INFORMATION AND SOFTWARE TECHNOLOGY

Exploring information from OSS repositories and platforms to support OSS selection decisions

Nesrine Sbai[a], Valentina Lenarduzzi[b], Davide Taibi[b], Sihem Ben Sassi[a,c], Henda Hajjami Ben Ghezala[a]

a RIADI Laboratory, Manouba University, La Manouba, 2010, Tunisia
b Tampere University of Technology, Tampere (Finland)
c MIS Department, College of Business Administration Taibah University, Al-Madinah, KSA

ARTICLE INFO

Received 23 August 2017,
Revised 1 July 2018,
Accepted 13 July 2018,
Available online 21 July 2018

ABSTRACT

Context: Individuals and organizations are increasingly adopting Open Source Software (OSS) for the benefits it provides. Although the OSS evaluation process and the information it requires are nowadays well known, users still have problems finding the right information and are not supported by any decision support system.
Objective: The aim of this study is to bridge the gap between OSS adoption models, especially with the aim of supporting users in evaluating the OSS they are planning to select.
Method: To reach this aim, we studied the processes and the information considered by the major OSS assessment models. Then we carried out a case study to identify which information can be automatically retrieved from the main OSS platforms, namely GitHub, SonarCloud, and StackExchange. Finally, we characterized the maturity of the projects available on these three platforms.
Results: Projects available on the three platforms are commonly old, stable, and mature ones. Moreover, thanks to the API provided, we were able to extract most of the information not commonly accessible from the main website.
Conclusions: Our results confirm that it is possible to develop a decision support system based on these three platforms, and that is also possible to evaluate both the quality and the maturity of the projects available there.

1. Introduction

Nowadays, Open Source Software (OSS) is becoming more and more accepted, and is often considered to have the same quality as Closed Source Software. Despite the free availability of the source code in OSS, its selection is still challenging. OSS users commonly look for OSS projects in repositories such as GitHub* or, when available, on the OSS homepage. Several works have analyzed the OSS adoption process, identifying the information commonly considered by the users [1, 2, 3, 4, 5]. However, this information is not always available in OSS repositories, increasing the uncertainty involved in the adoption of new products [6].
In order to support users during OSS adoption, we are working on a decision support system for selecting new OSS based on a set of automatically collected data.

In this work, we focus on OSS characteristics and data with the goal of understanding which information such a decision support system must rely on. Therefore, the goal of this paper is threefold:

1) to classify the information whose evaluation OSS adoption models recommend and the information that users usually consider as relevant during OSS adoption;
2) to map which of the previously identified information is available by combining the information available in three OSS platforms: GitHub, one of the most important OSS repository; SonarCloud†, a widely used platform for continuously assess software quality; and StackExchange[3]; a well-known and widely used platform for questions and answers;
3) to classify the projects available on the three platforms.

We performed this work as a case study, mapping first the information required by the different OSS evaluation models OpenBRR [1], OSSPAL [7], QSOS [3], OpenBQR [2], Capgemini-OSMM [8], SQO OSS [9] and

* https://github.com/
† https://sonarcloud.io/projects
[3]https://stackexchange.com/

Preprint.

Please, cite as:
Nesrine Sbai, Valentina Lenarduzzi, Davide Taibi, Sihem Ben Sassi, Henda Hajjami Ben Ghezala. “Exploring information from OSS repositories and platforms to support OSS selection decisions” Information and Software Technology, 2018, ISSN 0950-5849, https://doi.org/10.1016/j.infsof.2018.07.009.

QualOSS [10] and then comparing this with the information considered relevant by users during OSS adoption [4, 5]. Then we analyzed the projects available on GitHub, SonarCloud, and StackExchange with the goal of understanding which of this information can be retrieved and which projects are available on the three platforms. Finally, we analyzed the characteristics of the available projects based on the information previously identified (e.g., community size, project, age, and others).

The result of this work can be adopted by researchers and practitioners to understand which information they can automatically retrieve from the online platforms.

