OSS Integration Issues and Community Support: An Integrator Perspective

Claudia Ayala, Daniela Cruzes, Anh Nguyen, Reidar Conradi, Xavier Franch, Martin Höst, Muhammad Babar

OSS Integration Issues and Community Support: An Integrator Perspective

Claudia Ayala[1], Daniela S. Cruzes[2], Anh Duc Nguyen[2], Reidar Conradi[2]
Xavier Franch[1], Martin Höst[3], Muhammad Ali Babar[4]
1 Universitat Politècnica de Catalunya
{cayala, franch}@essi.upc.edu
2 Norwegian University of Science and Technology
{dcruzes, anhn, conradi }@idi.ntnu.no
3 Lund University, Sweden, martin.host@cs.lth.se
4 IT University of Copenhagen, Copenhagen, Denmark, malibaba@itu.dk

Abstract. The reuse and integration of Open Source Software (OSS) components provided by OSS communities is becoming an economical and strategic need for today's organizations. The integration of OSS components provides many benefits, but also risks and challenges. One of the most important risks is the lack of effective and timely OSS community support for dealing with possible integration problems. For gaining an understanding of the common problems that organizations face when integrating OSS components, and the role played by OSS communities, we performed an exploratory study on 25 OSS integration projects from different European organizations. The results show that the main way of reducing integration problems was the use of OSS components from well-established communities; therefore very few integration problems were identified. In most of the cases these problems were successfully solved with the support from the OSS community and/or colleagues. In addition, contrary to the common belief that understanding code from someone else is a hard and undesirable task, some integrators consider OSS code even more understandable than their own code.

1 Introduction

The free availability of Open Source Software (OSS) has over the last decade had a significant impact, not only on the software IT industry, but also on software-intensive organizations. OSS is significantly influencing the ways these organizations develop, acquire, use, and commercialize software [1], and actual evidence shows that organizations are clearly becoming a very important part of the OSS communities

In particular, the integration of OSS components is one of the most popular ways of adopting OSS [2]. It involves including OSS components into other software products or systems and this again may involve modifying, extending, or wrapping the OSS components.

OSS integration might have many benefits, such as significantly lower (purchasing) costs, availability of high quality products, adherence to open standards

and vendor independence [1]. However, it also implies several challenges. On the one hand, we may mention that OSS components do not always satisfy all the requirements. In certain cases, some "glue code" or modifications are required to make OSS components work together. This however creates a customized version of the OSS component. The integrator (i.e., the person(s) in charge of integrating OSS component(s) into the software system) is then faced with the issue of maintaining this derived version, and must decide how to handle these extensions and modifications. As a result, each organization that modifies OSS components and incorporates them in its own applications is faced with the issue of whether to contribute or not to the OSS community [3], [4]. On the other hand, some studies emphasize that high-quality OSS components rely heavily on having a large, sustainable community to develop code rapidly, debug code effectively, and build new features [5]. Thus, the organizations that integrate OSS components into their systems represent a potential base of contributing members needed to sustain the OSS communities [6].

It is therefore vital to provide evidence that help OSS communities to envisage strategies to improve potential integration issues; as well as organizations to meet some practical challenges related to OSS integration. In this context, the goal for this study is therefore gaining an understanding of the common problems that organizations face when integrating OSS components, and the role that the OSS communities play in such integration processes. Thus, we conducted an empirical study on European organizations from Norway, Spain, Sweden and Denmark. It consisted on semi-structured interviews with 25 integrators from different organizations that represented 25 different integration projects. Based on their answers, we were able to draw some observations. We report our main findings in this paper.

The rest of the paper is organized as follows: Section 2 provides an overview of existing evidence on OSS integration and companies' participation. Section 3 provides details of the empirical study. Section 4 summarizes the most relevant observations from the interviews. Section 5 discusses the results. Threats to validity are presented in Section 6, while Section 7 summarizes the conclusions and future work.

2 Background and Related Work

Recent systematic reviews reveal that integration is one of the most popular strategies of adopting OSS [2], [4].

The company-community relationships have been explored in works as [7], [8], [9]. In [7], the authors identify three types of organization-community relationships:

- Symbiotic: Both the community and the organization benefit from the relationship.
- Commensalistic: The organization benefits from the relationship but the community is not affected.

- Parasitic: The organization benefits from the relationship but at the same time it damages the community.

