An Exploratory Study of Contribution Barriers Experienced by Newcomers to Open Source Software Projects

Christoph Hannebauer Matthias Book Volker Gruhn

paluno – The Ruhr Institute for Software Technology
University of Duisburg-Essen
Gerlingstr. 16, 45127 Essen, Germany
{christoph.hannebauer | matthias.book | volker.gruhn }@uni-due.de

ABSTRACT

Contributing to a Free, Libre and Open Source Software (FLOSS) project is not a trivial task even for experienced developers: Beyond the effort required for understanding and editing a project's source code for one's own purposes, submitting the changes back to the community requires additional motivation, time, and social and technical effort. Although several surveys have examined the dynamics driving FLOSS contributors, most focus either on the motivations of core developers or indicators of potential long-term commitment, i.e. the small but quite involved and visible minority at the core of a project. Our survey in contrast examines the experiences of the much larger, but nearly invisible group of developers who are just making and submitting their first patch, and identifies barriers that hinder or even prevent them from making a valuable contribution.

Categories and Subject Descriptors

D.2.9 [**Software Engineering**]: Management—*programming teams*

General Terms

Economics, Human Factors

Keywords

FLOSS; contribution barrier; motivation; survey

1. INTRODUCTION

Integrating new developers into a software development project always comes with resource costs [1] – among other things, they need to familiarize themselves with the project's domain, requirements and vision, its architecture and technology, and its engineering tool set before they can contribute to the project effectively. In Free, Libre and Open Source Software (FLOSS) projects, these resource costs constitute a contribution barrier that new developers have to overcome before they can contribute bug fixes or new features.

In order to better understand the factors that the contribution barrier is comprised of, and to identify levers for reducing the contribution barrier, we conducted a survey of professional software developers who have modified the source code of a FLOSS application and asked them about the factors that motivated them, as well as those issues that complicated the contribution process or even prevented them from completing it. In contrast to previous surveys, we distinguished between factors that motivated them to modify an application and factors that motivated them to contribute modifications back to the project.

In this paper, we present the design of the survey along with the answers of our participants (Sect. 3), the analysis of the answers (Sect. 4), and conclude with recommendations on how to lower identified contribution barriers in FLOSS projects (Sect. 5).

2. RELATED WORK

Von Krogh et al. first described the concept of a "contribution barrier" in their analysis of the Freenet project [14]. They explained that for each module of a FLOSS project, the contribution barrier is a set of four specific hurdles that prevent new developers from modifying that module.

We use the term "contribution barrier" more broadly in this paper, so that it comprises all kinds of social and technical hurdles, not only the four specific hurdles von Krogh et al. described. Given that many of a new developer's first steps occur in relative isolation, we did not rely on documented communication, but instead asked developers directly about their experiences.

A number of surveys already asked software developers about their motives to contribute to FLOSS projects. According to Hertel et al. [5] and Hars and Ou [4], the enjoyment of programming was a major motivator for most developers. Besides this, pragmatic reasons such as needing a modification for their own project were most often mentioned not just in Hertel et al.'s, but also Lakhani and Wolf's survey [7]. Improvement of one's own programming skills was also frequently cited in these studies, and seems to be an important factor in starting FLOSS project involvement according to Gosh [2].

Krishnamurthy pointed out that these surveys do not dif-

CSI-SE'14, June 2, 2014, Hyderabad, India
ACM 978-1-4503-2857-9/14/06
http://dx.doi.org/10.1145/2593728.2593732

ferentiate between the different types of tasks in FLOSS projects [6]. Thus, we distinguish between the motivation for private source code modifications and the motivation for public submissions back to a project. In addition, there is only little explicit examination of FLOSS projects' contribution barriers, i.e. the forces that oppose developers' motivation. One counterexample is Steinmacher et al.'s analysis of contibution barriers in the Hadoop project [12]. Finally, all current surveys we are aware of only looked at developers that are already co-developers or core developers in FLOSS projects, i.e. people who have mastered the contribution process. Their results are therefore likely biased towards the views of a small, but very visible minority of expert contributors, while the much larger group of "newbie" or prospective FLOSS developers is much less visible and thus harder to reach. In our survey, we therefore deliberately asked professional developers whose primary focus is not open source about their experience with adapting and contributing to FLOSS projects.

3. QUESTIONNAIRE

To get a realistic impression of the challenges developers face as they take the first steps toward contributing to a FLOSS project, we did not look at mailing lists, bug reports and other public archives of project communication, as we expected impressions from these groups to be biased by the large amount of experienced FLOSS developers active on these channels.

