Noname manuscript No.
(will be inserted by the editor)

Overcoming Social Barriers when Contributing to Open Source Software Projects

Igor Steinmacher · Marco Gerosa · Tayana U. Conte · David F. Redmiles

the date of receipt and acceptance should be inserted later

Abstract Abstract An influx of newcomers is critical to the survival, long-term success, and continuity of many Open Source Software (OSS) community-based projects. However, newcomers face many barriers when making their first contribution, leading in many cases to dropouts. Due to the collaborative nature of community-based OSS projects, newcomers may be susceptible to social barriers, such as communication breakdowns and reception issues. In this article, we report a two-phase study aimed at better understanding social barriers faced by newcomers. In the first phase, we qualitatively analyzed the literature and data collected from practitioners to identify barriers that hinder newcomers' first contribution. We designed a model composed of 58 barriers, including 13 social barriers. In the second phase, based on the barriers model, we developed FLOSScoach, a portal to support newcomers making their first contribution. We evaluated the portal in a diary-based study and found that the portal guided the newcomers and reduced the need for communication. Our results provide insights for communities that want to support newcomers and lay a foundation for building better onboarding tools. The contributions of this paper include identifying and gathering empirical evidence of social barriers faced by newcomers; understanding how social barriers can be reduced or avoided by using a portal that organizes

Igor Steinmacher
School of Informatics, Computing, and Cyber Systems – Northern Arizona University (USA)
Department of Computing – Federal University of Technology, Paraná (Brazil)
E-mail: igorfs@utfpr.edu.br

Marco Aurelio Gerosa
Institute of Mathematics and Statistics – University of São Paulo (Brazil)
School of Informatics, Computing, and Cyber Systems – Northern Arizona University (USA)
E-mail: marco.gerosa@nau.edu

Tayana Uchoa Conte
Institute of Computing – Federal University of Amazonas (Brazil)
E-mail: tayana.conte@icomp.ufam.br

David F. Redmiles
University of California, Irvine (USA)
E-mail: redmiles@ics.uci.edu

proper information for newcomers (FLOSScoach); presenting guidelines for communities and newcomers on how to reduce or avoid social barriers; and identifying new streams of research.

Keywords Onboarding; difficulties; challenges; joining; socialization; open source software; FLOSS; social barriers; entry; newcomers; new contributor; newbies; beginners; qualitative study; online communities

1 Introduction

The Open Source Software (OSS) model has become an important driving force in today's software development, resulting in many prominent projects that are extensively used through the entire development stack, from kernels to sophisticated end-user applications (Steinmacher et al., 2017). Even startups and commercial projects are increasingly contributing to OSS (Anthes, 2016), as well as open-sourcing their projects (Pinto et al., 2018). It is no surprise that the OSS movement attracts a large, globally distributed community of volunteers. Developers participate in OSS because their contributions help them learn (David and Shapiro, 2008; Singh and Holt, 2013), gain visibility (Riehle, 2015; Cai and Zhu, 2016), benefit society (Parra et al., 2016), and even get jobs (Riehle, 2015; Greene and Fischer, 2016). There are hundreds of thousands of active OSS projects, comprising billions of lines of code, involving millions of software engineers, with the total amount of source code and the total number of projects doubling every 14 months (Deshpande and Riehle, 2008).

A great number of these OSS projects are classified as community-based (David and Shapiro, 2008), which are generally self-organized and dynamic, receiving contributions from volunteers spread across the globe. The survival, long-term success, and continuity of these projects, therefore, requires an influx of newcomers (Qureshi and Fang, 2011). These newcomers serve as a workforce as well as a source of innovation for ideas and work procedures (Kraut and Resnick, 2012). Although many want to volunteer, newcomers' face many difficulties entering OSS; they are akin to explorers finding their way in a hostile landscape (Dagenais et al., 2010).

Previous research related to newcomers' joining process examined the dynamics driving OSS contributors, mostly focusing on the motivations for becoming a member, roadmaps to becoming a core developer, or indicators of potential long-term commitment (Hars and Ou, 2002; Ye and Kishida, 2003; Jergensen et al., 2011; Schilling et al., 2012; Zhou and Mockus, 2012). An understudied aspect of the OSS joining process is the social hurdles that happen during the period after a newcomer decides to participate and before their first code contribution is accepted and included in the shared repository. What happens in this period may affect, for example, students in computer courses whose assignments include OSS project contribution and professional developers who find a bug or wish to customize a software product. When newcomers face barriers during this period, they can decide to give up contributing. Thus, as Fogel (2013) states, "if a project doesn't make a good first impression, newcomers may wait a long time before giving it a second chance." With a more in-depth understanding of the existing social barriers, researchers and community members can invest their efforts into better welcoming newcomers. This ultimately yields more contributions, such as casual contributions (Pham et al., 2013; Pinto et al., 2016), which are changes made by developers who are only casually or briefly interested in a project and do not intend to have a prolonged engagement.

In this paper, our goal is to understand the social barriers faced by newcomers to OSS projects and propose ways to overcome or avoid these barriers, discussing them in the light of the CSCW literature. To achieve this, we focused on the social barriers found in a two-phase study. In the first phase, we qualitatively analyzed data collected from the feedback of newcomers, a systematic literature review, a survey sent to OSS projects, and interviews with practitioners (experienced members and newcomers). To analyze the data, we used qualitative data coding procedures (Strauss and Corbin, 2007). The result of this analysis is a conceptual model comprising 58 barriers organized into 6 main categories. Among these barriers, 13 were classified as social barriers, which were similar across the different projects and are the focus of this paper. A more general view of the model is discussed elsewhere (Steinmacher et al., 2014). The social barriers we identified relate to issues that arose from the relationship between the newcomers and the established community.

In the second phase of the study, we developed FLOSScoach (Steinmacher et al., 2016), a portal to support newcomers making their first OSS project contribution, which was based on the conceptual model produced in the first phase. To assess the portal, we conducted a study with undergraduate students, relying on qualitative data from diaries. By applying the model to a practical application and assessing it, we could evaluate and improve the model, and verify how it supported newcomers.

Thus, the contributions of this paper include: (i) detailed empirical evidence, analyzed from different perspectives, of the social barriers faced by newcomers to OSS projects; (ii) findings about how a portal that organizes information for newcomers to OSS influences onboarding from the social barriers' perspective; (iii) a discussion of the social barriers (and possible mitigation strategies) in the context of CSCW and related literature; (iv) guidelines for communities that want to welcome newcomers and for newcomers willing to join an OSS project.

This paper extends previous works (Steinmacher et al., 2016, 2015b) by further detailing the analysis of social barriers in FLOSScoach, and providing both a segmented analysis of the barriers (by project characteristics and newcomers' previous experience) as well as a set of guidelines for OSS project communities and newcomers. OSS communities and researchers may leverage our findings to better understand the barriers in their context and design strategies to deal with them.