Several barriers to contribute back to the community have been also investigated as for instance by Ven and Mannaert [10] that found that deciding not to contribute can also be risky as one may be forced to maintain a parallel copy of the product.

Furthermore, Stol and Ali Babar in [4] did a systematic synthesis of the reported challenges of integrating OSS and ended up with a comprehensive list of challenges related to OSS integration. Even though there is a considerable body of research on the challenges of integrating OSS components in the development of software products [2], [4], the majority of these works refer to success stories derived from single case studies or experience reports that provide very limited information about the real industrial landscape of companies integrating OSS components. Moreover, the role that OSS communities play on supporting integrators to solve their integration problems has not been further explored.

Therefore, our overall objective is gaining an understanding of the common problems that organizations face when integrating OSS components, and the role that the OSS communities play in such integration processes.

3 Survey on OSS Integration Issues and Community Support

Our overall objective has been broken down into two research questions that are at their turn broken down into more concrete sub research questions.

On the one hand, RQ1 was aimed to inquiry on potential integration issues. RQ1.1 is focused on inquiring the most common integration problems. Furthermore, the literature has pointed out the underestimation of integration effort and inefficient debugging as problematic areas that require further investigation [20]; therefore, we stated RQ1.2 and RQ1.3 respectively. Finally, as a previous study [13] reported that getting OSS components information seems to become a continuous monitoring activity rather than being on a project demand basis, we stated RQ1.4 to understand how integrators monitor OSS communities.

On the other hand, as pointed out in the previous section, company-community relationships have been reported before (e.g., [7-9]), however, there are no sufficiently deep studies to further understand what kind of assistance and/or contributions are mostly requested/provided by integrators, and which means are used to do so. Therefore, RQ2.1, RQ2.2, and RQ2.3 were stated.

RQ1: How do integrators deal with integration issues?

RQ1.1-What are the most relevant integration problems?
RQ1.2-How are integration/testing costs estimated?
RQ1.3-What are the differences on locating/fixing bespoke software bugs vs. OSS related bugs?
RQ1.4-How are OSS communities being followed up?

RQ2: To what extent integrators interact with/contribute to the community?

RQ2.1-What kind of help do integrators request from the OSS community?

RQ2.2-What kind of contributions do integrators provide to the OSS community?

RQ2.3-Which means are used to interact with the OSS community?

3.1 Research Method

Interviews, observation and analysis of documents are some of the most common data collection methods. However, a stated in [12], qualitative (approached by interviews) and quantitative (approached by questionnaires) surveys are the two most relevant types of studies for component-based software engineering investigation. Thus, as the nature of our research questions was clearly exploratory, we decided to carry out the study using a qualitative research approach based on semi-structured interviews to collect data directly from software-intensive organizations that integrate OSS in software product development. Semi-structured interviews allowed us to have certain flexibility to further explore what was going on in the area.

Participants. Participating organizations were chosen from our direct or indirect industrial collaboration network. They include organizations with different sizes and in different application domains. 69 organizations were invited to participate by phone call and email. Some of the contacts were not eligible for participating due to several reasons, such as lack of integration of OSS components in the projects, or privacy of the OSS adoption strategy. We ended up with 25 integrators from different organizations that represented 25 different projects. Table 1 shows some details of the organizations and the analyzed projects.

The Instrument. The interview guide was carefully designed following the guidelines stated in [11] and previous experience performing international surveys from several members of the team [12], [13], [14, [15]. The survey was designed as a 5-section survey, with both closed and open questions. The closed questions were used to solicit information about the respondent and project context. The open questions were used to gather information on integration issues and community relationship. The survey also included an introductory section concerning relevant terminology and background in order to offer a common understanding to all participants. In this paper we report our finding related to the relationship among integrators and communities (other results from the study have been also reported in [16]). In general, the guide mostly focused on a single software development project with at least one release of the corresponding software product, and with integration of one or more OSS components. If the respondents had experience with several such projects, they were asked to choose the most familiar one.

Data Collection Procedure: The interview guide was sent to all participants some days before the interview meeting. In this way, they could be prepared for the interview. The participants were asked to fill in the first two parts of the survey and

give back to us beforehand. The next three parts of the survey were asked directly to the participant during the interview. Interviews were mainly performed in the mother tongue of the respondents (some exceptions occurred in Norway, where the interviews were performed in English) and when possible face-to-face in their working place or by phone, by one to three researchers of the team. Interviews lasted around 40 to 75 minutes each and were recorded for subsequent analysis.