2. Case Study Design

In this section we present case study design, following the guidelines proposed by Runeson et al. [11].

The goal of this study was to characterize the information on OSS projects available on GitHub, SonarCloud and StackExchange in order to understand whether the information retrieval process required for OSS selection can be supported by these three platforms. We thus formulated our main research question (RQ) as: "Which information can we extract from OSS platforms, and for which type of projects?", from which we derived the following three research questions:

RQ1: Which information is required during the OSS adoption process?

In this RQ, we aim at classifying the information commonly considered important by OSS users during OSS adoption. The results of this RQ will be then used as input of the next RQ. We analyze both the information suggested by the OSS adoption models and the information considered important from a set of surveys conducted with OSS users.

RQ2: Which information is available on GitHub, SonarCloud, and StackExchange?

In this RQ, we want to understand whether the relevant information obtained in RQ1 is available on these three platforms.

RQ3: To which extent can GitHub, SonarCloud, and StackExchange help to support OSS selection decisions?

In this RQ, we aim at characterizing projects common to the three platforms in order to assess their usefulness and contribution to the OSS selection process. To study RQ3, we derived two sub-questions.

RQ3.1: How many projects are available on all three platforms (GitHub, SonarCloud, and StackExchange)?

GitHub contains more than 10 million repositories, SonarCloud includes more than 4.5 thousand projects while there is no data on the number of projects on which users made questions on StackExchange. Therefore, we do not expect to find all the projects included in GitHub in SonarCloud or StackExchange, mainly because the vast majority of repositories are not related to real projects but to toy samples, sketches, or early development prototypes.

RQ3.2: How old and how large are the projects available on these three platforms?

Since we do not expect to find all the projects available on the three platforms in this RQ we aim at understanding whether these projects are mainly small and new ones or old, large, mature.

2.1. Study context

We analyzed projects freely available in GitHub, SonarCloud and StackExchange (including StackOverflow and all its sub-platforms) as they are among the most popular freely accessible repositories hosting data about OSS.

GitHub is one of the leading OSS software repositories, currently adopted by more than ten million OSS projects ranging from code sketches to early prototypes and mature projects.

SonarCloud is a continuous software quality monitoring platform based on SonarQube, that provides a free online service for OSS projects. It does not directly provide information for OSS selection: however, it provides some information that could be easily combined with that obtained from GitHub.

StackExchange is a well-known network of websites including Stack Overflow and many other sub-platforms, where users provide questions and answers on topics in various fields, including OSS.

2.2. Data collection and analysis

We collected data for RQ1, classifying the information reported in the OSS adoption models [1, 2, 3, 7, 8, 9, 10] and these considered relevant by the users, as reported in the two surveys [4, 5].In this step, we consider all the information reported in the adoption models and in the survey, including quality-related information. This process has been conducted systematically, following the guidelines for systematic mapping studies proposed by Petersen et el [12].

As for RQ2, we checked if the information identified in RQ1 can be automatically extracted from GitHub, SonarCloud and StackExchange. In case of calculated information such as "average number of lines of code per class" we checked the availability of both information "number of lines of code" and "number of classes". For completeness purpose, in the results we list both the information required by the OSS selection models and survey and these available in the OSS platforms.

Considering RQ3, we first extracted projects from SonarCloud and then searched for their availability in GitHub; finally, we extracted the total number of questions and answers available in StackExchange per project. The projects had to be publicly available, with all data available via GitHub and SonarCloud's API. Then, we compared the information required for OSS adoption obtained in RQ1 with the information available on the platforms obtained in RQ2. Finally, we took the total number of projects obtained in our dataset and analyzed their maturity by taking into consideration the age of the projects, its size, and the number of commits and committers. Moreover, we studied the distribution of the projects in our dataset per year.

3. Results

Table 2 reports the results for RQ1 and RQ2, presenting the classification of the characteristics, the information required, and the information that can be obtained online.