The remainder of this work is structured as follows: Section 2 discusses the literature related to this research; Section 3 presents an overview of the research method; Section 4 presents Phase I of this study, while Section 5 presents Phase II. In Section 6, we discuss our results, offer guidelines for both communities and newcomers, and provide some implications to research. Finally, in Section 7, we present some limitations of this research and, in Section 8, we offer concluding remarks.

2 Related Work

In this section, we present findings related to social barriers in open collaboration communities, with a special focus on OSS.

Perhaps due to its new contributors' decline (Halfaker et al., 2013), Wikipedia has been the subject of several studies. Zhu et al. (2013), for example, identified that received feedback influences newcomers' engagement. In addition, Halfaker et al. (2011) and Suh et al. (2009) found that newcomers are negatively impacted by a high number of edit reverts. Tsvetkova et al. (2017) analyzed bots interaction and showed that bots help, but may give rise to complex interactions with newcomers. From a more positive angle, Choi et al. (2010) found that welcome messages, technical assistance, and constructive criticism over time retarded the natural decline in newcomers' editing. Similarly, Faulkner et al. (2012) found that modifying first-time warnings prompted additional newcomer contribution. Offering a sentiment that seems to undergird each of these studies' implications, Halfaker et al. (2013) concluded that "Wikipedia has changed from the encyclopedia that anyone can edit to the encyclopedia that anyone who understands the norms, socializes himself or herself, dodges the impersonal wall of semi-automated rejection, and still wants to voluntarily contribute his or her time and energy can edit." Other studies have focused on different online communities, including Facebook (Burke et al., 2009), online health support groups (Wang et al., 2012b; Yang et al., 2017), collaborative map communities (OpenStreetMap) (Dittus et al., 2016), MovieLens (Karumur et al., 2016), and Slashdot (Lampe and Johnston, 2005).

Similar to other open collaboration communities (Forte and Lampe, 2013), community-based open source projects (David and Shapiro, 2008) work in a symbiotic way. Communities need new developers to remain sustainable, and a large, globally distributed community of developers wants to contribute for a variety of reasons (Parra et al., 2016; Riehle, 2015; Singh and Holt, 2013). In this sense, many studies focus on the joining process by defining stages and activities in the path to becoming core members or long-term contributors. For instance, the Onion Model (Nakakoji et al., 2002; Ye and Kishida, 2003) was presented as a general layered structure to organize OSS project member roles as well as the process a developer needs to follow in order to contribute. This model theorizes that OSS members have different roles, ranging from peripheral users to core members, and these roles are arranged in onion (concentric) layers. The following roles compose this model, organized from most peripheral to most central: peripheral user, user active in mailing lists, documenter, bug reporter, bug fixer, active developer, core developer, and project leader. Jergensen et al. (2011) studied whether this model still holds true in large project ecosystems, finding little evidence that individuals migrate from the edges of a project to the core through a gradual socialization process. Herraiz et al. (2006) found that more than half of developers commit before sending a message to the mailing list, thereby contradicting the onion model. They also found two groups with clearly different joining patterns: volunteer and hired developers. Whereas volunteers tend to follow a joining process, hired developers usually experience a quick integration. In addition to the Onion Model perspectives, other studies mapping the OSS project joining process include a work from von Krogh and von Hippel (2003), who proposed the concept of a joining script; and from Ducheneaut (2005), who similarly analyzed mailing list archives, offering an in-depth look at a successful newcomer's socialization history.

Other parts of the literature focus on the forces of motivation and attractiveness that drive newcomers toward projects. Lakhani and Wolf (2005), for example, found that extrinsic benefits (e.g.; better jobs, career advancement) primarily motivate new contributors, together with enjoyment, challenges derived from writing code, and improved programming skills. Hars and Ou (2002) reported that internal motivation plays a role, but noted that external factors, such as building human capital and personal software solution needs, are more influential. Shah (2006) distinguished between two different contributors: need-driven and hobbyists. More recently, Hannebauer and Gruhn (2016) showed that personal needs (a.k.a. scratching one's own itches) is the main motivational force that drives newcomers to contribute to OSS projects. This is also reported by Pinto et al. (2016) when studying the motives that drive developers to place single contributions to OSS projects. Several other studies dealt with motivation in OSS (Bonaccorsi and Rossi, 2004; Roberts et al., 2006; Jergensen, 2007; Oreg and Nov, 2008; David and Shapiro, 2008; Ke and Zhang, 2010; von Krogh et al., 2012).

Regarding newcomers' retention and long-term contribution, Qureshi and Fang (2011) quantitatively identified four distinct classes of newcomer retention behavior, considering their initial amount of interactions with core members and the growth of these interactions. Fang and Neufeld (2009) qualitatively revealed that initial participation conditions do not effectively predict long-term participation, but that situated learning and identity-construction behaviors were positively linked to sustained participation. Other research revealed that retention is influenced by an individual's familiarity with the project's coordination practices (Schilling et al., 2012) and attitude (Zhou and Mockus, 2012).

WWhile the current literature focuses on motivation and forces that lead developers to the project's core, studies neglect both those newcomers who do not envision a long-term engagement as well as those who want to place a single contribution. Counterexamples include Hannebauer et al. (2014) and Steinmacher et al. (2015b), which explicitly focus on barriers that influence newcomers' first contributions. Some other studies proposed ways to facilitate newcomers' first contributions. Wolff-Marting et al. (2013), for example, proposed two patterns to support newcomers in overcoming contribution barriers. The first pattern (Pre-configured Build Environment) aims to help newcomers set up their local environments and compile the project's source code; the second (Unit Tests for Contributors) aims to encourage newcomers to submit their patches and get them merged into the main branch of the project. Also aiming to support newcomers, Cubranic et al. (2005) presented Hipikat, a tool that builds a group memory comprising source code, email discussions, and bug trackers. The tool enables newcomers to request recommendations based on existent artifacts. With a similar goal, Wang and Sarma (2011) presented a tool to enable newcomers to identify bugs of interest and resources related to that bug, as well as to visually and interactively explore the bug's appropriate socio-technical dependencies. Canfora et al. (2012) proposed and evaluated an approach to identifying and recommending mentors for open source project newcomers by mining data from mailing lists and source code versioning systems. Closer to this work, Balali et al. (2018) analyzed the barriers faced by newcomers and mentors from the mentors' perspective, finding that social aspects are challenging during the mentorship process.