Table 1. Some details of the organizations and projects studied

Id	# Employees	Application Domain	Project Staff	Staff with experience in OSS integration	Some OSS used	% OSS of the system	Total effort (person/ months)
A	170	Defense (communications)	20-25	30%	JBPM, Jetty, Spring, LogBack, Maven	90%	>2000
B	1	ICT Industry	4	50%	Impact, LPng	10%	480
C	3	ICT Industry	2	100%	SolR, Xapian, Twisted: NLTK.	80%	12
D	350	Embedded systems	18	25%	Linux Kernel, MD5 Checksum	-	-
E	500	Oil and gas industry	2	50%	PDfLib, OpenPyExcel	77%	18
F	-	Public sector	200	60%	Flex Framework, Batch part of Spring	75%	-
G	230	Bank	4	100%	WideShot, CryptoPP, ParseXs	10%	36
H	190	Public sector	20	100%	JBoss, OpenSummer, USD	66%	1000
I	6	Finance	1.5	66%	Python, Soap and Django	90%	3
J	4	Public sector (Education)	3	100%	SunGridEngine, Cluster FS, Linux Debian, Ganglia	90%	30
K	100	Private services (entertaiment, sales)	3	100%	Apache, MySQL, PHP, FFTP tools	5%	7.5
L		Public sector	5	100%	Mantis, Ant, Apache	80-90%	72
M	150	Public sector (Education)	6	100%	Jasper Reports, DOJO, Apache, Quark	25%	157
N	30	ICT	7	14%	Jenkins, Cucumber, Mercurial	10%	84
O	15	ICT	3	100%	Joomla	50%	56
P	5	Public sector	2.5	67%	Zope and Plone	99%	6
Q	14	ICT	3	100%	Varnish, Engine egg	80%	9
R	500	ICT	25	80%	Jasper Reports, Junit, Jmeter, MediaWiki, OpenCSV	30%	900
S	2	Public sector	2	100%	RXTX, MySQL, Palcom	60%	36
T	>1000	ICT	250	50%	OSS platform	50%	1000
U	11	Energy	2	100%	Speed -Typo3CMS, FPDF, Apache, Stability	40%	20
V	2500	ICT	4	100%	Mongo DB	100%	8
W	4	Whole-sale, retail and entertainment	10	50%	Apache, MySQL, PHP Suite,	100%	24
X	1	Public sector (Education)	1	100%	Sbuntu Enterpise Cloud (UEC) & Eucalyptus, NappIt, pfSense, FreeBSD	100%	6
Y	7	Medical	1	0	Zope, Plone, Apache, Mysql, Ubuntu	100%	3

(-) respondent did not answer or asked to keep this information confidential.

Data Analysis: Interviews were prepared for analysis by the manual transcription of audio records to text documents (the transcripts vary from 13 to 21 pages in size). When needed, a summary of each interview was translated to English so that the whole research team could assess and discuss the data. We analyzed the filled-in questions and transcripts using a qualitative approach that consisted on the assessment of the interview documents by two different researchers and the subsequent generation of categories by grouping sentences or phrases that described the same idea, action or property [11]. We tried to be exhaustive with the categories in order to include as much detail provided by the respondents as possible.

4 Results

This section presents the results of the study. They are grouped in 2 subsections according to the research questions introduced above, when possible, we use tables to illustrate the resulting categories.

4.1 RQ1: How Do Integrators Deal with Integration Issues?

RQ1.1- What Are the Most Relevant Integration Problems?

Twenty out of 25 respondents did not mention any relevant integration problem in the project they based their answers on. Some of them commented: *"We use components that are like standards and with a big community behind, so it is hard that you are the first one that experiences a problem"* (K); *"In this case, we were lucky. The documentation was complete and updated"* (P).

Only five respondents mentioned that they experienced some kind of integration problem. Two of them said that they dropped and changed the OSS component to solve the problem. One emphasizes: *"[The potential problems] depend on getting the right component"* (X). Two respondents agreed that the problem was solved by learning how other people proceeded in similar cases: *"It was a problem related to incompatibility among versions. But, we solve it by searching in Google and finding people that have explained their solution for it"* (O); *"Yes, we had some problems, but they were already reported by someone else in the forum, so we just learn some tricks to solve it"* (R). One respondent stated that the problem came from the lack of documentation *"We struggle a little with data formats, because sometimes the documentation was incomplete"* (V).