Instead, we created an online questionnaire that we sent to 664 professional software developers from six organizations (five software companies and one university). These developers work on commercial software development projects across a variety of technologies and domains. Most of these people are very experienced developers who use open source software in their projects, but their employers have no particular policy or focus on making active contributions to FLOSS projects, so there is no institutionalized support for making contributions, and any such endeavors stem from the employees' own motivation. We believe such an environment is a rather common starting point for most potential FLOSS contributors.

Out of this audience, we received 22 surveys in response. Our invitation to the online questionnaire explained that anyone who has modified the source code of a FLOSS application may fill out the questionnaire. Thus, a possibly large fraction of the invited software developers had never modified the source code of a FLOSS application and therefore refrained from participating in the survey. This explains the relatively low return ratio of about 3%. It means, however, that findings need to be interpreted with a grain of salt. While we did not receive sufficient replies for a representative picture, we believe that our target population was reasonably broad, and that the answers already provide indications of common contribution barriers that should be fixed.

3.1 FLOSS Experience Level

We first asked respondents to categorize their FLOSS contribution experience. Participants that had never modified the source code of a FLOSS project were excluded from the rest of the questionnaire. The remaining participants had modified the source code of a FLOSS project. Some of them had also submitted a patch with their modifications back to a FLOSS project.

Out of the 22 responses, four indicated that they had made no modifications to FLOSS projects yet. Five developers did not answer any further questions, so 13 respondents who completed the whole questionnaire remained. Out of these, four developers indicated that they had only modified FLOSS source code, while nine developers responded that they had successfully submitted patches to projects.

We asked them which project they modified, and in which year the modification was made. While many existing surveys focus on a small number of specific FLOSS projects and thus may be biased by that project's particular barriers and group dynamics [13], all our respondents indicated that they worked on different FLOSS projects, so their experiences should be more generalizable. The majority of modifications was made within the last 1.5 years before the survey, while some changes also reached back as far as 2004.

3.2 Developers' Motivation

In order to distinguish the different forces motivating the modification of and submission to a FLOSS project, as well as those hindering contributions, we asked all participants about their general relationship to the project, and their reasons for their first modification of its source code. Those who had also submitted their modifications back to a FLOSS project were also asked about their reasons for the submission.

The initial reason for most developers' general involvement with a FLOSS project was the use of the project itself – either as a tool they used as end-users, or as a component they integrated into their own system. Other reasons such as providing consulting for the FLOSS project, making it interoperable with one's own products, or evaluating the project played only a minor role for the developers we surveyed.

In the next question, participants could rank their reasons to *modify* the software. The answers show a clear picture: The majority of developers made modifications because a malfunction or missing feature bothered them (option b), because they wanted to acquire experience with the project's technologies (a), and because they enjoyed the intellectual challenge (c). Satisfaction derived from working with the project's community (d) and submitting patches (e) were only minor motivators. In an interesting case, one participant responded that he wanted to "preserve part of [his own] project work spent on the usage of [the FLOSS component]," i.e. he refactored work originally performed outside the FLOSS component into it, so it could be reused in future endeavors. Fig. 1 illustrates how respondents ranked the possible options (multiple selections were possible, only highest ranks 1-3 shown). These results agree with Raymond's observation that most work starts from "scratching a personal itch" [9].

Afterward, participants could rank their reasons to *submit* their modifications to the FLOSS project. These reasons are more varied than the reasons for modifying a FLOSS component, as Fig. 2 indicates: Having one's changes reviewed by others (option g) and avoiding the work of reintegrating one's changes into new releases of the OSS project (b) were the most frequently mentioned and highest-ranked reasons, with a feeling of obligation to return something (i) being a close third. This is an interesting mix of motivations spanning aspects of learning (g), pragmatics (b) and ethics (i).

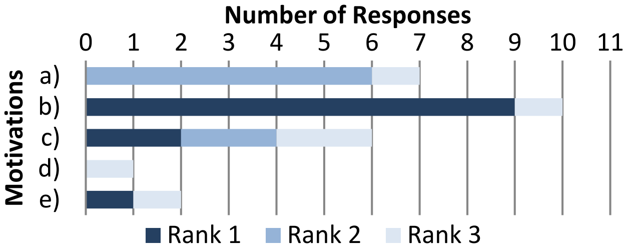

Figure 1: Number of ranked responses for modifying the source code of a FLOSS project: experience (a), own need (b), joy (c), social (d), submission (e)

