Understanding and Supporting the Choice of an Appropriate Task to Start With In Open Source Software Communities

Igor Steinmacher
Department of Computing
Federal University of Technology-Paraná (UTFPR)
igorfs@utfpr.edu.br

Tayana Uchôa Conte
Computing Institute – Federal University of Amazonas (UFAM)
tayana@icomp.ufam.br

Marco Aurélio Gerosa
Institute of Mathematics and Statistics
University of São Paulo (USP)
gerosa@ime.usp.br

Abstract

Open Source Software (OSS) projects leverage the contribution of outsiders. In fact, the sustainability of many projects relies on retaining some of these newcomers. Usually these communities do not coordinate the work of the newcomers, who go to the issue trackers and self-select a task to start with. We found that "finding a way to start" was reported as an important issue by practitioners. To further investigate this specific barrier, we conducted a qualitative analysis with data obtained from semi-structured interviews with 36 subjects from 14 different projects, including newcomers and experienced members. We used procedures of Grounded Theory – open and axial coding – to analyze the data. We found that newcomers are not confident enough to choose their initial task and they need information about the tasks or direction from the community to support choosing a task more suitable for them. We also present a set of strategies identified in the literature, interviews, and state-of-the-practice that can provide newcomers with such information, enabling them to be more confident when choosing their first tasks and collaborate with the community.

Keywords: Open Source Software, coordination, task selection, newcomers, onboarding, new developer, joining process

1. Introduction

Open Source Software projects rely on geographically distributed developers working as a team and incorporating their individual creations into a single, seamless body of source code [25]. Many OSS projects leverage contributions from volunteers and require a continuous influx of newcomers for their survival, long-term success, and continuity. According to Qureshi and Fang [21], it is essential to motivate, engage, and retain new developers to promote a sustainable number of developers in a project.

However, newcomers usually face many difficulties to make their first contribution to an open source project. Dagenais et al. [8] compare software project newcomers to explorers who need to orient themselves in a hostile environment. In OSS projects, newcomers are usually expected to learn about the project on their own [22]. Therefore, a major challenge for OSS projects is to provide ways to support newcomers' joining.

Understanding developer motivation and project attractiveness are well-explored topics in the literature [9, 17, 24, 36, 37]. However, little is known about the barriers that newcomers face when placing their first contribution in a project, while onboarding [36].

By conducting a qualitative study with data obtained from interviews with practitioners, we identified 58 barriers faced by newcomers when onboarding to OSS projects from interviews with OSS practitioners [26, 27]. Among these barriers, a barrier called "difficulty to find a task to start with" called our attention. We identified this barrier in interviews of participants from seven different projects, including experienced members, newcomers, and people who gave up onboarding. Moreover, evidence of this barrier was also identified in other studies conducted over different data sources: feedback from students that onboard OSS projects; answers to an open question submitted to OSS developers [29]; and systematic literature review on barriers to onboard in OSS projects [28]. This recurrence made us recognize the necessity to further investigate it.

"Difficulty to find a task to start with" is a problem inherently related to the coordination of OSS projects. Members of OSS projects usually coordinate their tasks by using issue trackers such as Bugzilla, Jira, etc. These issue trackers are systems where any user or developer is free to report, discuss, and choose a task to work on. When newcomers want to contribute, they usually access the issue tracker to choose a bug or a feature request that they can (or want to) handle. However, as reported by a newcomer *"it is a bit frustrating trying to find something I could do or fix."*

Thus, in this paper, we focus on this barrier, aiming at better understanding it and proposing a set of strategies to alleviate its problems. To achieve this, we

1530-1605/15 $31.00 © 2015 IEEE
DOI 10.1109/HICSS.2015.624

analyzed data from interviews conducted with 36 OSS practitioners from 14 different projects. We started our investigation aiming at answering a broad question ("What are the barriers that hinder newcomers' onboarding to OSS projects?") and, as the barrier called our attention, we decided to focus on it by analyzing the existing data and conducting another round of interviews. We used a qualitative research based on procedures of grounded theory to analyze the data. The analysis resulted in a model containing 14 concepts that help explaining the barrier. Moreover, analyzing the literature, the interviews, and the state-of-practice, we identified a set of strategies that can support newcomers overcoming this barrier.

This paper is structured as follows. Section 2 introduces the related research. Section 3, the research method. The results are presented in Section 4. A set of strategies to support the newcomers is presented in Section 5. Finally, Section 6 presents the conclusions and future directions.

2. Related Work

Newcomers' onboarding is an issue faced in many online communities. Many studies in the literature deal with newcomers joining process in collective production communities, including studies on Wikipedia [2, 10, 34] and on OSS projects [4, 11, 12, 15, 16, 20, 27, 29, 30, 33, 35]. Dagenais et al. [8] and Begel and Simon [1] present studies regarding newcomers joining process in industrial software projects.

Von Krogh et al. [15] analyzed interviews with developers, emails, source code repository, and documents of the FreeNet project. The authors proposed a joining script for developers who want to take part in the project. Nakakoji et al. [18] studied four OSS projects to analyze the evolution of their communities. They presented eight possible roles for the community members and structured them into a model composed of concentric layers, like the layers of an onion. This structure was later called the onion patch, and other authors conducted studies based on this model [12–14]. Although these papers deal with the evolution of members' participation in OSS communities, they focus on newcomers after the onboarding.