RQ1.2 How Are Integration/Testing Costs Estimated?

On the question about integration/testing costs, sixteen out of 25 respondents agreed that integration costs were estimated based on the experience of the development

team. One respondent said: *"There is a kind of guessing in this. We ask the development team and with their experience they come with numbers and we put a bill on it"* (U). In addition, there were some mixed views on how costly the OSS integration was. One respondent for example thought the cost was low: "*It is difficult to say, but in any cases it would be less than developing the component yourself. For the small component, the integration cost is very low anyway because they have a nicer interface…"* (G). But another respondent said: *"There is normally a lot of costs involved with testing and integration. Lots of money is involved from exchanging part to integrating part. Integration sometimes involves competition with closed systems or exchange with other systems"* (Y).

Three respondents pointed out that they used piloting as a way to estimate costs of integration. In these cases the pilot took from one to two months. Two respondents answered that the estimation was part of the preliminary study of the candidate components. In two interviews, respondents said that their organizations had a marketing department responsible for the estimation costs, so the respondents did not know details about such estimation. One respondent stated that their estimation was driven by a testing tool *"We used a testing tool. Integration and testing was around 20% of the whole development"* (Q). Finally, another integrator stated that they used specific templates for the estimation (T). Table 2 summarizes the obtained categories.

Table 2. Categories of Integration Costs Estimation

Count	Categories
16	Experience-Based
3	Did a pilot
2	In-house marketing department
2	Preliminary study of the candidate components and their integration problems
1	Testing tools
1	Templates

RQ1.3 What Are the Differences on Locating/Fixing Bespoke Software Bugs vs. OSS Related Bugs?

We inquired about the differences among bespoke vs. OSS bugs' locating/fixing process. Nine respondents stated that they do not try to locate bugs in the OSS components. One of them commented: *"the components we used are like standards. Everything has been proven several times and it is well documented, so we did not find bugs"* (K). Nine respondents emphasize that there was no difference on how they located/fixed the bugs. At this respect, one respondent said: "*in my experience, most open source libraries and components are well written and the author usually put pride in putting out something that is well commented and nice formatting, and usually it is quite easy to navigate around so; actually, the process is a bit similar"* (E). On the other hand, two respondents said that the main difference resides on the fact that it is harder to look at someone else's code. "*We run code. If it does not*

work, we isolate the faulty areas. Then we get to know whether it is in own code or OSS code. It is usually in our own code. It rarely happens that OSS component has errors and they are cumbersome to resolve as we don't know that code"(U). One striking answer was on one respondent stating *"It is harder to find bugs in our own code. In the OSS components we didn't have the same amount of bugs than those bugs from us, because they were pretty much stable components. We didn't have to do any formal testing in these OSS components*" (F). One respondent stated that an external company was subcontracted to fix those bugs related to the OSS component that were not trivial *"When there is a problem that is trivial or small, we try to fix it by our self. When the problem is something different from standard Linux libraries, we have a company to fix. It is a consultant that deals with third party libraries, mismatches…"* (D). Finally, 3 respondents did not answer to this question. Table 3 summarizes the resulting categories.

Table 3. Differences Among Bespoke Software Bugs vs. OSS Bugs

Count	Categories
9	Do not try to locate OSS bugs
9	No difference with locating bespoke software bugs
3	No answer
2	It is harder to look at someone else code
1	Subcontract a company to fix OSS components bugs.
1	It is harder to find bugs in own code

RQ1.4 How Are OSS Communities Being Followed Up?

Fifteen respondents answered that they did not have someone following up with the OSS project. Some of their comments are: *"No, only if there is a problem we go to the community"* (J); *"We don't have anyone watching the update stuff…We don't usually update the OSS component. For instance, now we chose the JBPM version 4.4. We wait sometime until someone realizes that there is a new version, but we don't watch the community"* (A).

Eight respondents stated that there was a responsible for OSS component issues. In seven of these eight cases, such a person was a colleague in the organization. One of them commented *"Yes, there is a community coordinator who is the one that is the face of a community, and hence he/she follows the trends in this community.*" One respondent stated that instead of having a dedicated person inside the organization, they subcontracted a company to select the OSS components and support them in any integration issue (D).

Finally, two respondents did not answer this question.

4.2 RQ2: To What Extent Integrators Interact With/Contribute to the OSS Community?

RQ2.1 What Kind of Help Do Integrators Request From the OSS Community?