As opposed to the traditional focus on motivation, attractiveness, and retention in OSS projects, our study focuses specifically on the initial contribution barriers. Other studies cite barriers that influence newcomers' overall experiences, as can be observed in a previous literature review (Steinmacher et al., 2015a); however, they do not provide an in-depth understanding of the barriers, their relations, or their relevance in multiple projects. Thus, there is little exploration of the barriers faced by newcomers who ultimately stop contributing or do not become members.

3 Method Overview

Our research aimed to understand the social barriers that hinder newcomers from posting their first contribution to an OSS project. We used data collected from multiple sources and different empirical methods to achieve our goal. The research design comprises two phases, as presented in Figure 1. We briefly describe each phase in the following subsections and present details for each phase in the corresponding section (Section 4 for Phase I and Section 5 for Phase II).

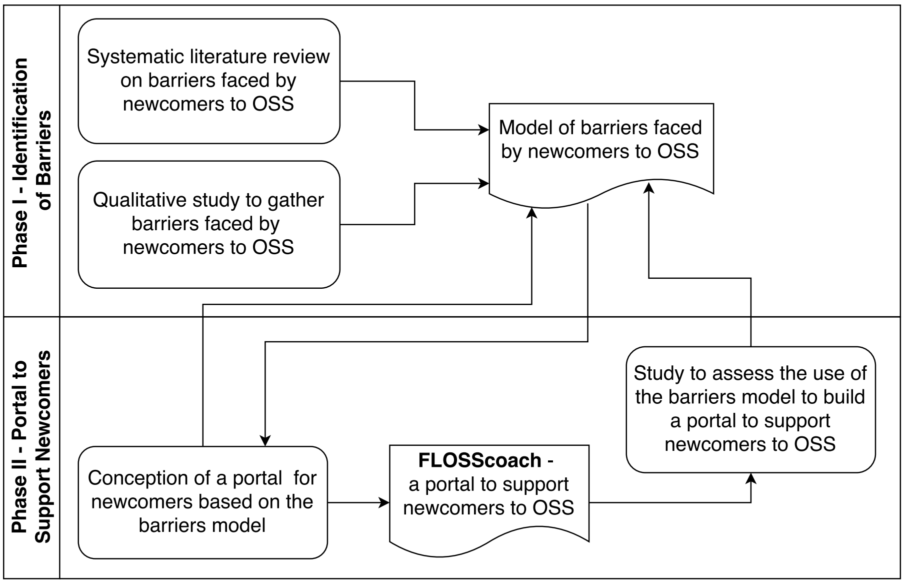

Phase I - Identification of Barriers
Systematic literature review on barriers faced by newcomers to OSS
Qualitative study to gather barriers faced by newcomers to OSS
Model of barriers faced by newcomers to OSS
Phase II - Portal to Support Newcomers
Study to assess the use of the barriers model to build a portal to support newcomers to OSS
Conception of a portal for newcomers based on the barriers model
FLOSScoach - a portal to support newcomers to OSS

Fig. 1 Research method overview

Phase I comprises studies aimed to identify barriers faced by newcomers when making their first contributions to an OSS project. This phase relied on feedback from students, a systematic literature review, answers to an open question, and interviews with both newcomers and experienced OSS project members. Three models emerged from these studies, and we compiled these models to generate a unified model of barriers for OSS newcomers. This model was used as an input for the following phase.

Phase II comprises building a portal based on the barriers model and evaluating it focusing on the social barriers. We built the portal using information and strategies gathered from practitioners, organizing them according to the barriers model. We conducted a study relying on diaries written by students during their attempt to contribute. The results enabled us to improve the barriers model and the portal according to the feedback received, as presented in Section 5.1.2.

4 Phase I – Identification of Barriers

To identify and understand the barriers faced by OSS project newcomers, we conducted a qualitative study relying on: (i) feedback from 9 students after they contributed to OSS projects; (ii) 21 studies gathered via a systematic literature review (SLR) that aimed at identifying and organizing the barriers evidenced by the literature; (iii) 24 answers to an open question sent to 9 OSS projects; and (iv) semi-structured interviews with 35 developers from 13 different projects, including newcomers, dropouts, and experienced members. We chose a qualitative approach because the onboarding process occurs in a complex, social environment, in which the context of its occurrence is important (Dittrich, 2014). To analyze the data, we used coding procedures (Strauss and Corbin, 2007), which are increasingly used to study human aspects of Software Engineering (Hoda et al., 2010; Dagenais et al., 2010; Smolander et al., 2008; Treude and Storey, 2010; Pham et al., 2013).

Figure 2 depicts the method. The first dataset was collected from 9 students who contributed to OSS projects. From their feedback, we noticed that there were recurrently reported barriers. Motivated by the students' reports, we conducted a systematic literature review to identify the available research related to these barriers (Steinmacher et al. 2015b). To gather more understanding about the barriers, and to check how the practice was related to the literature, we surveyed OSS developers. We analyzed the data from these practitioners, alongside the data from the students, to create a preliminary barriers model. The results of this preliminary model were used as input for an in-depth investigation. This investigation was made by means of semi-structured interviews conducted with both experienced members and newcomers in order to identify and explore the barriers from different perspectives.

During our study, we interacted with and received inputs from members and newcomers from 19 different OSS projects. We provide some details about these projects in Table 1. We also present the number of subjects who contributed to each project (3 questionnaire answerers did not report the project they contributed to). In the following sections, we detail the methods used to conduct the systematic review and to collect the data from practitioners, followed by details about the data analysis. As mentioned earlier, student feedback was collected before the SLR. However, for clarity purposes, we present the analysis of the former together with the data from other practitioners (Section 4.2).

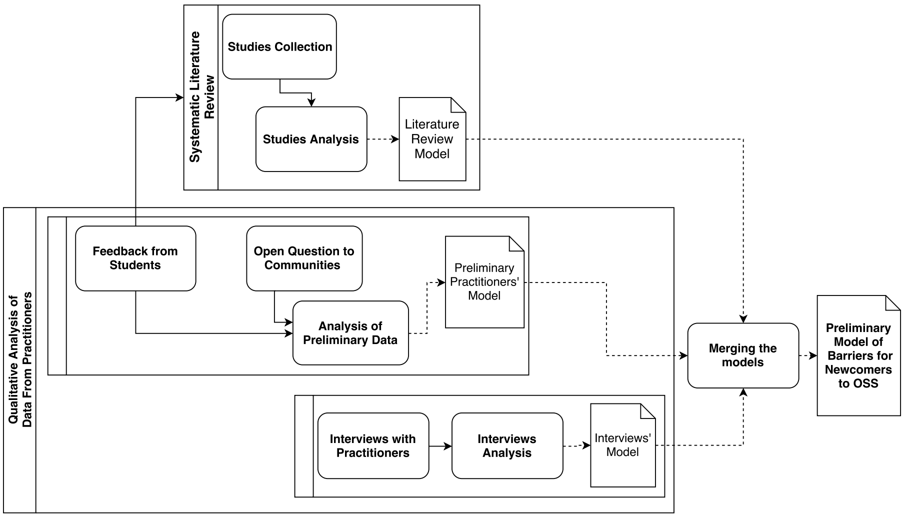

Systematic Literature Review
Studies Collection
Studies Analysis
Literature Review Model
Qualitative Analysis of Data From Practitioners
Feedback from Students
Open Question to Communities
Analysis of Preliminary Data
Preliminary Practitioners' Model
Interviews with Practitioners
Interviews Analysis
Interviews' Model
Merging the models
Preliminary Model of Barriers for Newcomers to OSS