Some researchers tried to understand the barriers that influence the retention of newcomers. Zhou and Mockus [38] worked on identifying the newcomers who are more likely to remain in the project in order to offer active support for them to become longterm contributors. Jensen et al. [12] analyzed mailing lists of OSS projects to verify if the emails sent by newcomers are quickly answered, if gender and nationality influence the kind of answer received, and if the reception of newcomers is different in users and developers lists. Steinmacher et al. [30] used data from mailing list and issue tracker to study how reception influences the retention of newcomers in an OSS project. Park and Jensen [20] show that visualization tools support the first steps of newcomers in an OSS project, helping them to find information more quickly.

Mentoring is also explored as a way to support newcomers. Malheiros et al. [16] and Canfora et al. [4] proposed different approaches to identify and recommend mentors to newcomers of OSS projects by mining data from mailing lists and source code versioning systems.

Finding the appropriate task is usually classified as a problem because new developers have difficulty to find bugs or features that are of interest, that match their skill sets, are not duplicates, and are important for their future community [35]. Von Krogh et al. [15] found that in 56.7% of the cases members of the community encouraged the new participants to find some part of the software architecture to work on that would match with their specialized knowledge. In only 16.7% of the cases, new participants were both encouraged to join and given specific technical tasks to work on. This occurs because, according to their interviews, the community expects new participants to find their own task to work on instead of receiving a specific piece of work.

Park and Jensen [20] reported that "... subjects expressed a need for information specific to newcomers, for instance, how to get involved and become active (e.g. communication channels, available sources of information for starters, etc.), what to contribute to (e.g. open issues, required features, sample tasks to start with), and working practices." Capiluppi and Michlmayr [5] after looking at project history reported that new developers, when joining the project, tend to work more easily on new modules than on older ones, and potential new developers should be actively fostered adding new ideas or directions to the project.

Regarding support when dealing with tasks, Čubranić et al. [7] presented Hipikat, a tool that supports newcomers by building a group memory and recommending source code, emails messages, and bug reports to support newcomers. Wang and Sarma [35] presented a Tesseract extension to enable newcomers to identify similar bugs through synonym-based search and visually explore the appropriate socio-technical dependencies for a bug in an interactive manner. These tools can help newcomers by increasing their awareness about the tasks and support.

From the literature analysis, we could observe that, from the community's perspective, newcomers should

be able to find the most appropriate task themselves, as reported by von Krogh et al. [15]. However, other researches showed that the projects should give special attention to this issue and support the newcomers finding their tasks. The goal of this work is to better understand this problem from both perspective, to enable researchers and community to focus on creating strategies that effectively support newcomers to OSS projects.

3. Research Method

We chose to conduct a qualitative research to understand the difficulties to find an appropriated task and proposing a set of strategies to alleviate the problems caused by this barrier. Qualitative research produces results that cannot be achieved through statistical procedures or similar methods [31]. The results of this kind of approach are richer and more informative, helping to answer questions involving variables that are difficult to quantify, such as human characteristics like motivation and perception [23].

3.1. Data Collection and Analysis

We used semi-structured interviews as data collection method. Semi-structured interviews have an intermediate control level, and comprise a mix of closed and open questions, making it possible to collect the information the researcher is seeking and also unpredicted ones [23].

The participants were recruited primarily through mailing list and forums from projects. We also invited newcomers and project owners directly, identifying them in project pages and by mining and following projects' mailing lists and issue trackers. We made no distinction related to gender or nationality. Additionally, participants needed to have software development experience, because we were interested in barriers for contributing to OSS projects and not for learning how to develop software.

We recruited subjects that belonged to four different groups:

- **Experienced members:** project owners, managers, or developers that commit code directly to the software repository;
- **Succeeded newcomers**: participants that started to contribute to the project less than one year before the interview;
- **Dropout Newcomers**: volunteers that tried to contribute to the project, but gave up;
- **Onboarding Newcomers**: volunteers that were trying to place their first contribution.

We interviewed 36 OSS developers from 14 different projects, including 11 experienced members, 16 newcomers that succeeded, 6 dropout newcomers, and 3 newcomers that were still trying to place their first contributions. Some more information about their profile are presented in Table 1. The participants received an ID, shown in the first column. The first character of the ID represents the profile of the participant: "E" for Experienced member, "N" for Succeeded newcomer, "D" for Dropout newcomer, and "O" for Onboarding Newcomer.

The interviews were conducted using textual based chat tools, like Google Hangouts, since the participants are used to this kind of tool for their professional and personal activities. Each interview was conducted individually and the data was logged in a local computer. Interviews began with a short explanation of the research, followed by some questions to profile the interviewees regarding their technical experience and main occupation.