The analysis of the interviewees' responses regarding the support from the community shows that thirteen respondents did not explicitly request help from the OSS community. Instead, they just used what it was already available on the community portal or managed to solve doubts by consulting their colleagues or using Google. *"We did not make any contact extending the normal use of community forums and discussion boards. Most of our issues could be handled by information already available in the community portal"* (F).

Ten respondents stated that for some specific aspects, they requested community help and were satisfied with the obtained support: *"In a couple of technical aspects, we asked for opinions about what it was the better way to proceed"* (P); *"[There is] usually a very quick response"* (E).

One respondent stated that they started requesting help and became involved in the community so now they are active co-providers: *"We were the ones that uploaded this part of the OSS, so we were the ones that better knew such part"* (Q). Finally, one respondent stated that asked for help but did not get it (X).

RQ2.2 What Kind of Contributions Do integrators Provide to the Community?

We asked the respondents what kind of contributions they provided to the community. We consolidated their answers as shown in Table 4.

On the one hand, twelve respondents stated that they had reported bugs, but only nine of them eventually contributed by fixing them. Some of their motivations were: *"bug fixing is something we would sent back definitely because we are very interested to give it into the main branch so we don't have to fix it every time we do an update"*(H); *"It is so much easier to get the bug fixed if you submit the fix of course. And with the open source project you can do that*"(C).

On the other hand, twelve respondents stated that they mostly take advantage of the community without contributing: *"we have not done anything. We just used the components"* (N); *"We do not dedicate a budget to OSS bugs notification nor contribution activities"* (L). In addition, 4 respondents stated that they became co-providers of the community by contributing back some OSS components.

Finally, five respondents emphasize that they participate in organizations or activities to promote the OSS culture as for instance *"We are founding members of Open source foundation"* (U), or *"We presented our resulting system in Workshops and Seminars to show how integrating OSS components can work"* (J).

Table 4. Results of integrators' contribution to the OSS community

Answer	Own bug reports	Bug fixes with code	Become co-providers	Promoting the OSS culture
YES	12	9	4	5
NO	12	15	19	9
Unknown/no answer	1	1	1	10

RQ2.3 Which Means Are Used to Interact With the OSS Community?

Nineteen respondents mentioned that they use to different extent bulletin boards, forums, email lists and the bug tracking system from the community project. Forums and bulletin boards were mentioned the most. However, there were six extreme cases were the respondents did not need any kind of direct interaction with the community: *"No cooperation with community. We just downloaded the software"* (V). *"We did not need to communicate with the community as the components we used were very well documented"* (K);*"We don't need direct contact with open source projects. We use the product because we have so much competence, either in the team or friend-to-friend. So, we don't need to communicate with the community directly"* (A); *"We mostly read the documentation and things published in the OSS community, but did not collaborate directly. Furthermore, in cases when problems appeared, we used Google to find related hits or portals like StackOverFlow"* (M).

5 Discussion of Main Findings

In this section we discuss the obtained results and establish whenever possible links to the findings of previous studies.

For most of the analyzed projects, integrators did not mention any relevant problem. Although this was an interesting observation, (as integration problems have been highlighted as one of the main concerns of organizations that integrate third-party components [20]), it is important to understand these results in the context of the analyzed projects. In fact, in the analyzed projects, integrators tried to minimize potential integration problems by selecting OSS components that fulfilled an adequate level of documentation/information and/or ensuring that they would have enough (own or subcontracted) expertise to solve the potential problems. Thus, it can be observed from Table 1 that the OSS components used by most of the respondents refer to OSS projects with great activity and vitality. In addition, some of these OSS components have become de facto standards.

It is worth to mention that although some works have claimed that much of the literature does not reflect the huge diversity in OSS initiatives and projects, focusing instead on large, well-established communities. In our case, even if we did not have control over the projects selected by the organizations, we ended up mostly analyzing projects that integrated OSS components from well-established

communities as organizations actually use these kinds of components. Thus, we agreed with Choi et al [18] that demonstrated that the mature status of well-known OSS projects likely attracts users given their greater activity and vitality. However this pathway is unavailable for most of the OSS projects and those newly initiated projects that struggle to attract users and contributors [19]. This also confirms the importance of studies that help OSS communities -especially those newly initiated projects that need to attract users- to envisage strategies for attracting integrators.