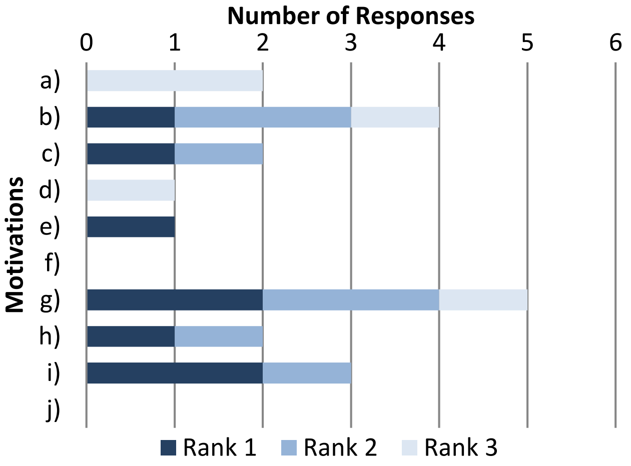

Figure 2: Number of ranked responses for submitting patches to a FLOSS project: altruism (a), avoid branching (b), experience (c), visibility (d), respect (e), licensing (f), reviews (g), FLOSS idea (h), obligation (i), fighting closed source software (j)

We hypothesize that the complementarity of these factors, between which no compromise is necessary, together creates a quite strong motivation that offsets the time and effort involved in submitting a patch.

Other motivating factors were gaining experience with FLOSS procedures (c), belief in the open source idea (h) and altruism (a), as well as (to a lesser degree) community respect (e) and personal visibility (d). Licensing requirements (f) or an aversion against closed-source software (j) did not play a role among our respondents.

Those participants who had not tried to submit their private modification back to the FLOSS project were asked about the reason for this. The few developers in our survey who had decided against contributing justified this with too high expected effort, or the impression that others would have no use for their modification.

3.3 Contribution Barrier

Next, we addressed the contribution barrier experienced by the developers. We asked all respondents about the time they spent on each task among a list of tasks involved with a FLOSS contribution. The bandwidth of answers in our survey is shown in Fig. 3. Finding the cause of a problem and the right location in the source code (FND) consistently required one of the highest efforts, with writing the code to solve the problem (SLV) a close second. Obtaining the current version of the source code (DL), reproducing the problem (REP), creating the patch (PAT), and submitting it (SUB) took more manageable time. The highest maximum and average time effort was spent on building the system (BLD).

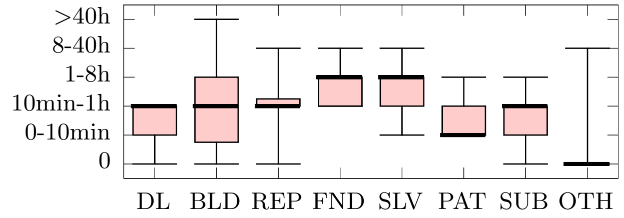

Figure 3: Time spent on tasks required for modifying and submitting parts of a FLOSS project

Following these quantitative effort estimates, we asked those participants who had tried to submit a patch to the FLOSS project about their qualitative experiences with the contribution barrier with open-ended questions. Of the respondents who successfully submitted a patch to a FLOSS project, two answered that they did not experience any barriers. Among the remaining answers, the by far most frequently mentioned barrier was the setup of the development environment in one form or another.

4. SURVEY ANALYSIS

The survey results indicate that a major part of the contribution barrier is the effort invested until the first build, and is also in agreement with Fitzpatrick, who suggests that the first build is the hardest step in joining any software development project [11, p. 70f].

At first sight, the difficulty in setting up the build system may be surprising, since the participants of this survey are professional software developers who are familiar with build systems and should already have a development environment installed and configured on their machine. There are two explanations why the familiar development environments do not suffice for the modifications of the source code of some FLOSS projects. The survey gives merely hints on which explanation best describes the situation of the participants.

Firstly, developers may not have used the programming language of the FLOSS project before, and need a new development environment for the new language. Often, the need for the FLOSS component may stem from project requirements, but in our survey, all participants who found the setup of the development environment to be the biggest contribution barrier also indicated that they had started to work on the FLOSS component because they wanted to acquire experience in the technologies it uses. This indicates that developers are willing to tackle the contribution barrier even if the main reason for working on the FLOSS component is not a pressing project need, but a learning endeavor.

Secondly, even if the FLOSS project uses a programming language that the existing development environment supports, the core developers' platform may be different from that of the new software developers, making a platform-specific setup necessary. An example for a FLOSS project with such limitations is Mozilla: Depending on the subproject and the branch of the subproject modified, one version of the same development environment may be compatible while another is not [8].

Easy access to the source code was cited as another contribution barrier by our respondents. One participant stated that she had been discouraged from submitting patches earlier because there was “no browse access to the ‘trunk’ version of the source code”. Making it easier for developers to get all code, tools and information they need should therefore be part of a project’s strategy for encouraging new contributors, especially given that the technological and intellectual barriers further down the process are still hard enough, as discussed below.

The primary effort drivers, finding (FND) and fixing (SLV) the problem, are independent of FLOSS projects – they mirror the general intellectual challenge of programming, and are addressed by research on program comprehension, bug fixing, and related fields. Among the steps that are more specific to the context of FLOSS contribution, the effort for building the system (BLD) stands out in particular.