The analysis of the information that should be considered by our decision support system (RQ1) shows that different sets of information are considered by each OSS adoption model and by the two surveys [5, 4]. However, when considering the high-level characteristics of the information, a similar set of characteristics is always present, such as community support, availability of documentation, and product quality. The analysis of the availability of this information for RQ2 on GitHub, SonarCloud, and StackExchange shows that it is possible to obtain some information related to the high-level characteristics but not all the sub-information required. The same behavior was seen when comparing the information required by the OSS evaluation models with that considered relevant by the users based on empirical evaluations, where the same set of characteristics is considered, but some information required for evaluation, such as economic factors or license costs, are not available in the platforms.

As for RQ3.1, in July 2017 we retrieved 4503 projects from SonarCloud. After that we searched for matches with GitHub projects and obtained a set of 1638 projects. All the projects identified have questions and answers on StackExchange. Moreover, the number of projects common to the three platforms is constantly evolving (Fig 1). In terms of rate, new projects added to the three platforms have increased from about 3% in 2010 to about 25% in 2016. This number reaches 20% for the first seven months of 2017 and is expected to be around 35% by the end of the year. This result shows

that the three platforms attract interest and that it is possible to rely on their data when gathering data for providing decision support for OSS selection. Considering RQ3.2, Table 1 reports a summary of results for RQ2, reporting the average size, age and activity of the projects available in the three platforms.

The replication package, containing the raw data of this work is publicly available [14].

4. Discussion and conclusions

This work allowed us to identify the projects for which information is available on GitHub, SonarCloud, and StackExchange. Based on the aforementioned results, we can see that the trend towards projects being available on these platforms is growing every year. Moreover, the projects available on the three platforms are commonly stable and mature ones, so users can easily avoid new, immature ones.

The data available in the three platforms (Table 2) is very consistent, since it is automatically calculated by the platforms and do not strictly depends on how the developers use them. GitHub provides data independently from commit frequency (e.g. number of committers, number of forks, etc.). Only the license type needs to be specified by the developers. The same principle applies to SonarCloud, by default projects are analyzed every commit. However, in exceptional cases, developers could schedule the analysis less frequently.

The main contribution of the carried study is summarized on the support that may be provided to the user during the OSS adoption by: (1) providing a classification of information recommended by OSS selection models along with the one considered as relevant by users and (2) highlighting which information can be retrieved and from which platform. This work is therefore an important step forward towards the classification of the information available on the three platforms to determine OSS related characteristics. In the near future, we will implement a decision support system based on this information and we will start testing it with practitioners. The decision support system will dynamically retrieve the information and present them in a similar format as Table 2 from the platform APIs. Our decision system will allow users to clearly see all the information in one single platform and compare different projects based on the information. It will be developed to support the comparison of projects that are available at least in GitHub. The importance of the different data will be defined by the user that is selecting the OSS. However, in order to ease the process, we will initially set the importance based on the results provided in [4] and [5].

Moreover, we plan to work further on characteristics that are needed by users and / or recommended by models while they are not provided by the OSS platforms. For this point, we may consider studying other OSS platforms such as Launchpad, GitLab and Bitbucket or to retrieve the required characteristic from the OSS website when available.

Finally, existing research in software analytics and mining software repositories could be a useful support of this work, helping to use unexplored, incomplete or biased data retrieved from repositories.