Fig. 2 Research method followed to identify the barriers faced by OSS newcomers

Table 1 Details of projects that were part of the study (data collected in Feb. 2015, at openhub.net)

Project	Lines of Code (LoC)	Programming language	Age (years)	Hosted by	Students	Answers question.	Inter-viewees
OpenOffice	11M	C++	14	Apache		3	5
aTunes	110K	Java	8	SourceForge		3	
Audacity	273K	C++	13	Github		2	
CoGrOO	124K	Java	8	Github			3
Etherpad	42K	JavaScript	3	Github			3
GNU Emacs	1,5M	Emacs Lisp	30	Savannah		1	
FreePlane	126K	Java	4	SourceForge			
Gephi	174K	Java	6	Github			2
Integrade	190K	Java	11	Launchpad			1
JabRef	105K	Java	12	SourceForge	3		6
jEdit	300K	Java	15	SourceForge		1	2
LibreOffice	7M	C++	14	Own host	3	6	6
Moodle	2,5M	PHP	13	Own host			2
Firefox	12M	C/C++/Javascript	12	Own host	3	3	
Noosfero	463K	Ruby/JS	7	Gitlab			2
OpenVPN	321K	C	9	SourceForge		1	
Pardus	3,5M	Python	9	Own host			1
TextMate	94K	C++/Ruby	6	Github			1
ZXing	51K	Java	7	Github			1

4.1 Systematic Literature Review

We conducted a systematic literature review (Steinmacher et al., 2015a) to identify the barriers reported by the literature. The following question guided our SLR: *What are the barriers that influence newcomers' onboarding to OSS projects?*

After some iterative refining, the following query was used to retrieve the studies from the ACM, IEEE, Scopus, and Springer Link digital libraries. These libraries were selected because they index relevant venues for this study, support searches using Boolean expression, and provide access to complete texts. We considered for selection papers that were available for download, written in English, dealt with OSS newcomers' onboarding, presented empirical results, and were published in peer-reviewed journals or event proceedings. We excluded studies that analyzed only newcomers' motivation for joining a project.

("OSS" OR "Open Source" OR "Free Software" OR FLOSS OR FOSS) AND (newcomer OR "joining process" OR newbie OR "new developer" OR "new member" OR "new contributor" OR novice OR beginner OR "potential participant" OR retention OR joiner OR onboarding OR "new committer")

For each selected paper, we conducted snowball sampling (Jalali and Wohlin, 2012), checking if the authors of the selected studies published other relevant studies not retrieved from the digital libraries. We analyzed their profiles in ACM, IEEE, DBLP, and personal homepages (when available).

After executing the query on the digital libraries systems, our resulting sample comprised 291 candidate papers. For each paper, two independent researchers analyzed title, abstract, and keywords. In a consensus meeting, we agreed on 33 papers. We checked other papers published by the authors of these 33 candidate studies, finding 20 other candidate papers. After analyzing these papers, we selected nine additional papers, resulting in 42 candidate papers. After further analysis, 21 papers were deemed relevant and were considered for data extraction. Next, we classified the barriers reported in the selected studies.

4.2 Data from practitioners

For the qualitative practitioner study, we gathered data from: (i) students that contributed to OSS projects; (ii) answers to an open question sent to OSS projects; and (iii) semi-structured interviews with newcomers to and members of OSS projects.

The first source comprised feedback received from four PhD candidates and five undergraduate students who contributed to OSS projects as part of a course assignment. The students contributed to JabRef (2 graduate/1 undergraduate), LibreOffice (3 undergraduate), and Mozilla Firefox (3 graduate) projects. All the students were newcomers to the projects they were contributing to. The PhD candidates were all males, experienced developers, and 30 years old or older. The undergraduate students, including four males and one female ranging between 21 and 24 years old, were attending the last semester of the Internet Systems course and therefore were preparing to join the software development industry. After the conclusion of the assignment, their feedback was collected by means of an open-ended questionnaire. The open questions enabled students to debrief and explain the barriers they faced while trying to place their first code contribution. The data was collected at the end of the course.

The second data source comprised answers to a questionnaire sent to OSS project developers via developer mailing lists, which comprised two questions designed to profile the contributor (project and contribution time) and an open question: "*In your opinion, what are the main difficulties faced by newcomers when they want to start contributing to this project? (Consider technical and non-technical issues).*" We received 24 complete answers, as detailed in Tables 2 and 3. The questionnaire was posted and the answers received during October 2013. We sent messages to a convenience sample of nine projects from different domains: Apache OpenOffice, aTunes, Audacity, LibreOffice, emacs, FreePlane, jEdit, Mozilla Firefox, and OpenVPN. None of these projects delivers development frameworks or scaffolding technologies, which are generally more complex, demanding a higher degree of specific skills and knowledge. These characteristics could hide some possible barriers encountered by newcomers, since they could face complex problems related to the inherent project technology and domain. Moreover, we focused on projects written in C/C++ and Java, which are the main languages used in the undergraduate courses in which we conducted the Phase II of this research.

Table 2 Project to which participants mainly contribute

Project	Count
LibreOffice	6
OpenOffice	3
aTunes	3
Mozilla Firefox	3
Audacity	2
jEdit	1
OpenVPN	1
FreePlane	1
Emacs	1
Did not inform	3

Table 3 Period of contribution for questionnaire respondents

Time contributing to the project	Count
Less than 6 months	7
Between 6 months and 1 year	3
Between 1 year and 3 years	6
More than 3 years	8

The final data collection entailed semi-structured interviews with practitioners. Semi-structured interviews include a mixture of open-ended and specific questions, which are designed to elicit foreseen and unexpected information types (Seaman, 1999). We conducted interviews in order to complement the findings gathered from sources 1 and 2, thereby deepening and broadening our understanding of the newcomers' barriers. We recruited subjects belonging to four different groups: *Experienced Members* (project owners, managers, or developers that commit code directly to the software repository), *Successful 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), and *Onboarding Newcomers* (volunteers attempting to place their first contribution). Table 4 shows some profile information about the students and interviewees. The participants received an ID, shown in the first column. The first character of the ID represents the profile of the participant: “S” for students, “E” for Experienced members, “N” for Successful newcomers, “D” for Dropout newcomers, and “O” for Onboarding newcomers.