Table 1. Profile of the participants ***(H = Hours per week in OSS; F = First Project?; C = Country; Y = Years in the project)***

ID	*H*	*F*	*Project*	*C*	*Y*	*Collection*
E1	< 5	N	JabRef	FR	8	I1
E2	05-10	Y	Etherpad	DE	3	I1
E3	10-20	N	JabRef	DE	3	I1, I2
E4	05-10	N	jEdit	CA	10	I1
E5	05-10	N	LibreOffice	DE	15	I1, I3
E6	> 20	N	LibreOffice	HU	10	I1, I2
E7	> 20	N	Moodle	AU	5	I1
E8	> 20	N	Noosfero	BR	5	I1
E9	> 20	N	Pardus	TR	8	I1, I3
E10	05-10	N	Cogroo	BR	5	I1
E11	< 5	N	Noosfero	BR	7	I1
N1	< 5	Y	JabRef	DE	0	I1, I3
N2	< 5	Y	Gephi	BR	0	I1
N3	05-10	Y	Gephi	IN	1	I1
N4	05-10	Y	Moodle	IN	0	I1
N5	< 5	Y	JabRef	DE	0	I1
N6	< 5	Y	jEdit	US	0	I1, I2
N7	< 5	Y	TextMate	US	0	I1
N8	> 20	Y	Zxing	GR	0	I1
N9	< 5	Y	Cogroo	BR	0	I1
N10	< 5	Y	Integrade	BR	0	I1, I2
N11	< 5	Y	Cogroo	BR	0	I1
N12	N/I	N	Etherpad	UK	0	I1
N13	10-20	N	LibreOffice	BR	1	I1, I2, I3
N14	05-10	Y	LibreOffice	BR	1	I1
N15	N/I	Y	Etherpad	FR	0	I1
N16	05-10	N	JabRef	DE	0	I1
D1	05-10	N	Hadoop	US	0	I1
D2	< 5	Y	Hadoop	IN	0	I1
D3	< 5	N	JabRef	DE	0	I1
D4	< 5	Y	OpenOffice	BR	0	I1
D5	< 5	Y	LibreOffice	IN	0	I1
D6	< 5	Y	OpenOffice	IN	0	I1
O1	< 5	N	OpenOffice	IN	0	I1
O2	10-20	Y	LibreOffice	CN	0	I1, I2
O3	< 5	Y	OpenOffice	GR	0	I1

The first step of the study consisted of a first iteration of interviews (represented as It1 in the last column of Table 1) with the participants to answer a broad question ("What are the barriers that hinder newcomers' onboarding to OSS projects?"). The interviews of this step focused on understanding the process of onboarding to OSS project, verifying the importance of receiving newcomers and identifying the barriers from newcomers and experienced members' perspective. During the analysis of these interviews, we recurrently needed to clarify some doubts to better understand some information. Therefore, we contacted some of the interviewees and conducted few other interviews with them (represented as It2 in the last column of Table 1).

We qualitatively analyzed the data using procedures of Grounded Theory (GT) [31], which is based in the concept of coding. Coding means attaching codes, or labels, to pieces of text which are relevant to a particular theme or idea, grouping and examining the ideas to explain a phenomenon [23]. Coding can be divided into three steps: open coding, where concepts are identified and their properties and dimensions are discovered in the data; axial coding, where connections between the codes are identified and grouped according to their properties to represent categories; and selective coding, where the core category is identified and described.

Although the purpose of the GT method is the construction of substantive theories, its use does not necessarily need to remain restricted only to researches with this goal. According to Strauss and Corbin [31], the researcher may use only some of its procedures to meet one's research goals.

We used ATLAS.ti[1] to perform the coding. During open coding, we assigned codes to sentences, paragraphs, or revisions. This procedure overlapped the axial coding, in which connections between the categories were identified. In practice, open and axial coding were executed several times to refine the emerging codes and categories.

During the analysis of the first set of interviews, we learned about different barriers. We encouraged the interviewees to constantly report barriers they faced or observed and to give more details on each barrier reported. After performing the open coding and axial coding we were able to identify 58 different barriers, grouped in six categories, some of them split in subcategories, as shown in Table 2.

Table 2. Catalog of categories and subcategories that emerged from the interviews

Category	Sub Category	# of Barriers Identified
Newcomers' Characteristics	Newcomers behavior	9
	Newcomers technical background	5
Newcomers Need Orientation	**Find a way to start**	**6**
	Find a mentor to provide a guided tour	2
	Find/follow suggested tours	3
Technical Hurdles	Code/architectural hurdles	8
	Hurdles to submit changes	4
	Local environment setup hurdles	4
Cultural Differences		2
Reception Issues		4
Documentation Problems		10

As our work progressed, we decided to focus on a recurrent and specific category of barriers highlighted in the interviews: find a way to start (OR.1). More specifically, we are interested in better understanding the barrier called difficulty finding a task to start with, once the barrier was recurrently evidenced, appearing in 12 interviews. To investigate this specific barrier we conducted an in-depth analysis of the existing data, and conducted another iteration of interviews. In this step, we conducted interviews with five participants (represented as It3 in the last column of Table 1). The results of this analysis are presented in the following section.

4. Difficulty to Find a Task to Start With

The result of our analysis was the emergence of concepts that enabled us to better understand this barrier. These concepts represent a more detailed view on the reasons why newcomers understand the choosing or finding an appropriate task as a barrier to onboard to the project. The resulting model comprising the concepts and relationships among them is presented as a network in Figure 1.