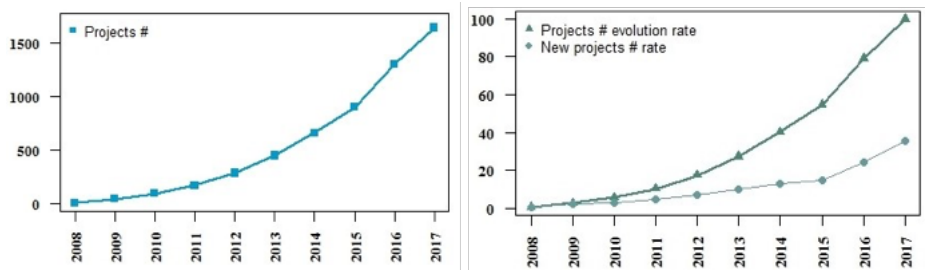

Figure 1: Evolution of number of projects per year

Table 1: Results for RQ 3.1 and RQ 3.2

# Commits per project	**1-1k**	**1k-10K**	**>10K**	**Average**
% projects	85%	13%	2%	1130
# Committers per project	**0-10**	**11-20**	**>20**	**Average**
% projects	80%	8%	12%	7
Project age	**<1 year**	**1-2 years**	**>2 years**	**Average**
% projects	32%	20%	48%	29 months
Project size (# LoC)	**1-10K**	**10K-100K**	**> 100K**	**Average**
% projects	80%	16%	4%	22446
Main Programming language	**Java**	**Javascript**	**C#**	**Others**
% projects	69%	11%	6%	14%

References

[1] A. Wasserman, M. Pal, C. Chan, The business readiness rating model: an evaluation framework for open source, in: EFOSS Workshop, Como, Italy, 2006.

[2] D. Taibi, L. Lavazza, S. Morasca, Openbqr: a framework for the assessment of oss, in: Open Source Software, IFIP International Conference on, Springer, Limerick, Ireland, 2007, pp. 173–186. doi:10.1007/978-0-387-72486-7_14.

[3] Method for qualification and selection of open source software (qsos) version 1.6 Atos Origin (april 2006), http://qsos.org/.

[4] V. D. Bianco, L. Lavazza, S. Morasca, D. Taibi, A survey on open source software trustworthiness, IEEE Software 28 (5) (2011) 67–75. doi:10. 1109/MS.2011.93.

[5] D. Taibi, An empirical investigation on the motivations for the adoption of open source software, in: Software Engineering Advances, the 10th International Conference on, IARIA, Barcelona - Spain, 2015.

[6] G. Basilico, L. Lavazza, S. Morasca, D. Taibi, D. Tosi, Op2a: Assessing the quality of the portal of open source software products, in: Int'l conf on Web Information Systems and Technologies. 2011.

[7] A. I. Wasserman, X. Guo, B. McMillian, K. Qian, M.-Y. Wei, Q. Xu, Osspal: Finding and evaluating open source software, in: OSS, Buenos Aires, Argentina, 2017.

[8] W. C. Duijnhouwer FW, Open Source Maturity Model., Capgemini Expert Letter, 2003.

[9] I. Samoladas, G. Gousios, D. Spinellis, I. Stamelos, The SQO-OSS quality model: Measurement based open source software evaluation, in: OSS 2008.

[10] M. Soto, M. Ciolkowski, The qualoss open source assessment model measuring the performance of open source communities. In: ESEM '09, 2009.

[11] P. Runeson, M. Host, A. Rainer, and B. Regnell, Case Study Research in Software Engineering: Guidelines and Examples. John Wiley & Sons, 2012

[12] K. Petersen, S. Vakkalanka, L. Kuzniarz. Guidelines for conducting systematic mapping studies in software engineering: An update. Information and Software Technology. vol. 64, pp. 1-18. (2015)

[13] H. Gall, H., Menzies, L. Williams, T. Zimmermann. Software Development Analytics (Dagstuhl Seminar 14261). Dagstuhl Reports. Vol. 4. No. 6. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2014.

