and do everything faster"*. The main evidence pointed out by the participant was a bug referring to a version of *Mozilla Firefox*, which occurred when transferring an image saved by the browser to the download folder. This behavior was evidenced only in Linux as stated *"seems to be reproducible only on Linux"*. [7]

For the *Eclipse* project, *Portability, Reliability* and *Compatibility* were suggested. The justification betrays a concern,

[3] https://www.mozilla.org/en-US/about/manifesto/

[4] https://bugzilla.mozilla.org/show_bug.cgi?id=664314

[5] https://bugs.eclipse.org/bugs/show_bug.cgi?id=290182

[6] https://bugs.documentfoundation.org/show_bug.cgi?id=74104

[7] https://bugzilla.mozilla.org/show_bug.cgi?id=1287823

on the part of the community, in meeting the above software quality criteria. "*Eclipse* provides IDEs and platforms for almost all languages and architectures"; "IDEs built on extensible platforms for creating desktop, web and cloud IDEs"; "Develop your software wherever you go". For this participant, the most prominent bug was evidenced as *"Toolbar does not display custom widgets properly"*. This problem occurred in a newly released version of Eclipse Neon and on the Windows platform, as shown in *"I also tested it on OS X. Seems not a problem there, so I guess this issue is limited to the Windows platform"*. [8]

For *LibreOffice*, *Usability* and *Compatibility* were proposed as relevant quality criteria. As a justification, the participant cites that the system *"includes several applications that make it a powerful free office suite and open source, is compatible with a wide range of document formats such as Microsoft Word, Excel, PowerPoint and Publisher. In addition, it has a modern and open standard, the OpenDocument Format (ODF)"*. From evidences presented, we highlight the concern of the developers with the following bug *"Multiple animated GIFs cause 100% CPU utilization in Impress"*. To fix this bug, a code adjustment was made to improve CPU performance, as presented in *"To reach more with the current approach we would have to re-implement AnimatedGIF import"*. [9] [10]

E. Participant 5

For the *Mozilla Firefox* project, *Maintainability* and *Reliability* were proposed. The participant provided the following evidence: this project "provides a platform for presentation of bugs by the developers as well as their status. It uses a version release system in which versions are released after undergoing reliability tests". The main evidence presented to justify *Maintainability* was the URL of the project's Bugzilla.

For the *Eclipse* project, *Interoperability* and *Maintainability* were proposed. The following quote was provided as justification: "Eclipse has a high range of compatibility with various development platforms. It boasts a well-designed and well-organized platform for submitting bugs to its developer". The evidence presented to the *Maintainability* was the URL of the Bugzilla project. For *Interoperability*, the following URL from *Eclipse*[11] was presented.

For *LibreOffice*, *Maintainability* and *Usability* were proposed. The choice is justified by pointing that "it features a well-designed and well organized platform for presenting bugs to developers. It encourages users to participate on the improvement the software, through feedbacks. The *LibreOffice* site itself has a Community tab for the community participation". The evidence presented to the *Maintainability* was the URL of the Bugzilla project's. For *Usability*, the participant presented the URL of *LibreOffice*, which highlights the information explaining how new users can participate and interact with the project's products. [12]

F. Participant 6

For *Mozilla Firefox*, *Reliability* and *Maintainability* were proposed. As justification, "reliability because there is a study specifically for each release version of the software, each release version is kept separate from the others according to the reliability users have the resources belonging to each version, after they have been tested". *Maintainability* because "Mozilla Firefox provides an infrastructure for the management and fixing of bugs according to the versions of the system". The evidence presented to justify *Maintainability* was an issue [13] which a developer making the adjustment removes the comment from the code claiming that the code was self-explanatory *"Because it is no longer relevant. We make the message collapsible if the 'collapsible' proposes truth, and I find the code self-explanatory here"*. However, another developer reports that this is not a good excuse and asks for the re-addition of the comment since *"someone who is unfamiliar with the code can scan it more easily"*.

For *Eclipse*, *Reliability* and *Maintainability* were proposed. As justification, "Compatibility because Eclipse is an open source software with a focus on developing an extensible platform for creating of new projects. *Maintainability* because *Eclipse* provides access to bugs and improvements proposed by its users, and to development code". The evidence presented to justify *Maintainability* in the *Eclipse* project was a high priority bug fix cycle. [14] The fix was very quick during this cycle.

For the *LibreOffice* project, *Maintainability* and *Usability* were suggested as relevant quality attributes. Justification states that *LibreOffice* is "an open source software project that encourages its users, be they developers or not, to test the system so that they contribute to the quality improvement, providing them back information about its use". *Maintainability* is justified because "the official LibreOffice site provides an environment conducive to the identification of system bugs and control of the development of fixes for those bugs". Evidence cited regarding *Usability* was a sample of the project URLs that encourage users to give their opinion on the *LibreOffice* programs. For *Maintainability*, the URL of the project's Bugzilla is presented, along with the information that this infrastructure describes the bugs raised by developers, their degree of seriousness and status report to help monitoring each bug fix. The site also invites users to participate. [15]

V. Conclusions

To point out the selected quality attributes, participants 2, 4 and 6 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 5 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

[8] https://bugs.eclipse.org/bugs/show_bug.cgi?id=498196

[9] https://bugs.documentfoundation.org/show_bug.cgi?id=98500

[10] https://cgit.freedesktop.org/libreoffice/core/commit/?id=285744fef87f4ca0278834b97d7f618bdba5f4c0