The phenomenon we are trying to understand in this case is the "Difficulty to choose/find their tasks." Five issues emerged as part of the barrier under investigation, and three problems that are consequences of this barrier.

Some developers reported the lack of direction at the beginning as part of the explanation why is it difficult to find a task to work on. We found that some newcomers expect that someone indicate them what are the tasks they should start with. To explain this, one of them reported:

"We feel more secure when someone is guiding us. I think this is related to the experience with large projects and Open Source… Less experienced developers always stand on the back foot"

[1] http://www.atlasti.com

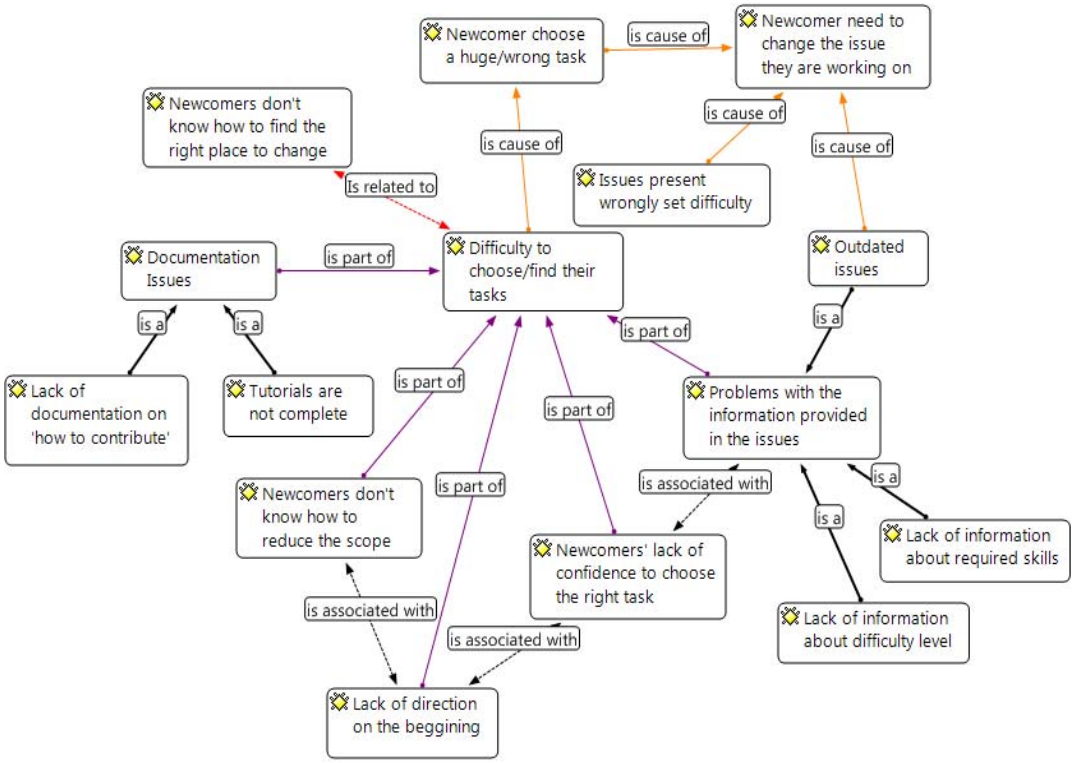

Figure 1. Result of qualitative analysis represented as a network

A newcomer that gave up because he could not identify a proper task explained:

"Fixing issues I thought would be easy ... but, then I really didn't know which one to pick... whether they are important ... Basically I was not active contributor at that time ... if some meaningful direction could be provided then I would have started." – [D2]

As it is possible to observe in the previous quote, this issue is related to the **lack of confidence** that newcomers have when trying to choose the task. The newcomer said, "*I really didn't know which one to pick, which one I was 'authorized' to pick... whether they are important*." It shows he was not sure about what he could do, what would be enough and important. Another newcomer told us a case of a close friend that tried to join the project:

"A college colleague gave up… He did not find a task that he felt confidence to try."

This occurs because newcomers, even when they are aware they can choose a task, they are not sure about what are easy or not, and what they can touch:

"it is frustrating, there is a bunch of issues, but I don't have the proper knowledge to judge what exactly I can touch…"

In the other side, experienced members reported that **newcomers do not know how to reduce the scope**. However, none of them put it as a newcomer's fault. One said: "*sometimes they want to contribute but don't know how to reduce the scope when starting*." Another practitioner [E6] reported:

"the task chosen [by the newcomer] makes some sense, but is huge, and the newbie thinks she'll be able to implement it in a few days."

The same participant reported the association among the inability of reducing the scope and lack of direction when asked if he had seen some case in which a newcomer gave up without finishing an assignment:

"When they choose a task that is reasonably sized, just we don't give the necessary help."