We interviewed 35 participants from 13 different projects (Pardus, TextMate, ZXing, Gephi, jEdit, Moodle, Integrade, Noosfero, Apache OpenOffice, CoGrOO, Etherpad, JabRef, and LibreOffice), including 12 experienced members, 16 newcomers that succeeded, 4 dropout newcomers, and 3 newcomers that were still attempting to place their first contributions. The participants were recruited primarily through mailing list and forums from 15 different projects. In addition, we also directly invited newcomers, identifying them by mining and following projects’ mailing lists and issue trackers. Only adults older than 18 were eligible to participate in this study; but we made no distinction related to gender, nationality, or other personal characteristics. However, participants were expected to have software development experience, primarily because we were interested in the barriers to contributing to a project, not to learning how to program.

We used a semi-structured format, in which a script (interview guide) supported the interviewing process. We started with pilot interviews with five developers involved in OSS to adjust the script. After that, we recruited the subjects and conducted the interviews. All interviews were conducted using text-based chat tools, like Google Talk, because the participants used this means of communication in their work and it facilitates data collection and interview scheduling.

4.2.1 Data Analysis

To analyze the data, we selectively applied open coding, whereby concepts are identified and their properties and dimensions are discovered, and axial coding, whereby connections among codes are identified and grouped according to their properties to represent categories (Strauss and Corbin, 2007).

We split our analysis into two steps. The first (preliminary) step consisted of analyzing the data obtained from the students’ feedback and the open questions sent to communities. In the second step, the codes and categories found in the preliminary study were used as seeds for the interview coding. During open coding, we assigned codes to sentences, paragraphs, or revisions. This procedure overlapped the axial coding, in which we identified connections between categories. We executed open and axial coding several times to refine the emerging codes and categories.

In the first step, open coding was conducted in parallel by three researchers. Each researcher independently quoted and coded the documents. After coding, the researchers discussed the quotes and codes until they came to a consensus for the entire document set. This was done to mitigate the bias caused by a single researcher and to reach a common understanding about the nomenclature and criteria. After the discussion, we began axial coding iterations, followed by discussions and changes in codes and categories.

For the second step, we analyzed the data obtained from the interviews. The analysis process was similar to the one applied in the first step. This time, a single researcher conducted the coding procedures, discussed any questions with the other two, and proposed new or merged categories.

Table 4 Profile of the participants (H = Hours per week in OSS; F = First Project?; Y = Years in the project)

ID	H	F	Project	Country	Y
E1	<5	N	JabRef	France	8
E2	05-10	N	Etherpad	Germany	3
E3	10-20	N	JabRef	Germany	3
E4	05-10	N	jEdit	Canada	10
E5	05-10	N	LibreOffice	Germany	15
E6	>20	N	LibreOffice	Hungary	10
E7	>20	N	Moodle	Australia	5
E8	>20	N	Noosfero	Brazil	5
E9	>20	N	Pardus	Turkey	8
E10	05-10		CoGrOO	Brazil	5
E11	<5	N	Noosfero	Brazil	7
E12	05-10	N	OpenOffice	Mexico	8
N1	<5	Y	JabRef	Germany	0
N2	<5	Y	Gephi	Brazil	0
N3	05-10	Y	Gephi	India	0
N4	05-10	Y	Moodle	India	0
N5	<5	Y	JabRef	Germany	0
N6	<5	Y	jEdit	United States	0
N7	<5	N	TextMate	United States	0
N8	>20	Y	ZXing	Greece	0
N9	<5	Y	CoGrOO	Brazil	0
N10	<5	Y	Integrade	Brazil	0
N11	<5	Y	CoGrOO	Brazil	0
N12	N/I	N	Etherpad	United Kingdom	0
N13	10-20	N	LibreOffice	Brazil	0
N14	05-10	Y	LibreOffice	Brazil	0
N15	N/I	Y	Etherpad	France	0
N16	05-10	N	JabRef	Germany	0
D1	<5	N	JabRef	Germany	0
D2	<5	Y	OpenOffice	Brazil	0
D3	<5	Y	LibreOffice	India	0
D4	<5	Y	OpenOffice	India	0
O1	<5	N	OpenOffice	India	0
O2	10-20	Y	LibreOffice	China	0
O3	<5	Y	OpenOffice	Greece	0
S1	N/I	N	Mozilla	Brazil	0
S2	N/I	Y	LibreOffice	Brazil	0
S3	N/I	Y	LibreOffice	Brazil	0
S4	N/I	Y	Firefox	Brazil	0
S5	N/I	Y	JabRef	Brazil	0
S6	N/I	Y	Firefox	Brazil	0
S7	N/I	N	JabRef	Brazil	0
S8	N/I	N	JabRef	Brazil	0
S9	N/I	Y	LibreOffice	Brazil	0

After obtaining the model from the interview analysis, we iteratively reanalyzed the models obtained from all studies, relying on their respective data. The goal of this reanalysis was to combine the findings to create a single model accommodating all the barriers evidenced. Once again, we merged some barriers and reorganized the categories.

4.3 Resulting Barriers Model

The resulting model, presented in Figure 3, aggregates 58 barriers organized in 6 categories and several subcategories.

After conceiving the model, we delved into social barriers. To classify the social barriers, we defined them as those that involve *or directly influence human social interactions and are related to the relationship between community and newcomers*. Three researchers discussed and came to a consensus on the social barriers. These barriers are highlighted in Figure 3 and isolated in Figure 4.

In the remainder of this section, we detail the 13 social barriers. For each barrier, we present empirical evidence and discuss how our findings relate to the broader CSCW literature (including other open collaboration communities, such as Wikipedia). Whenever we discuss a social barrier, we present the number of participants (or studies) that identified the barrier per data source. The numbers represent, in order:

- *SLR:* number of studies from the SLR that evidenced the barrier (out of 21);
- *stu:* number of students that reported the barrier in their feedback (out of 9);
- *OQ:* number of mentions in the open questions (out of 24);
- *int:* number of interviewees that reported the barrier (out of 35); and
- the number of *projects* in which the barriers were evidenced, considering only the data from practitioners (out of 19).

4.3.1 Reception Issues

We identified four barriers related to reception issues, as presented in Table 5. This table presents the sources in which the barrier was evidenced and the number of times it was reported per source and per profile. Reception issues were evidenced in all sources and reported by both newcomers and experienced members.