Regarding the way bugs were processed, we found, on the one hand, that nine out of 25 respondents do not even try to locate bugs on OSS components; instead, they rely on the expected functionality. On the other hand, other nine respondents emphasized that it was not difference on the way they fixed bugs in their own code instead of fixing bugs from OSS components, mainly because the OSS code was understandable and well commented. In addition, one also said that OSS code is even more understandable than their own code. Most of them also claimed that finding bugs in OSS was not usual. In addition, it was interesting to see that 8 organizations have a responsible of the community trends. This seems to show the importance that the OSS communities are gaining in the organizations.

Regarding costs estimation, we found that most integrators did not further estimate integration costs; instead, they just made an informal approach based on their experience. So, it seems that the claim from Li et al [20] about the relevance of estimating the time that the component(s) integration takes, do not hold in most of our analyzed projects.

Furthermore, in most of the analyzed projects, integrators managed to deal with their integration problems by themselves, without requesting specific help to the community. They mostly used information/documentation already available in the community portal or asked their colleagues for help. In line with this observation, our results also show that forums and bulleting boards from OSS communities were typically used in a passive way (i.e., integrators navigated through documentation and previous posts more than actively participate by adding new posts or content).

It is worth to highlight that the perception of the integrators about the support received from the community was good. 24 out of 25 said that they managed to solve the potential integration problems by using the information available in the portal or requesting help to the community with usually a quick response. Only one case stated that he/she did not receive the expected help.

Regarding the integrator's contribution, our results show that most integrators had limited interaction/contribution to the communities. This confirms the observations from [21-24] that emphasize that most organizations seem to have rather limited contributions to the OSS communities. Furthermore, although our results show that the most frequent way to contribute was by providing bug reports without code, the number of integrators that also submitted the code for fixing the bug was also high. This seems to confirm the claim from [25] and [26] regarding that the number of organizations contributing to OSS seems to be increasing. In addition, other ways of contributing that have been usually overlooked by previous research are related to

activities to promote the OSS culture by for instance funding OSS initiatives or sharing the knowledge with colleagues.

Regarding the involvement of the approached organizations in terms of the company-community relationships described by Dahlander and Magnusson [7], [8] (see section 2), our results show that almost all studied organizations seemed to have a commensalistic relationship with the community (i.e., the organization just benefits from the community). It was interesting to see that 4 out of 25 organizations have become active members of the community as co-providers of some specific parts of the OSS project, thus establishing a symbiotic relationship.

Furthermore, a common motivation for those that contributed to the community seemed to be to make sure that modifications to the component's code were maintained, while a common inhibitor to contribute in those organizations that did not contribute was that their budget did not include time neither resources to participate in the communities. These factors have been also mentioned by Ven and Mannaert [10]. In addition, most integrators that did not contribute to the communities also mentioned that they try to use the component as is (i.e., without modifications). This agrees with the results stated by Li et al. [20] that showed evidence that the source code of OSS components is seldom modified, or Höst et al. [27] that in a focus group meeting found that practitioners based on their experience do not recommend adapting OSS components that are included in products. However, if they need to adapt them, the recommendation is to do this through "glue code".

6 Limitations of the Study

This study was performed by means of a rigorous planning and the establishment of protocols for data collection and data analysis. This was especially important as the research involved several researchers and participants from different countries. In addition, the interview guide was carefully designed and piloted to improve its understandability. As a result, some changes in the interviews were done to enhance the elicitation process. Some vocabulary was defined at the beginning of the interview guide to homogenize concepts.

Some relevant decisions were taken for approaching a further understanding of the project contexts. One of these was to focus most of the questions of the interview guide on a single product development project so we could further inquire and analyze specific contexts of the projects. This enhanced the value of our analysis and observations. In addition, we sent the interview guide in advance to the respondents so that they could be informed of the kind of questions to be asked. As a result, when performing the study, we rarely experienced respondents having difficulty remembering project details. Furthermore, we explained to the respondents that our study was not focused on analyzing "wrong practices" but on knowing "how integration is done in industrial practice". In several cases we experienced that the interviewer(s) shall skip some questions given time restrictions of the respondent;

therefore, some questions results did not cover all participants. Despite this, the results obtained for these questions were valuable as most of the respondents provide their answers. With respect to the data analysis strategy, recording all interviews (and later on transcribing them) contributed to a better understanding and assessment of the data gathered. The generated categories were analyzed, discussed and reviewed by all researchers of the team to ensure their accuracy, understanding and agreement.