Some interviewees also mentioned **documentation issues** as part of the difficulty to find a task to work. We could identify two kinds of issues from the interviews: **lack of documentation on how to contribute** and **incomplete/outdated tutorials.** These issues were mainly reported by newcomers that recently joined a project. One of the them summarizes the first one when he says that "*there is no good guide for starter*" [O1]. Two other newcomers, who were onboarding to the project when we conducted the interviews, reported a hard time with the incomplete tutorials:

"A proper up-to-date guide with tutorials would have really helped. As in when I began, I was totally new to open source, but the tutorials skipped over certain instructions and I had no idea what I was doing wrong." – [D6]

"the tutorial's first task is where the problems began. [Project member] helped me get past this task. He said that the guide is outdated and guided me through this particular task. And I tried to do the examples but I started getting errors again and that's where I have stopped for now" – [O3]

Problems with the information provided in the issues were recurrently reported by newcomers and experienced members. From their discourse, we could clearly distinguish three types of issues: **outdated issues**; **lack of information about required skills**; and **lack of information about difficulty level.** An experienced member reported that the lack of information frustrated him when he was trying to contribute to a project:

"…the issues that I can contribute were not clearly defined so it was my job as a newcomer to find out how to contribute there were no easy or junior bugs as in some projects." [E9]

However, defining this as an easy bug or an issue as good for newcomers is sometimes not enough. A newcomer also evidenced that the skills needed to accomplish that task would be clear:

"…the issues should indicate the area of knowledge, like C++, build, shell script" [N1]

In addition to the issues that help us understanding the difficulties newcomers face to find an appropriate task, we also evidenced other issues caused by this barrier. We found that, due to these difficulties, **newcomers choose a wrong/huge task**, as per this quote from an experienced member: "*most of them [newcomers] do not know how to start... what can they do first and can choose the wrong task*" [N9]. A newcomer that was contributing for one year to a project reported this problem, accrediting the problem to a mistake when the issue was classified as easy:

"Sometimes, I tried to work on a task that was classified as easy hacks and it was too complex that only experienced members could find the solution… the developers who registered these easy hacks, sometimes made mistakes when classifying the difficulty" [N13]

Another newcomer reported his difficulties finding a task. The coordination mechanism used by the project failed to support his choice. He reported that he had to change the task twice, but for different reasons:

"when I finally did have something to do, it basically completely changed two times … one task was too hard and the other was a feature that was already implemented, but the task was not updated" [N8]

When searching for an issue to work on, newcomers want to be aware of some details about the issues; mainly they want to know whether they will be able or not to handle the task, what links this kind of problem with the **lack of confidence to choose a task**. The way they do this is looking at the issue tracker, aiming at coordinating their contribution with the community. Therefore, newcomers need up-to-date information to support their decision, to make it easier and clearer for them to be aware of what can they do, what they are able to do, where they can look for (or who can provide) support. The main point now is to identify potential strategies to improve this coordination mechanism in a way the newcomers can feel more comfortable in choosing their first task. In the next section, we present some strategies that can support the orientation of newcomers to overcome this barrier.

5. Orienting Newcomers Finding their Way

More than highlighting the barriers and its reasons, we also propose a set of strategies to orient the newcomers when they are trying to find their way. Using the metaphor by Dagenais et al. [8], newcomers are explorers that need to orient themselves in an unknown landscape. Therefore, our goal is to indicate what maps, signposts or features can be placed in the OSS landscape to support newcomers overcoming this barrier. To propose these strategies, we used three different sources: literature, interviews, and state-of-practice. We present these strategies in the following subsections.

5.1. Solutions found in literature

While conducting a Systematic Literature Review [28] and analyzing the related works, we identified possible strategies to support newcomers. We could identify two papers proposing strategies to support newcomers finding their way to start. Cubranic et al. [7] proposed Hipikat, a tool that recommends similar bugs and a set of files that will probably be used to accomplish that task. Wang and Sarma [35], proposed a Tesseract extension to enable newcomers to identify bugs of interest, resources related to that bug, and visually explore the appropriate socio-technical dependencies for a bug in an interactive manner.

These tools can give the newcomers more information about the issues by integrating data from other sources available in the projects. By providing more information, the tools increase newcomers'

awareness about the difficulty and skills needed to address the issues. Newcomers can look at similar issues, related mailing threads, and possible artifacts that need to be changed. Moreover, the strategy main goal is on supporting the newcomer overcoming the barrier "Newcomers do not know how to find the right place to change."

In a different sense, Wolff-Marting et al. [36] proposes a pattern related to the difficulty to find a task to start with in OSS projects. The solution proposed in this pattern is to "save some low-hanging fruit for beginners," classifying some tasks as easy problems, and leaving them for the newcomers who want to contribute to the project.

5.2. Solutions Identified During Qualitative Analysis

During the analysis of the interviews, we also have identified and coded the solutions proposed by the participants. The method applied to gather these solutions was the one presented for identifying the barriers, following procedures of Grounded Theory. To make the identification possible, we have explicitly included a question about possible solutions to the barriers in the interview script.

The analysis led us to identify two different kinds of strategies from the discourses of the interviewees: suggested strategies; and existing strategies that are already in place. In Table 3, we list what are the existing practices and new suggestions reported.