Table 5 Evidence of reception issues per source

Barriers	**Literature**	**Students Feedback**	**Open Question**			**Interviews**		
			< 6 mos.	**6 mos. to 3 yrs.**	**> 3 yrs.**	**Drop-outs**	**New-comers**	**Experi-enced**
Not receiving an answer	• (5)	• (1)						• (1)
Delayed answers	• (2)	• (1)					• (1)	• (2)
Impolite answers	• (2)	• (1)					• (1)	• (2)
Receiving answers with too advanced/ complex contents			• (1)				• (2)	• (1)

Not receiving an answer (*SLR: 5; stu: 1; OQ: 0; int: 1; 1 project*) was found to be a problem. From the students' feedback, we observed that their forum post was never answered and they ended up working on an incorrect issue: "They never answered our forum post. We spent a lot of effort on something that was already being done..." [S5]. In the interviews, an experienced member highlighted: "In my opinion, the first [barrier] is not getting any reply" [E1]. In our SLR, we also identified studies focusing on the reception barrier (von Krogh and von Hippel, 2003; Singh, 2012; Steinmacher

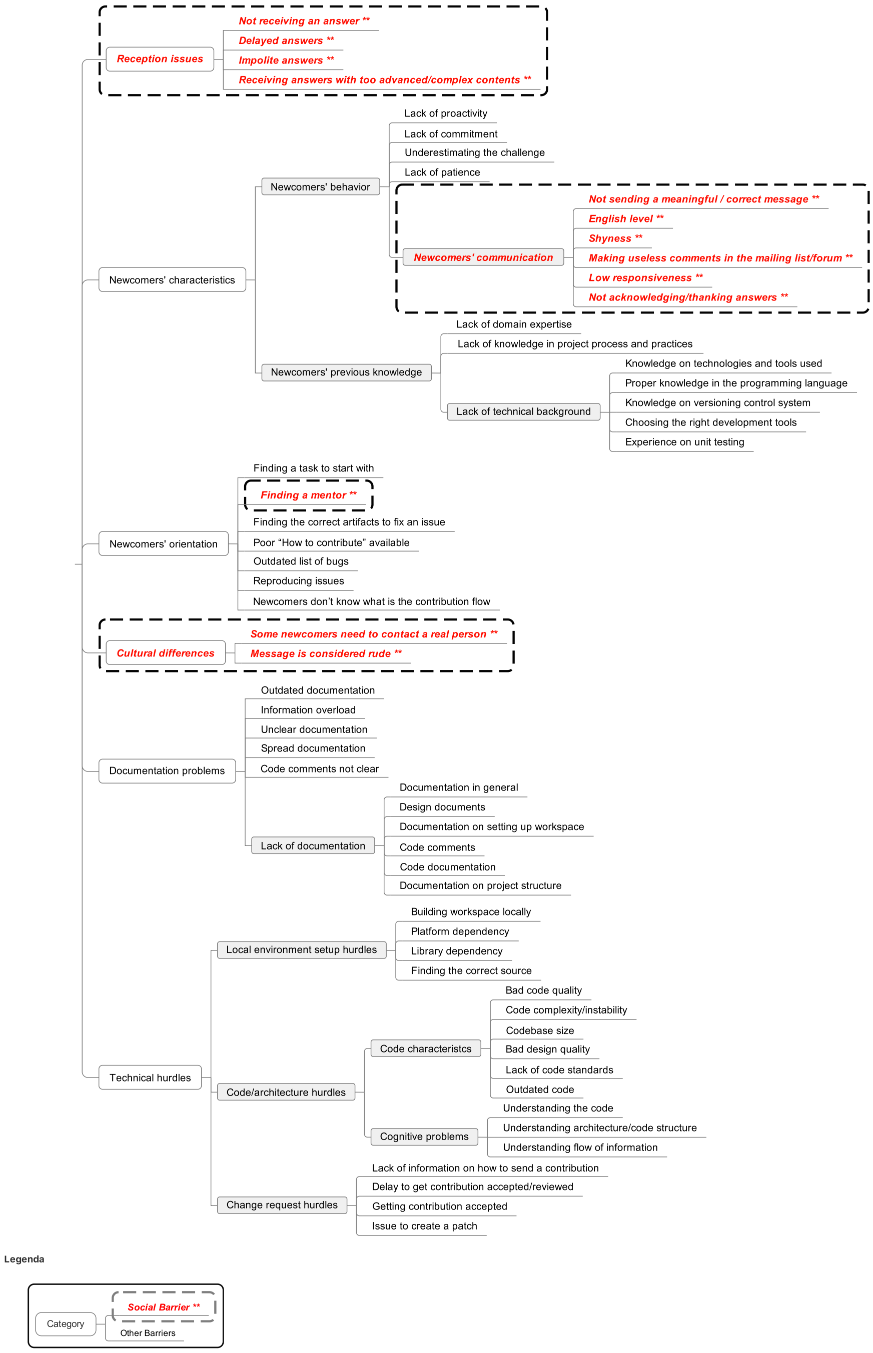

Reception issues
Not receiving an answer **
Delayed answers **
Impolite answers **
Receiving answers with too advanced/complex contents **
Newcomers' characteristics
Newcomers' behavior
Lack of proactivity
Lack of commitment
Underestimating the challenge
Lack of patience
Newcomers' communication
Not sending a meaningful / correct message **
English level **
Shyness **
Making useless comments in the mailing list/forum **
Low responsiveness **
Not acknowledging/thanking answers **
Newcomers' previous knowledge
Lack of domain expertise
Lack of knowledge in project process and practices
Lack of technical background
Knowledge on technologies and tools used
Proper knowledge in the programming language
Knowledge on versioning control system
Choosing the right development tools
Experience on unit testing
Newcomers' orientation
Finding a task to start with
Finding a mentor **
Finding the correct artifacts to fix an issue
Poor "How to contribute" available
Outdated list of bugs
Reproducing issues
Newcomers don't know what is the contribution flow
Cultural differences
Some newcomers need to contact a real person **
Message is considered rude **
Documentation problems
Outdated documentation
Information overload
Unclear documentation
Spread documentation
Code comments not clear
Lack of documentation
Documentation in general
Design documents
Documentation on setting up workspace
Code comments
Code documentation
Documentation on project structure
Technical hurdles
Local environment setup hurdles
Building workspace locally
Platform dependency
Library dependency
Finding the correct source
Code/architecture hurdles
Code characteristcs
Bad code quality
Code complexity/instability
Codebase size
Bad design quality
Lack of code standards
Outdated code
Cognitive problems
Understanding the code
Understanding architecture/code structure
Understanding flow of information
Change request hurdles
Lack of information on how to send a contribution
Delay to get contribution accepted/reviewed
Getting contribution accepted
Issue to create a patch
Legenda
Category
Social Barrier **
Other Barriers