Regarding external validity, we addressed several topics in our study. Some of the most relevant ones are listed. First, the companies in this study were selected by a strategy combining convenience and maximum variation sampling from 4 different countries (Spain, Norway, Denmark and Sweden). Second, we had no control over the projects chosen by the respondents. Nevertheless, most of the resulting projects from the participating companies did not cover domains such as real time or life critical requirements neither development for product lines. We are aware that these factors may have an impact on integration, and so we highlight that our findings should not be taken as assertions but also as potential hypotheses that need to be further validated. Thus, we emphasize that our results should not be generalized and might be interpreted with caution, keeping in mind the context from the participating organizations.

7 Conclusions

We have described the main findings from an exploratory study based on semi-structured interviews to integrators from organizations that integrate OSS components in their software products. The study aimed to explore the problems that organizations face when integrating OSS components, and the role that the OSS communities play in such integration processes.

The reported results might be valuable for researchers, organizations and OSS communities that may use the provided evidence to more clearly understand the real OSS integration problems that integrators face and properly align their efforts for facing them.

On the one hand, researchers may get an overview of the state of the practice, identify new research questions, and position and align their own work. On the other hand, organizations may use the provided evidence to understand how other companies integrate OSS and leverage their own integration strategy identifying the practical challenges they might face when doing so. Finally, OSS communities can be informed of the perception of integrators regarding support and to envisage improvements for fostering the collaboration of integrators with the community; this is especially useful for newly initiated OSS communities that usually struggle to attract contributors.

That is, researchers might need to establish new agendas or check potential hypothesis generated by our results. Practitioners might have to adjust processes or

methodologies. And OSS communities might have to crate special integration groups or improve integration documentation.

Acknowledgments

We thank all people that participated in piloting an early version of the interview guide and the interview participants who took time from their workdays to participate in our interviews.

This work has been partly supported by the Spanish project TIN2010-19130-C02-01, and the Industrial Excellence Center EASE - Embedded Applications Software Engineering, (http://ease.cs.lth.se).

Jyoti Nandrajog contributed to the interview part of this study in Denmark.

References

1. Ayala, C.P., Cruzes, D., Hauge, Ø.; Conradi, R. Five Facts on the Adoption of Open Source Software. IEEE Software. March-April 2011, pp. 95-99.
2. Hauge, Ø., Ayala, C.P., Conradi, R.Adoption of Open Source Software in Software-Intensive Organizations - A Systematic Literature Review. Information & Software Technology 52(11): 1133-1154 (2010).
3. Bac, C., Berger, O., Deborde, V., Hamet, B. Why and how to contribute to libre software when you integrate them into an in-house application?, in: Proceedings of the First International Conference on Open Source Systems, 2005, pp. 113–118.
4. Stol, K., Ali Babar, M. Challenges in using open source software in product development: a review of the literature, in: Proceedings of the 3rd Workshop on Emerging Trends in FLOSS Research and Development, ACM, Cape Town,South Africa, 2010, pp. 17–22, 10.1145/1833272.1833276.
5. Aberdour, M. Achieving Quality in Open Source Software. IEEE Software. January-February 2007, pp. 58-64.
6. Nakakoji, K. Yamamoto, Y., Nishinaka, Y., Kishida, K., Ye, Y. “Evolution Patterns of Open-Source Software Systems and Communities,” Proc. International Workshop Principles of Software Evolution, ACM Press, 2002, pp. 76–85.
7. Dahlander, L., Magnusson, M.G., Relationships between open source software companies and communities: observations from Nordic firms, Research Policy 34 (4) (2005) 481–493, doi:10.1016/j.respol.2005.02.003.
8. Dahlander, L., Magnusson, M.G., How do firms make use of open source communities?, Long Range Planning 41 (6) (2008) 629–649, doi:101016/ j.lrp.2008.09.003.
9. Capra et al. A Survey on Firms’ Participation in Open Source Community Projects. OSS 2009.
10. Ven, K., Mannaert, H., Challenges and strategies in the use of open source software by independent software vendors, Information and Software Technology 50 (9-10) (2008) 991–1002, doi:10.1016/j.infsof.2007.09.001.