Table 3. List of solutions that emerged from the qualitative analysis of interviews

	Existing	Suggestion
Classifying tasks by difficulty/complexity	•	•
Including information about skills needed to work on the task		•
Presenting code examples in the easy issues	•	
Step-by-step tutorials pointing to easy issues	•	•
Providing Mentorship		•
Listing the areas of the project that good for newcomers		•

Classifying the tasks by difficulty, including information about skills needed, and **presenting code examples in the easy issues** are strategies that can possibly support newcomers overcoming the problems by increasing the awareness information provided in the issues. Members and newcomers from LibreOffice, Apache Open Office, and Noosfero reported that classification of easy issues was a practice of these projects. All of them are part of a step-by-step tutorial. These strategies need to be carefully conducted, to avoid the issues reported previously, such as setting the wrong difficulty or leaving the issue with outdated information.

Mentorship, which is a strategy well used in the industry [1, 8], was not reported as a current practice in OSS projects by any interviewee. However, both newcomers and experienced members informed that mentoring could help newcomers finding their way in the project (and also in other activities). This appears in the following two quotes:

"if the more experienced developers had more time to work together with the newcomers through pair programming or even just mentoring…in my experience it's a very fast way to transmit the knowledge" [E8]

"may be someone saying hey let's see can you fix jira-xxx. It would be way easier, right? Normally that happens in a paid job, right? Probably some basic handholding would have helped" [D2]

5.3. Solutions from the State-of-the-practice

In addition to the strategies found in the literature and identified from the interviews, we searched for practices currently applied in projects. We analyzed the webpages from six highly popular projects to check what kind of strategies is applied to support newcomers. The projects chosen in this case were Mozilla, MediaWiki, Ubuntu, GNOME, Apache Open Office, and Libre Office.

For each of these projects we accessed the how-to-contribute web page. From there, we reached the newcomers guidelines. The next step was to map strategies that could be accessed from this page to support newcomers finding their way in the project. We report the strategies provided in Table 4.

In addition to the solutions found in the project portals, it is important to mention that OpenHatch [19] provides a set of solutions to help lowering the barriers to entry into the open source community. Their approaches include browsing easy or small tasks from open source projects bug trackers.

Table 4. List of strategies identified in the projects webpages

Strategy	MediaWiki	Mozilla	Ubuntu	Gnome	Open Office	LibreOffice
Specific page (tutorial) for newcomers	•	•	•	•	•	•
List of easy tasks by difficulty	•	•		•	•	•
List of easy tasks by required skills		•				•
List of easy tasks by topic/subproject		•		•	•	•
List of mentored bugs		•				
Mentorship per subproject				•		
Form to send an email with your goals and skills requesting a task		•				

5.4. Summary

In this section we summarize the strategies reported in the previous sections, and which issues they can help addressing or solving. This information is presented in Table 5**Error! Not a valid bookmark self-reference.**. The link between strategy and issue is hypotheses that need to be empirically evaluated. As most of the strategies identified depends on the manual actions from members of project, they are subject to mistakes. However, we understand that a careful use of them can support newcomers overcoming the difficulty to find an appropriate task to start with.

Interesting facts that can be noticed from the table is that strategies used in practice are not evaluated nor mentioned in the literature; and the strategies proposed in the literature to support newcomers were not adopted for any of the projects analyzed. These facts leave room for further researches to assess why they are not adopted and the effectiveness of using these approaches in real scenarios.

6. Threats to Validity

We are aware that each project has its singularities, that the OSS universe is huge, so, the level of support and the barriers can differ according to the project or the ecosystem. Our strategy to consider different projects and different profiles of developers aimed to mitigate this limitation, identifying recurrent mentions of barriers from multiple perspectives.

Another threat to the validity of the results is the subjectivity of the data classification. We used the Grounded Theory procedures to mitigate this threat, given that the GT requires the entire analysis to be grounded in the data collected. Additionally, the analysis process was discussed along with two other researchers, to encourage a better validation of the interpretations through the mutual agreement.

During interviews, the experienced members should have answered questions pointing out to newcomers in general issues instead of those issues faced during their very first contribution. This should have occurred since project members sometimes are not fully aware of who are the people who are interacting. To avoid this kind of situation, before each interview we reinforced that we were interested in the barriers faced by newcomers when they were seeking to place their first contribution.

The practices identified in the projects were found in projects' webpages, however the projects may provide other strategies to support newcomers. Although, we explicitly asked about the strategies in the interviews, we plan to contact more core members of the projects analyzed to verify whether they have other practices to support newcomers.

As we sent open invitations to mailing list, there should be sampling bias in our interviewees and open question respondents, namely self-selection bias and social desirability bias. But, getting different sources and analyzing the answers in context to identify specificities, we tried to avoid the effect of that.

Table 5. Map of strategies identified that can offer support newcomer finding their tasks

Strategy proposed	Identified in				Can solve (hypotheses)
	Literature	Qualitative Analysis		State-of-the-practice	
		Suggestion	Existing		
Specific page (tutorial) for newcomers		•	•	•	– Documentation issues
List of easy tasks by difficulty	•	•	•	•	– Problems with information provided in the issues – Lack of confidence to choose a task
List of easy tasks by required skills				•	– Problems with information provided in the issues – Lack of confidence to choose a task
List of easy tasks by topic/subproject				•	– Problems with information provided in the issues – Lack of confidence to choose a task
List of mentored bugs		•		•	– Lack of direction on the beginning – Lack of confidence to choose a task – Newcomers don't know how to reduce the scope
Mentorship per subproject		•		•	– Lack of direction on the beginning – Lack of confidence to choose a task – Newcomers don't know how to reduce the scope
Form to send an email with your goals and skills requesting a task				•	– Lack of direction on the beginning – Lack of confidence to choose a task
Examples in the easy bugs		•			– Problems with information provided in the issues – Lack of confidence to choose a task
Recommend artifacts that can be related to the issues	•				– Problems with information provided in the issues – Lack of confidence to choose a task – Newcomers don't know where to place the changes
Recommend similar bugs and documents to support newcomers	•				– Problems with information provided in the issues – Lack of confidence to choose a task