[14] N. Sbai, V. Lenarduzzi, D. Taibi, S. Ben Sassi, H. Ben Ghezala "Exploring information from OSS repositories and platforms to support OSS selection decisions: Raw Data", Mendeley Data 2018, v1. http://dx.doi.org/10.17632/dbkm2xdct9.1

Table 2: Results for RQ1 and RQ2

	RQ1										RQ2
	OSS Selection Models							Surveys			Platforms
Information	M1: OSSPAL	M2: OpenBRR	M3: QSOS	M4: OpenBQR	M5: OSMM	M6: SQO_OSS	M7: Qual_OSS	# of models	S1: [4]	S2: [5]	GitHub (GH) SonarCloud (SC) StackExch. (SE)
Community and support	✓	✓	✓	✓	✓	✓	✓	7	✓	✓	
Number of contributors	✓	✓	✓		✓	✓	✓	6	✓	✓	GH
Number of subscribers	✓	✓	✓		✓		✓	5	✓	✓	GH
Availability of questions/answers		✓			✓	✓	✓	4		✓	SE
Number involved developer per company				✓				1	✓	✓	
Number of independent developer				✓	✓	✓		3			
Community size	✓	✓	✓		✓		✓	5	✓		GH, SE
Quality of professional support		✓	✓	✓	✓		✓	5	✓	✓	
Availability of training		✓	✓		✓			3	✓	✓	
Clear Project Management		✓	✓		✓		✓	4			
Economic				✓	✓			2	✓	✓	
Competitiveness					✓			1		✓	
Economic advantage				✓	✓			2	✓	✓	
License cost								0	✓	✓	
Documentation	✓	✓	✓	✓	✓	✓	✓	7	✓	✓	
Availability of documentation/books		✓	✓	✓		✓	✓	5	✓	✓	
Documentation comment lines								0			SC
Documentation comment density								0			SC
Availability of architectural documentation		✓						1	✓	✓	
License	✓	✓	✓	✓	✓		✓	6	✓	✓	
License type	✓	✓	✓	✓	✓		✓	6	✓	✓	GH
Law conformance								0	✓		
Maturity	✓	✓	✓	✓	✓	✓	✓	7		✓	
Number of forks	✓		✓					2			GH
Stability	✓		✓		✓	✓		4			
Number of releases		✓	✓	✓			✓	4			GH
Age			✓		✓			2			GH
Number of commits	✓							1			GH
Maturity fault detection								0			GH
Maturity fault removal								0			GH
Quality											
Reliability	✓	✓	✓	✓	✓	✓		6	✓	✓	SC
Number of Faults (open, closed...)	✓	✓	✓	✓				4	✓		GH
Average fault age		✓	✓	✓				3			GH
Reliability remediation effort								0			SC
Performances	✓	✓			✓	✓		4	✓		
Scalability	✓	✓						2			
Security	✓	✓			✓	✓	✓	5	✓	✓	SC
Number security vulnerabilities code		✓						1			SC
Information for security	✓	✓				✓	✓	4		✓	SC
Code Quality	✓	✓		✓		✓	✓	5	✓		
Code Complexity (class, methods)	✓			✓				2	✓		SC
Cognitive complexity								0			SC
Code Size (Lines of Code)	✓							1			SC
Number of Classes								0			SC
Number of Files								0			SC
Duplications								0			SC
Coding standard violations								0			SC
Test coverage		✓				✓	✓	3	✓		SC
Skipped unit tests								0			SC
Unit test failures/success								0			SC
Maintainability		✓	✓	✓		✓	✓	5	✓	✓	
Analyzability						✓		1			SC
Changeability						✓	✓	2	✓		SC
Testability		✓				✓	✓	3	✓		SC
Code Modifiability			✓				✓	2	✓	✓	
Code reusability		✓		✓				2	✓	✓	
Code smells								0			SC
Technical debt								0			SC
Modularity		✓	✓		✓			3	✓		
Usability	✓	✓			✓			3	✓		
Portability		✓			✓			2	✓		
Flexibility		✓	✓		✓			3		✓	
Adaptability			✓		✓			2		✓	
Other											
Independence from other SW				✓				1		✓	
Collaboration with other product					✓			1	✓		
Plugin support	✓							1			
Development Language	✓			✓	✓			3	✓		GH, SC
Multiplatform support	✓				✓			2		✓	
Compliance with standards		✓	✓	✓	✓			4	✓		
# Information considered	24	33	25	22	30	18	21		35	27	