Fig. 3 Model of barriers for newcomers to OSS

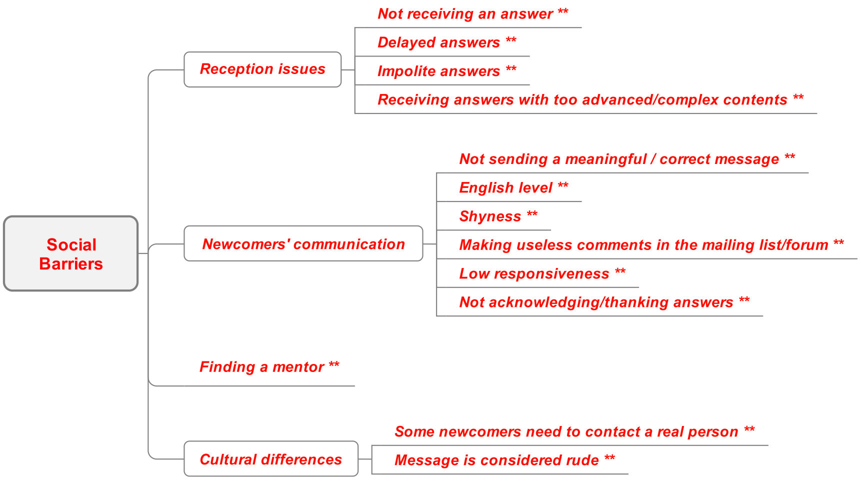

Social Barriers
Reception issues
Not receiving an answer **
Delayed answers **
Impolite answers **
Receiving answers with too advanced/complex contents **
Newcomers' communication
Not sending a meaningful / correct message **
English level **
Shyness **
Making useless comments in the mailing list/forum **
Low responsiveness **
Not acknowledging/thanking answers **
Finding a mentor **
Cultural differences
Some newcomers need to contact a real person **
Message is considered rude **

Fig. 4 A closer look at the social barriers

et al., 2013b; Stol et al., 2010; Jensen et al., 2011). Analyzing the Freenet project, von Krogh and von Hippel (2003) found that "…29 (10.5%) participants did not receive any reply to their initial posting and subsequently did not appear on the developer list again." Singh (2012) reported a similar behavior: "non-returning newcomers can be attributed to not receiving a response…" CSCW research similarly studies this particular barrier, and the results reported are in accordance with those in the OSS literature and evidenced in the qualitative analysis. Joyce and Kraut (2006) analyzed newcomers' posts to Usenet and found that newcomers who received a reply to their first posts were 12% more likely to post to the community again. Analyzing newcomers to Slashdot, Lampe and Johnston (2005) evidenced similar results.

Compounding the lack of answers, we also found delayed answers as a barrier (*SLR: 2, stu: 1; OQ: 0; int: 3; 3 projects*): "[a problem was] a huge delay to receive an answer. It was necessary to send more than one email to receive an answer after a week. Demotivating. I was about to give up" [S6]. It also bothered a newcomer to the ZXing project: "The biggest 'bottleneck' would have probably been the slow pace in communication…if you have a deadline a few days every now and then it can be quite bothersome" [N8]. In the SLR, we identified evidence of this barrier in two studies (Jensen et al., 2011; Zhou and Mockus, 2012). Jensen et al. (2011) analyzed the mailing lists of four OSS projects and found that "None of the newbies who failed to receive a reply within 24 hours of posting their first question were still posting to the mailing lists beyond the study period." However, we could not find any specific evidence of delayed or late reply in the broader CSCW literature analyzed for this study. Additional research is necessary to verify whether this is an issue in other domains.

Impolite answers (*SLR: 2; stu: 1; OQ: 0; int: 3; 4 projects*) also appeared in the students' feedback and in the interviews. For example, an experienced member reported: "…and of course, one more thing is the developers' attitude. Some developers may not be suitable for receiving newcomers, they may get angry pretty quickly and kill the interest of the newcomers. Very few of the newcomers know how to behave against this kind of tough developer" [E9]. One of the OSS studies analyzed in our SLR (Singh, 2012) reported that non-returning newcomers can be attributed to receiving a condescending response. In the context of Wikipedia, Farzan and Kraut (2013) reported that newcomers become demotivated due to experienced editors' hostile behavior, such as reversions and deletions without proper/polite explanation. Zhu et al. (2012), Suh et al. (2009), and Halfaker et al. (2011) also found that negative feedback reduced motivation in Wikipedia.

Receiving answers with too advanced/complex contents (*SLR: 0; stu: 0; OQ: 1, int: 3; 3 projects*) was another barrier evidenced in our data obtained from practitioners. In some cases, newcomers received answers that required in-depth knowledge about the project and technologies. For example, a newcomer reported: "The reason I didn't find the reply helpful is due to they talk a little bit out of my understanding of the project" [O2]. Another newcomer reported a similar problem: "I found it awesome to get a quick and nice reaction, but the suggestions I could do seemed a bit farfetched to give to a beginner" [N8]. We could not find any literature reporting or supporting this specific finding.

From the analysis of these 4 barriers, we noticed that reception could result in a smaller number of returning newcomers. Open collaboration communities, therefore, ought to be attentive to newcomer reception. Indeed, welcome messages, assistance, and constructive criticism increase the retention of newcomers (Choi et al., 2010). The literature on open collaboration communities presents automated strategies that can be used to soften reception issues (Faulkner et al., 2012; Halfaker et al., 2013), which are mainly focused on automated answer/feedback to newcomers' contributions, including the usage of bots and its perils (Tsvetkova et al., 2017). We are not aware of any approach like this in OSS communities, but this strategy's applicability and effectiveness should be explored and assessed.

4.3.2 Newcomers' Communication Behavior

Newcomers' characteristics may also result in barriers. Newcomers are expected to possess a minimum requirement of previous technical background to perform a development task. In addition, the community also expects newcomers to have certain social skills. We could evidence 6 barriers under this category, as presented in Table 6. Notably, only experienced members reported these barriers, whereas the newcomers did not mention their own communication behavior as possible problems.

