<table>
<thead>
<tr>
<th><strong>Title</strong></th>
<th>Industry Questions about Open Source Software in Business: Research Directions and Potential Answers</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Author(s)</strong></td>
<td>Ihara, Akinori; Monden, Akito; Matsumoto, Ken-Ichi</td>
</tr>
<tr>
<td><strong>Citation</strong></td>
<td>2014 6th International Workshop on Empirical Software Engineering in Practice, 12-13 Nov. 2014, Osaka, Japan</td>
</tr>
<tr>
<td><strong>Issue Date</strong></td>
<td>2014</td>
</tr>
<tr>
<td><strong>Resource Version</strong></td>
<td>author</td>
</tr>
<tr>
<td><strong>Rights</strong></td>
<td>© 2014 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.</td>
</tr>
<tr>
<td><strong>DOI</strong></td>
<td>10.1109/IWESEP.2014.12</td>
</tr>
<tr>
<td><strong>URL</strong></td>
<td><a href="http://hdl.handle.net/10061/12741">http://hdl.handle.net/10061/12741</a></td>
</tr>
</tbody>
</table>
Industry Questions About Open Source Software in Business: Research Directions and Potential Answers

Akinori Ihara*, Akito Monden*, Ken-ichi Matsumoto*
*Graduate School of Information Science, Nara Institute of Science and Technology, JAPAN
Email: (akinori-i, akito-m, matumoto)@is.naist.jp

Abstract—As open source software (OSS) has become an integral part of today’s software businesses, many software companies rely on OSS to develop their customer solutions and products. On the other hand, they face various concerns in using OSS, such as technical support, quality, security and licensing issues. This paper focuses on OSS-related FAQ in industry, and tries to answer them or to provide research directions based on lessons learned from recent mining OSS repository researches.

I. INTRODUCTION

Today’s many software businesses rely on open source software (OSS) as it has now become an essential infrastructure in various computer environments. Typically, recent mobile phone companies are tackling Android operating system, which is now the most widely used smartphone platform in the world. Software/game companies and enthusiasts are also developing various applications for Android. As of July 2013, the Google play repository reached 1 million applications and 50 billion downloads [35].

While OSS enables companies to develop software systems at low cost, numbers of concerns are annoying the companies because using OSS is quite different from conventional software development. An industry survey of 916 Japanese software companies in 2009, conducted by the Information-technology Promotion Agency, Japan, has revealed numbers of questions (Table I) in using OSS [14], such as technical support, quality, security and licensing issues. These questions come from the nature of OSS — there are various major/minor versions and branches, with a huge number of security/bug patches available, while there is neither a person who is responsible of the quality of OSS, nor a customer service desk to get a technical support. This paper tries to answer industry questions based on lessons learned from mining software repositories (MSR) researches. Many OSS projects provide public software repositories — typically, source code repository, bug tracking repository and mailing list repository; and, this enables MSR researches to get useful information from the repositories for OSS developers and users.

II. OSS USER COMPANY SURVEY

This section introduces OSS user company survey conducted by Information-technology Promotion Agency, Japan [14]. Participants for the survey are 916 Japanese IT companies including software companies (556), IT service companies (121), Internet-related service companies (62) and unknown (177). 536 of them locate at provincial area, and 306 are at metropolitan area.

III. POTENTIAL SOLUTIONS FOR INDUSTRY QUESTIONS

Figure 1 shows experience in using OSS. Notably, 51.6% of the surveyed companies have experience in using OSS for their customer systems, and 66.8% including in-house systems. This indicates that many software companies rely on OSS to develop their customer solutions, and OSS are now an integral part of today’s software businesses.

Table I shows major questions or concerns when using OSS in industry. The column “% Companies” indicates the percentage of companies who have posed each question. From next Section, we will seek for potential solutions for these questions from past MSR studies.

A. Question 1: How can we get rapid technical help from an OSS community?

This question was posed by 67.3% of the surveyed companies [14]. Industry developers often face issues to be solved when they use OSS in their businesses. As there is no help desk for OSS in general, developers need to somehow get solutions from an OSS community.

In order to bridge the gap between developers and users, mailing lists have been used [11][25][27]. With growth of social Q&A sites such as the StackExchange network (e.g., StackOverflow), many people turn their interests to such sites. These sites are rapidly changing the way of collaborating between developers and users [2][4][24][33].

As a suggested answer or a research direction, recent MSR researches have focused on online software forums to get relevant solutions for a particular issue [9]. It is often the case that an issue faced by one developer or user has been faced by many others before. Therefore, one may find solutions from thread discussions in an online forum. To find a relevant solution quickly, Gottipati et. al. proposed a semantic search engine framework to process software threads and recover relevant answers according to user queries [9].
TABLE I: Six questions or concerns when using OSS in industry.