7. Conclusions

Although this study focused on OSS communities, better supporting newcomers is an important issue in many other communities. Many virtual communities count on volunteer contributions, what makes it easy for the members to leave their groups [3] since these volunteer members generally are not bound to their groups through an employment contract [6]. Moreover, the impoverished awareness information, lack of trust, and the relatively weak interpersonal ties between members in many online groups make it more problematic to attract and retain people than in face-to-face groups [32]. Therefore, studying newcomers and the problems faced by them in virtual communities is a contemporary problem that still needs to be further investigated.

In this paper, we qualitatively analyze data collected from newcomers, dropouts, and members of OSS projects, aiming at understanding the barriers faced by newcomers to OSS projects. The results of the analysis helped us understanding the "difficulty to find a task to start with" faced by newcomers. It was possible to evidence some of the actual issues that explain this barrier. Lack of confidence to choose a task appeared as a relevant concept in our analysis. Newcomers need the project to provide enough information about the tasks or the community to support their decision about which task is more suitable for them.

We identified some already existent and suggested strategies that can support newcomers. Most part of the already applied strategies aimed to provide more information in the tasks reported on the issue tracker. By applying appropriate filters, newcomers can find a subset of tasks that can fit their goals, interests and skills, and pointing them a possible mentor. Another observation is that the strategies found in the literature were not identified in any project or interview.

In fact, by identifying possible strategies to mitigate the issues and help newcomers overcome the difficulty to find a task, we left a set of hypotheses opened. The next step of this research encompasses planning and conducting experiments to assess the effectiveness of the strategies identified in supporting newcomers onboarding to OSS projects.

8. Acknowledgements

The authors would like to thank Fundação Araucária, CNPq (process 477831/2013-3), FAPEAM (processes 062.00146/2012; 062.00600/2014; 062.00578/2014; and 01135/2011), NAPSoL-PRP-USP, NAWEB, and FAPESP for their financial support. Igor Steinmacher received grants from CAPES (BEX 2038-13-7) during the development of this work.

9. References

[1] Begel, A. and Simon, B. 2008. Novice Software Developers, All over Again. *4th International Workshop on Computing Education Research* (2008), 3–14.

[2] Bryant, S.L. et al. 2005. Becoming Wikipedian: Transformation of Participation in a Collaborative Online Encyclopedia. *Proceedings of the 2005 International ACM SIGGROUP Conference on Supporting Group Work* (Sanibel Island, Florida, USA, 2005), 1–10.

[3] Burke, M. et al. 2009. Feed Me: Motivating Newcomer Contribution in Social Network Sites. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Boston, MA, USA, 2009), 945–954.

[4] Canfora, G. et al. 2012. Who is Going to Mentor Newcomers in Open Source Projects? *Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering* (Cary, North Carolina, 2012), 44:1–44:11.

[5] Capiluppi, A. and Michlmayr, M. 2007. From the Cathedral to the Bazaar: An Empirical Study of the Lifecycle of Volunteer Community Projects. *Open Source Development, Adoption and Innovation*. J. Feller et al., eds. Springer Boston. 31–44.

[6] Choi, B. et al. 2010. Socialization Tactics in Wikipedia and Their Effects. *Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work* (Savannah, Georgia, USA, 2010), 107–116.

[7] Cubranic, D. et al. 2005. Hipikat: a project memory for software development. *IEEE Transactions on Software Engineering*. 31, 6 (2005), 446–465.

[8] Dagenais, B. et al. 2010. Moving into a new software project landscape. *32nd International Conference on Software Engineering* (2010), 275–284.

[9] Franca, A.C.C. et al. 2011. Motivation in software engineering: A systematic review update. *Evaluation Assessment in Software Engineering (EASE 2011), 15th Annual Conference on* (Apr. 2011), 154–163.

[10] Halfaker, A. et al. 2011. Don't Bite the Newbies: How Reverts Affect the Quantity and Quality of Wikipedia Work. *7th Intl. Symposium on Wikis and Open Collaboration* (2011), 163–172.

[11] Hannebauer, C. et al. 2014. An Exploratory Study of Contribution Barriers Experienced by Newcomers to Open Source Software Projects. *Proceedings of the 1st International Workshop on CrowdSourcing in Software Engineering* (Hyderabad, India, 2014), 11–14.

[12] Jensen, C. et al. 2011. Joining Free/Open Source Software Communities: An Analysis of Newbies' First Interactions on Project Mailing Lists. *System Sciences (HICSS), 2011 44th Hawaii International Conference on* (2011), 1–10.