11. Oates, B.J., 2006. Researching Information Systems and Computing. Sage Publications. London.
12. Conradi, R.; Li, J.; Slyngstad, O.P.N.; Kampenes, V.B.; Bunse, C.; Morisio, M., Torchiano, M.; 2005. Reflections on Conducting an International Survey on Software Engineering. Proceedings of the International Symposium on Empirical Software Engineering (ISESE'05), IEEE CS Press, pp. 214-223.
13. Ayala, C., Hauge, Ø., Conradi, R., Franch, X., Li, J. Selection of Third Party Software in Off-The-Shelf-Based Software Development - An Interview Study with Industrial Practitioners. The Journal of Systems & Software 84 (2011), pp. 620-637.
14. Denger, C., Feldmann, R.L. Höst, M., Lindholm, C., Shull, F. A Snapshot of the State of Practice in Software Development for Medical Devices, Short paper at First International Symposium on Empirical Software Engineering and Measurement, September, 20-21, 2007, Madrid, Spain
15. Babar, A.M., Gorton, I.: Software Architecture Review: The State of Practice. IEEE Computer 42(7): 26-32, 2009.
16. Nguyen D.A., Cruzes, D., Conradi, R. Höst, M., Franch, X., Ayala, C.P.: "Collaborative Resolution of Requirements Mismatches When Adopting Open Source Components". Requirements Engineering: Foundation for Software Quality Lecture Notes in Computer Science, 2012, Volume 7195/2012, 77-93, DOI: 10.1007/978-3-642-28714-5_7
17. Sandelowski, M., Barroso, J. Handbook for Synthesizing Qualitative Research. Springer, 2007.
18. Choi, N., Chengalur-Smith, I.,Whitmore, A. Managing First Impressions of New Open Source Software Projects. IEEE Software. November-December 2010. pp., 73-77.
19. Capiluppi, A., Lago, P, Morisio, M. "Evidences in the Evolution of OS Projects through Changelog Analyses," Proc. 3rd IEEE Workshop Open Source Software Eng. (WOSSE 03), ICSE, 2003, pp. 10–24.
20. Li, J., Conradi, R., Slyngstad, O.P.N., Torchiano, M., Morisio, M., Bunse, C., A State-of-the-Practice Survey of Risk Management in Development with Off-the-Shelf Software Components. IEEE Transactions on Software Engineering, 34(2), 271-286, 2008.
21. Bonaccorsi, A., Rossi, C. Comparing motivations of individual programmers and firms to take part in the open source movement: from community to business, Knowledge, Technology, and Policy 18 (4), pp 40-64, 2006, doi:10.1007/s12130-006-1003-9.
22. Chen, W., Li, J., Ma, J., Conradi, R., Ji, J., Liu, C.,. An Empirical Study on Software Development with Open Source Components in the Chinese Software Industry. Software Process: Improvement and Practice 13(1), 89-100, 2008.
23. Glance, D.G., Kerr, J., Reid, A. Factors affecting the use of open source software in tertiary education institutions, First Monday 9 (2).
24. Hauge, Ø., Sørensen, C-F., Conradi, R. Adoption of open source in the software industry, in: Russo et al. Proceedings of the 4th IFIP Working Group 2.13 International Conferences on Open Source Software (OSS2008) – Open Source Development Communities and Quality, September 7th–10th, Milano, Italy, IFIP Advances in Information and Communication Technology, vol. 275/2008, Springer, 2008, pp. 211–222, doi:10.1007/978-0-387-09684-1_17.
25. Robles, G., Dueñas, S., González-Barahona, J.M. Corporate involvement of libre software: study of presence in debian code over time, in: Feller et al. [29], pp. 121–132, doi:10.1007/978-0-387-72486-7_10.

26. Aaltonen, T., Jokinen, J. Influence in the Linux kernel community, in: Feller et al. [29], pp. 203–208, doi:10.1007/978-0-387-72486-7_16.
27. Höst, M., Orucevic-Alagic, A., Runeson, P. Usage of Open Source in Commercial Software Product Development - Findings from a Focus Group Meeting, in proceedings of International Conference on Product Focused Software Development and Process Improvement (PROFES), pp. 143-155, 2011.
28. Robson, C., 2002. Real World Research: A Resource for Social Scientists and Practitioner-researchers. Second Edition. Blackwell Publishers Inc.
29. Feller, J., Fitzgerald, B., Scacchi, W., Sillitti, A. (Eds.), Proceedings of the 3rd IFIP Working Group 2.13 International Conference on Open Source Software (OSS2007) – Open Source Development, Adoption and Innovation, June 11th–14th, Limerick, Ireland, IFIP Advances in Information and Communication Technology, vol. 234/2007, Springer, 2007, doi:10.1007/978-0-387-72486-7.