<table>
<thead>
<tr>
<th>Questions</th>
<th>% Companies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Question 1: How can we get rapid technical help from an OSS community?</td>
<td>67.3%</td>
</tr>
<tr>
<td>Question 2: How much longer does an OSS project sustain?</td>
<td>58.8%</td>
</tr>
<tr>
<td>Question 3: How can we fix bugs and/or add new functions to OSS?</td>
<td>43.4%</td>
</tr>
<tr>
<td>Question 4: How can we understand and identify OSS licenses?</td>
<td>34.8%</td>
</tr>
<tr>
<td>Question 5: Which product and which version should we use?</td>
<td>32.8%</td>
</tr>
<tr>
<td>Question 6: How can we assess the quality of OSS products and the maturity of OSS projects?</td>
<td>29.5%</td>
</tr>
</tbody>
</table>

B. Question 2: How much longer does an OSS project sustain?

This question was posed by 58.8% of the companies [14]. It is often the case that an OSS project suddenly disappears or no longer continues updating products. For example, a famous web browser Netscape was discontinued and all support was terminated on March 2008 due to losing market share. As another example, the GIMP project, which started as an academic project, has stopped, because the creators left the university for work, and mostly ended their relationship with the GIMP. After all, the project was almost stopped for more than a year until someone else took over its control. This indicates that the coordination activity by core-developers is important to keep attracting people [23][38].

As a suggested research direction, activity metrics of developer/user communities [36] could be used to assess the project sustainability. If activity became low, it can be a symptom of project decay. In addition, some MSR researches focus on communication healthiness in the developer and user communication network [17][26]. It turned out that communication between user and developer was not active enough in the end of Netscape project. On the other hand, successful community (such as Apache) had many coordinators, who act as a bridge between user and developer communities [17]. These results suggest that activities and communications in OSS user/developer communities should be measured to assess the project healthiness.

Another direction is to analyze OSS projects from two point of views “stickiness” and “magnetism.” Yamashita et al. [37] found that some projects attract new developers (magnet), and some retain existing contributors (sticky). For example, the Homebrew is one of the successful projects having high stickiness and magnetism.

C. Question 3: How can we fix bugs and/or add new functions to OSS?

This question was posed by 43.4% of the companies. In case an industry developer finds a bug (failure) or wants to add a new functionality to an OSS product, debugging and adding functionality in source code is usually very difficult because the code has been developed by someone else in the OSS community. Therefore, a technique to assist enhancing OSS code is greatly demanded.

To answer this question, several techniques from MSR researches can be used. One technique is called bug localization, which identifies a particular location in source code where a fault is likely to be existing, using as input detailed description of failure occurrence (such as actions to reproduce the failure, the name of a function where the failure has occurred, and so on). To this end, several studies propose the use of Information Retrieval (IR) based classifiers to locate bugs [16][21][32][39]. For example, text mining techniques are used to manage keywords in the failure description and source code to identify fault location. Recently, various improvements to bug localization have been proposed, and now it has become a hot topic in MSR [22][34].

Another technique is called a co-change analysis or a logical coupling analysis, which identifies a set of source files that needs to be changed together using as input past source code commit logs [1]. This technique helps developers to fix a bug and/or adding functionality to source code. Co-change histories are also used in recent bug localization studies [31].

Feature location is also a useful technique for OSS users when they want to identify an initial location in the source code that corresponds to a specific functionality (feature) to be enhanced or fixed [7]. A user can input a natural language query, execution trace or source code artifact, to obtain a ranked or visualized output of source code fragments such as files, classes, methods, functions or statements.

D. Question 4: How can we understand and identify OSS licenses?

This question was posed by 34.8% of the companies. There is a wide range of variations in Open Source licenses; and, many users do not understand them correctly. As a reference model of OSS licenses, the Open Source Initiative (OSI) provides the Open Source Definition (OSD) as follows. The OSI accepts about 70 OSS licenses and provides a categorized list of licenses. This list may help OSS users to understand the variety of licenses.

Another industry concern with OSS license is that a single OSS product often contains multiple licenses; and, thus, identifying all licenses in the product is often very difficult. One solution to this concern is use of an OSS license matching system. German et al. [8] proposed a text mining approach to automatically identify all licenses in a given set of source files.

Inspection of industry software products for possible OSS license violations is also becoming increasingly important as more reusable OSS code becomes available online [13][20]. Currently, several services are available for OSS code detection and management such as Black Duck Software’s Protx (www.blackduckssoftware.com) and Palamida (www.palamida.com). Furthermore, the “provenance” of source

1The Open Source Initiative, http://opensoource.org
code [6] is also an important issue to understand where each program fragment come from and where it goes.

E. Question 5: Which product and which version should we use?

This question was posed by 32.8% of the companies. There are a huge number of OSS products in the world, and there may be too many similar OSS products in each application domain. Therefore, it is not easy to find the most relevant OSS product for a particular business objective. In addition, many OSS products have multiple versions where reliability, functionality and compatibility are different. The latest version is often not a good choice since it may be not stable enough. Finding appropriate version is also a difficult problem for industry developers.