[13] Jensen, C. and Scacchi, W. 2007. Role Migration and Advancement Processes in OSSD Projects: A Comparative Case Study. *29th Intl. Conf. on Software Engineering, 2007. ICSE 2007.* (2007), 364–374.

[14] Jergensen, C. et al. 2011. The Onion Patch: Migration in Open Source Ecosystems. *19th ACM SIGSOFT Symposium and the 13th European Conf. on Foundations of Software Engineering* (2011), 70–80.

[15] Krogh, G. von et al. 2003. Community, joining, and specialization in open source software innovation: A case study. *Research Policy*. 32, 7 (2003), 1217–1241.

[16] Malheiros, Y. et al. 2012. A Source Code Recommender System to Support Newcomers. *36th Annual Computer Software and Applications Conf. (COMPSAC)* (2012), 19–24.

[17] Meirelles, P. et al. 2010. A study of the relationships between source code metrics and attractiveness in free software projects. *2010 Brazilian Symposium on Software Engineering (SBES),* (2010), 11–20.

[18] Nakakoji, K. et al. 2002. Evolution Patterns of Open-source Software Systems and Communities. *Proceedings of the International Workshop on Principles of Software Evolution* (Orlando, Florida, 2002), 76–85.

[19] OpenHatch 2014. Community tools for free and open source software. http://openhatch.org.

[20] Park, Y. and Jensen, C. 2009. Beyond pretty pictures: Examining the benefits of code visualization for open source newcomers. *Proceedings of VISSOFT 2009 - 5th IEEE International Workshop on Visualizing Software for Understanding and Analysis* (2009), 3–10.

[21] Qureshi, I. and Fang, Y. 2011. Socialization in Open Source Software Projects: A Growth Mixture Modeling Approach. *Org. Res. Methods*. 14, 1 (2011), 208–238.

[22] Scacchi, W. 2002. Understanding the requirements for developing open source software systems. *IEE Proceedings Software*. 149, 1 (2002), 24–39.

[23] Seaman, C.B. 1999. Qualitative methods in empirical studies of software engineering. *Software Engineering, IEEE Transactions on*. 25, 4 (Jul. 1999), 557–572.

[24] Shah, S.K. 2006. Motivation, Governance, and the Viability of Hybrid Forms in Open Source Software Development. *Manage. Sci.* 52, 7 (2006), 1000–1014.

[25] Singh, P.V. 2010. The Small-world Effect: The Influence of Macro-level Properties of Developer Collaboration Networks on Open-source Project Success. *ACM Trans. Softw. Eng. Methodol.* 20, 2 (Sep. 2010), 6:1–6:27.

[26] Steinmacher, I. et al. 2014. Preliminary empirical identification of barriers faced by newcomers to Open Source Software projects. *Proceedings of the 28th Brazilian Symposium on Software Engineering (SBES 2014)* (2014), 1–10.

[27] Steinmacher, I. et al. 2015. Social Barriers Faced by Newcomers Placing Their First Contribution in Open Source Software Projects. *Proceedings of the 18th ACM conference on Computer supported cooperative work & social computing* (Vancouver, BC, Canada, Feb. 2015), 1–13.

[28] Steinmacher, I. et al. 2014. Systematic review on problems faced by newcomers to open source projects. *10th International Conference on Open Source Software* (2014), 10pp.

[29] Steinmacher, I. et al. 2014. The Hard Life of Open Source Software Project Newcomers. *International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE 2014)* (2014).

[30] Steinmacher, I. et al. 2013. Why do newcomers abandon open source software projects? *International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)* (2013), 25–32.

[31] Strauss, A. and Corbin, J.M. 1998. *Basics of Qualitative Research : Techniques and Procedures for Developing Grounded Theory*. SAGE Publications.

[32] Tidwell, L.C. and Walther, J.B. 2002. Computer-Mediated Communication Effects on Disclosure, Impressions, and Interpersonal Evaluations: Getting to Know One Another a Bit at a Time. *Human Communication Research*. 28, 3 (2002), 317–348.

[33] Tsay, J. et al. 2014. Influence of social and technical factors for evaluating contribution in GitHub. *International Conference on Software Engineering (ICSE 2014)* (2014).

[34] Vora, P. et al. 2010. The n00b Wikipedia Editing Experience. *6th Intl. Symposium on Wikis and Open Collaboration* (2010), 36:1–36:3.

[35] Wang, J. and Sarma, A. 2011. Which bug should I fix: helping new developers onboard a new project. *Proceedings of the 4th International Workshop on Cooperative and Human Aspects of Software Engineering* (Waikiki, Honolulu, HI, USA, 2011), 76–79.

[36] Wolff-Marting, V. et al. 2013. Patterns for tearing down contribution barriers to FLOSS projects. *12th Intl. Conf. on Intelligent Software Methodologies, Tools and Techniques* (2013), 9–14.

[37] Ye, Y. and Kishida, K. 2003. Toward an Understanding of the Motivation Open Source Software Developers. *Proceedings of the 25th International Conference on Software Engineering* (Portland, Oregon, 2003), 419–429.

[38] Zhou, M. and Mockus, A. 2012. What make long term contributors: Willingness and opportunity in OSS community. *Software Engineering (ICSE), 2012 34th International Conference on* (Jun. 2012), 518–528.