As a solution to the challenges associated with building binaries from the code, closed source projects often use Continuous Integration (CI) systems. These CI systems automatically check whether modifications are compatible with each other, and test them on a broad range of platforms. A missing CI system can be a contribution barrier, as one developer in our survey indicated:

> “My productive system had a very special configuration (including dated versions of some of the components) and I was unable to test my fix against alternative configurations.”

While large FLOSS projects usually employ a CI system, small FLOSS projects usually rely on software forge services such as Sourceforge and github for key infrastructure such as Version Control Systems (VCSs) and bug trackers [10]. These software forge services however do not provide CI to FLOSS projects [3].

5. CONCLUSION

In this paper, we presented the results of a survey of professional software developers to identify components of the contribution barrier that newcomers to FLOSS projects encounter. In contrast to other surveys, we deliberately did not target typical FLOSS communication channels (mailing lists, bug trackers etc.) which might have biased the results to a more expert-dominated view, but surveyed developers outside the established communities who were asked about their experiences in making their first modifications and contributions in FLOSS projects. This way, we identified two prominent components of the contribution barrier that work as opposing forces to developers’ motivation to contribute:

- The effort necessary for the first build of an executable application from the source code is the biggest part of the contribution barrier.
- Missing unit tests as provided by CI systems further increase the contribution barrier.

We suggest providing preconfigured build environments [15] to lower the effort for the first build. Public CI services [3] can provide the missing unit tests.

6. REFERENCES

[1] F. P. Brooks, Jr. *The Mythical Man-Month.* Addison Wesley, 20th anniversary edition edition, 1995.

[2] R. A. Ghosh. Understanding Free Software Developers: Findings from the FLOSS Study. Technical report, MERIT/Institute of Infonomics, University of Maastricht, 2005.

[3] V. Gruhn, C. Hannebauer, and C. John. Security of public continuous integration services. In *Proceedings of the 2013 Joint International Symposium on Wikis and Open Collaboration (WikiSym + OpenSym 2013).* ACM, 2013.

[4] A. Hars and S. Ou. Working for free? Motivations of participating in open source projects. In *Proceedings of the 34th Annual Hawaii International Conference on System Sciences, 2001*, page 9 pp., jan. 2001.

[5] G. Hertel, S. Niedner, and S. Herrmann. Motivation of software developers in open source projects: an internet-based survey of contributors to the linux kernel. *Research Policy*, 32(7):1159 – 1177, 2003. Open Source Software Development.

[6] S. Krishnamurthy. On the intrinsic and extrinsic motivation of free/libre/open source (FLOSS) developers. *Knowledge, Technology, & Policy*, 18:17–39, 2006. 10.1007/s12130-006-1002-x.

[7] K. Lakhani and R. G. Wolf. Why hackers do what they do: Understanding motivation and effort in free/open source software projects. *Social Science Research Network*, pages 1–27, 2003.

[8] Mozilla Developer Network. Windows build prerequisites (1.9.0 and 1.9.1 branches), July 2011. `https://developer.mozilla.org/en-US/docs/Windows_Build_Prerequisites_%281.9.0_and_1.9.1_branches%29` [accessed 2014-01-02].

[9] E. S. Raymond. *The Cathedral and the Bazaar.* O’Reilly & Associates, Inc., Sebastopol, CA, USA, 1st edition, 1999.

[10] D. Riehle, J. Ellenberger, T. Menahem, B. Mikhailovski, Y. Natchetoi, B. Naveh, and T. Odenwald. Open collaboration within corporations using software forges. *IEEE Software*, 26:52–58, 2009.

[11] P. Seibel and B. Fitzpatrick. Brad Fitzpatrick. In *Coders at Work*, chapter 2, pages 49–90. Apress, Berkely, CA, USA, 2009.

[12] I. Steinmacher, I. Wiese, A. Chaves, and M. Gerosa. Why do newcomers abandon open source software projects? In *6th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE), 2013*, pages 25–32, May 2013.

[13] T. Østerlie and L. Jaccheri. A critical review of software engineering research on open source software development. *Proceedings of the 2nd AIS SIGSAND European Symposium on Systems Analysis and Design*, 2007.

[14] G. von Krogh, S. Spaeth, and K. R. Lakhani. Community, joining, and specialization in open source software innovation: A case study. *Research Policy*, 32(7):1217–1241(25), July 2003.

[15] V. Wolff-Marting, C. Hannebauer, and V. Gruhn. Patterns for Tearing Down Contribution Barriers to FLOSS Projects. In *Proceedings of the 12th SoMeT*, Sept. 2013.