One answer to find a relevant product is to use software search engines, such as SPARS 2, Koders 3, Jarhoo 4. These search engines are especially useful for searching for a particular software component. Another answer is to use an automatic software categorization system, which helps users to find similar OSS products [15].

Regarding finding appropriate version, we could learn from “the wisdom of the crowd.” Mileva et al. [19] analyzed the frequency of use of 450 different library versions and developed a library recommender system called AKTARI 5. They found that developers often downgrade to previous library because of bugs found in new version. Generally, if a specific version is used by many developers, we could consider it is of high quality.

Similarly, Sunada also followed the idea of “the wisdom of crowds,” to clarify frequently-used libraries in Java open source software development [30]. He analyzed frequently-used libraries in many domains, and clarified domain-specific libraries, as well as domain-independent libraries such as basic function libraries (e.g. commons-collections, commons-lang) and logging libraries (e.g. commons-logging and log4j). Figure 2 shows the result for Enterprize applications in Business/Office domain [30].

F. Question 6: How can we assess the quality of OSS products and the maturity of OSS projects?

This question was posed by 29.5% of the companies. When a company considers using an OSS product for their business, the company needs to assess the quality of the product as well as the maturity of the project. Also, the company often needs to explain the quality of the OSS product to their customers. To answer this question, we introduce QualiPSo (Quality Platform for OSS) project. In 2006, QualiPSo was launched to establish the reputation of reliability and quality for OSS by EC (European Communities). 21 organizations in 11 countries collaborated to investigate 7 fields (Quality/ Interoperability/ Factory/ Organization/ Life Cycle Matunity/ Business/ Legal) of OSS. MOSST (Model for OSS trust worthiness) and OMM (Open Source Maturity Model) were developed based on the result of MSR techniques and interview to industry developers who use OSS.

The MOSST is a customizable model for estimating the trust that OSS developers and end-users can have in the qualities of OSS products. One of the functions is to visualize the reliability of OSS. They assessed the reliability of the OSS based on over 90 metrics measured from source codes and binary code of OSS. Then it shows it in 3 levels (good, acceptable, poor). The MOSST provides advices to improve the reliability of OSS for OSS developers. Also, it helps the reputation of reliability and quality for OSS users. They have already applied in over 100 OSS projects to evaluate the MOSST as 14 March, 2012.

The OMM is a CMMI-like process model for OSS development. This process model aims to help in building trust in development processes of companies using or producing OSS. The OMM comprises trustworthy elements required for OSS development based on surveys and best practices from CMMI. The trustworthy elements are grouped into 3 maturities levels (basic, intermediate and advanced). Now, they are trying to establish the standard reliability evaluation index for OSS.
IV. DISCUSSION

Although numbers of solutions have been proposed for OSS user companies, another discussion point would be how we can help industry to adopt MSR techniques, and how we can lower the barrier of entry. In this point of view, development of tools and/or services that implements MSR techniques is required. One of such tools is Ninka developed by Manabe and German, which implements their license identification method proposed in [8]. Another direction would be to provide tools, services or datasets to lower the barrier of repository mining [10][18]. Current such services focus on researchers to help, so further services for industry are demanded.

On the other hand, recent companies often participate and contribute to OSS projects to understand, enhance and/or fix bugs of OSS products so that they can use the products more easily for their businesses, while there are also barriers for such newcomers. Steinmacher et al. [29] explored the difficulties of participating in OSS projects. They identified 38 barriers of 7 different categories, which may motivate new studies and tools to better supporting newcomers. Dagnas et al. [5] focus on the OSS documentation issue, which is one of the barriers. Canfora et al. [3] and Steinmacher et al. [28] focus on “mentors” that trains newcomers to understand the OSS projects to lower the barriers.

Finally, we believe understanding of software ecosystems, which include OSS developers and users as well as industry and academia, is very important. Recently, Information and Software Technology (journal) has published a special issue on software ecosystems [12]. Further studies are needed to promote win-win relationship among the stakeholders.

V. CONCLUSION

This paper introduces commonly-asked industry questions about OSS for a business use, and tries to answer them or to provide research directions based on lessons learned from recent MSR researches. We hope that industry developers find their own answers in their context based on suggested answers in the paper. We also hope, in future, more powerful solutions will come up from further MSR researches to assist OSS users in various application domains.

ACKNOWLEDGMENT

This work has been conducted as a part of “Research Initiative on Advanced Software Engineering in 2013” supported by Software Reliability Enhancement Center (SEC), Information Technology Promotion Agency Japan (IPA). Also, part of this research was conducted under Japan Society for the Promotion of Science, Grant-in-Aid for Young Scientists (B) (25730045), and Scientific Research (C) (22500028).

REFERENCES