[11] http://www.eclipse.org/home/newcomers.php

[12] https://www.libreoffice.org/community/get-involved/

[13] https://bugzilla.mozilla.org/show_bug.cgi?id=1308840

[14] https://bugs.eclipse.org/bugs/show_bug.cgi?id=505535

[15] https://www.libreoffice.org/community/developers/

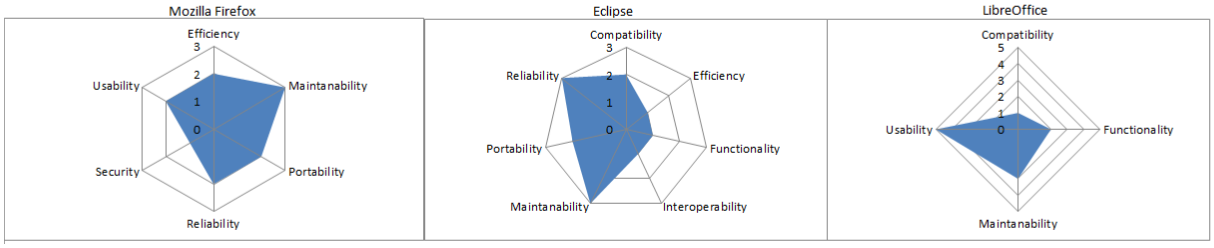

Fig. 3: Software product quality attributes prioritized on releases according to participants

through portals and wikis, does not necessarily reflect the relationship of these attributes to the release schedules of those projects. Finally, participant 3 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.

Figure 3 depicts the influence of the software product quality attributes on releases according to the participant's perception. The maintainability attribute is the common attribute in the three projects. Moreover in the LibreOffice project, usability was quoted by all the participants, due to the concern of the developer's community with the ease of use of its products by the users, and the strong competition with proprietary and open source products available in the market.

A. Threats to validity

Two possible limitations were identified in this exploratory study. The first threat that can be pointed out is the small number of users participating in the study. Since only 6 students were involved, the dimension of the sample may have an influence on the results obtained. To reduce this risk, a tutorial with a test project was presented, in the hope of filling possible gaps that could possibly occur in the data taken from public repositories. Another potential threat was a possible misinterpretation of the requested activity. To mitigate this risk, the authors were available to answer any doubts that could have arisen during execution of this activity.

B. Future works

Considering the attributes selected by the participants of this study and presented in Figure 3, it was possible to identify evidence of the relationship of these target attributes with priorities of registered issues to be fixed and or implemented in next immediate product releases. As ongoing work, we are extending this study with experienced developers in order to contrast with results of this study.

References

[1] M. Michlmayr, B. Fitzgerald, and K.-J. Stol, "Why and how should open source projects adopt time-based releases?" *Software, IEEE*, vol. 32, no. 2, pp. 55–63, 2015.

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

[3] K.-J. Stol and B. Fitzgerald, "Inner source–adopting open source development practices in organizations: A tutorial," *IEEE Software*, no. 4, pp. 60–67, 2015.

[4] P. C. Rigby, B. Cleary, F. Painchaud, M.-A. Storey, and D. M. German, "Contemporary peer review in action: Lessons from open source development," *Software, IEEE*, vol. 29, no. 6, pp. 56–61, 2012.

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

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

[7] L. Thomas, S. R. Schach, G. Z. Heller, and J. Offutt, "Impact of release intervals on empirical research into software evolution, with application to the maintainability of linux," *Software, IET*, vol. 3, no. 1, pp. 58–66, 2009.

[8] D. Bijlsma, M. A. Ferreira, B. Luijten, and J. Visser, "Faster issue resolution with higher technical quality of software," *Software quality journal*, vol. 20, no. 2, pp. 265–285, 2012.

[9] K. Crowston, H. Annabi, and J. Howison, "Defining open source software project success," *ICIS 2003 Proceedings*, p. 28, 2003.

[10] I. ISO, "Iec 25010: 2011,," *Systems and Software EngineeringSystems and Software Quality Requirements and Evaluation (SQuaRE)System and Software Quality Models*, 2011.

[11] ISO/IEC, *ISO/IEC 9126. Software engineering – Product quality*. ISO/IEC, 2001.

[12] S. G. Shiva and L. A. Shala, "Software reuse: Research and practice." in *ITNG*, 2007, pp. 603–609.

[13] G. de Souza, R. P. Mellado, D. Á. Montini, L. A. V. Dias, and A. M. da Cunha, "Software product measurement and analysis in a continuous integration environment," in *Information technology: New generations (itng), 2010 seventh international conference on*. IEEE, 2010, pp. 1177–1182.

[14] M. Perepletchikov, C. Ryan, and Z. Tari, "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"*. Springer, 2005, pp. 442–451.

[15] J. Offutt, "Quality attributes of web software applications," *IEEE software*, vol. 19, no. 2, p. 25, 2002.

[16] K. Henningsson and C. Wohlin, "Understanding the relations between software quality attributes-a survey approach," in *Proceedings 12th International Conference for Software Quality*. Citeseer, 2002.

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

[18] D. A. da Costa, S. L. Abebe, S. McIntosh, U. Kulesza, and A. E. Hassan, "An empirical study of delays in the integration of addressed issues." in *ICSME*, 2014, pp. 281–290.

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

[20] S. L. Pfleeger, "Experimental design and analysis in software engineering," *Annals of Software Engineering*, vol. 1, no. 1, pp. 219–253, 1995.

[21] R. K. Yin and D. Campbell, "Case study research: Design and methods. applied social science research methods series, vol. 5," 2003.