Three experienced members reported not sending a correct / meaningful message (*SLR: 2; stu: 0; OQ: 0; int: 3; 2 projects*) as a barrier. Community members may not answer a message if they do not understand it: "in general I answer the questions that are well written … some people post things that… I don't know how to answer. So, I wait until someone else makes an attempt and see if the original poster will make a better effort the second time to post something that I can understand" [E4]. As per our SLR, Singh (2012) studied this problem in OSS forums. She demonstrated that the community responds better to informative subject lines, comprehensible posts, and

Table 6 Evidence of newcomers' communication behavior per source

Barriers	**Literature**	**Students Feedback**	**Open Question**			**Interviews**		
			< **6 mos.**	**6 mos. to 3 yrs.**	> **3 yrs.**	**Drop-outs**	**New-comers**	**Experi-enced**
Not sending a meaningful/ correct message	• (2)							• (3)
English level								• (4)
Shyness								• (2)
Making useless comments in the mailing list/forum								• (1)
Low responsiveness								• (1)
Not acknowledging thanking answers received								• (1)

correct messages, which is also studied in other CSCW domains. Burke et al. (2007), for example, analyzed Usenet communities, finding that self-introductions can double the odds of receiving answers. Arguello et al. (2006) also analyzed Q&A history and found that on-topic messages and use of vernacular language increased reply likelihood. Joyce and Kraut (2006) also analyzed Usenet communities and reported that newcomers were more likely to receive a response if they asked a question or wrote a longer post.

A more specific and related barrier regards level of English comprehension and use (*SLR: 0; stu: 0; OQ: 0; int: 4; 3 projects*). English is adopted in most OSS projects. After reporting problems with incorrect messages, E4 amended: "for some people it is due to English as a second language, I understand that but still..." Another participant reinforced this issue: "Having a decent English is needed" [E1]. This is difficult to address, but providing guidelines or asking for clarification may help. For larger communities, matching a newcomer to a member that speaks the same language might be helpful.

Shyness (*SLR: 0; stu: 0; OQ: 0; int:2; 2 projects*) was also reported as a barrier. An experienced member reported that once he gave up contributing because he felt timid about asking the community: "I was trying to solve a bug... by myself. I was kind of shy to ask for help" [E11]. Preece et al. (2004) analyzed the MSN bulletin board and found that 28.3% of the lurkers gave shyness as a reason for not posting. A possible approach to deal with this issue would be *breaking the ice*. As soon as newcomers subscribe, a member could approach them; automatic greetings could also be used. For this category of barriers, we found that CSCW literature mainly studies and provides evidence on how, in order to be well received, newcomers should behave when sending their first messages (Preece, 2004; Arguello et al., 2006; Joyce and Kraut, 2006; Burke et al., 2007). The evidence found relies on studies conducted with historical data of Q&A communities. For the other barriers, we could not find any evidence in the literature.

4.3.3 Finding a mentor

In accordance with Dagenais et al. (2010), who analyzed closed-source software projects, we found that newcomers often face unfamiliar and rugged landscapes when starting to contribute to an OSS project. Consequently, they need proper orientation to find their way into the project and contribute correctly.

Table 7 Evidence of difficulty finding a mentor

Barriers	**Literature**	**Students Feedback**	**Open Question**			**Interviews**		
			< **6 mos.**	**6 mos. to 3 yrs.**	> **3 yrs.**	**Dropouts**	**Newcomers**	**Experienced**
Difficulty Finding a Mentor	• (3)	• (3)	• (2)			• (1)	• (2)	• (2)

Difficulty finding a mentor (*SLR: 3; stu: 3; OQ: 2; int: 5; 7 projects*) was identified as a social barrier for newcomers to OSS. This was evidenced in all the sources we examined, and it was reported both by newcomers and experienced members, as presented in Table 7. One newcomer reported: "…basically, see, I was not an active contributor at that time…if some meaningful direction could be provided then I would have started. This direction I didn't get…someone with my profile…who want to do some stuff with open source project probably some basic handholding would have helped." [D2]. This barrier was also evidenced in the systematic review. Cubranic et al. (2005) reported, "It can be difficult for newcomers to join such groups [OSS projects] because it is hard to obtain effective mentoring." To alleviate this problem, Canfora et al. (2012) proposed a tool that recommends mentors to newcomers. They evaluated the tool by surveying project members, finding that mentoring is important to newcomers. Some studies exist on mentorship in open collaboration communities. For instance, Musicant et al. (2011) qualitatively analyzed data from Wikipedia's program "Adopt-a-user"[1] and found that several key mentor functions are missing or inconsistently fulfilled. Most adopters focus on establishing their legitimacy rather than proactively guiding, protecting, and supporting the long-term growth of adoptees. Choi et al. (2010) analyzed Wikipedia socialization tactics and found that they rarely assign new members a mentor or provide clear guidance about how to behave in a project. Discovering what motivates experienced members to properly mentor newcomers is therefore likely a fruitful future research area.

4.3.4 Cultural Differences

Differences related to individuals' cultural backgrounds are a known problem in distributed software development (Steinmacher et al., 2013b), including OSS development (Herraiz et al., 2006). Volunteers have diverse national, organizational, and professional backgrounds, resulting in different values, perceptions, and work behaviors. According to Herbsleb and Moitra (2001), this can lead to serious misunderstandings

[1] `https://en.wikipedia.org/wiki/Wikipedia:Adopt-a-user`

and conflicts. In our study, barriers from this category appeared only during the interviews, as reported by two experienced members and one newcomer (Table 8).

Table 8 Evidence of cultural differences per source

Barriers	**Literature**	**Students Feedback**	**Open Question**			**Interviews**		
			< 6 mos.	**6 mos. to 3 yrs.**	**> 3 yrs.**	**Dropouts**	**Newcomers**	**Experienced**
Some newcomers need to contact a real person							• (1)	• (2)
Message was considered rude							• (2)	

Some newcomers may consider a message rude (*SLR: 0; stu: 0; OQ: 0; int: 2; 2 projects*) due to cultural interpretation: "All the community is very nice. Of course, there are some [nationality] guys. One time, a guy was rude with me, but, you know, we . . . are not used to the [nationality] way to talk directly"' [N13]. In the OSS scenario, projects rely only on textual communication and often involve people who do not want to spend time writing careful messages.

Another kind of barrier evidenced concerns for the need for a personal contact (*SLR: 0; stu: 0; OQ: 0; int: 3; 2 projects*) to create a bridge, or a stronger connection, to the project. One experienced member reported the specific case of his compatriots: ". . . people behave more emotionally in our country, I mean, newcomers need some special attention. Maybe since we are Mediterranean people, I don't know, but I think this is not the case in many projects in the world." [E9]. An experienced member from another project also reports the same issue in a more general context: "Although it may be a cultural aspect of open source that people prefer to make initial contact with a real person, so I don't have a problem with that" [E7].

Although the literature underscores cultural differences among participants as an important aspect (Preece, 2001; Koh et al., 2007; Ji et al., 2010; Nguyen and Fussell, 2013), a systematic review by Steinmacher et al. (2013a) showed that few studies investigated or dealt with cultural issues in distributed software development. This category of barriers could be another fruitful research topic.

4.4 Segmented analysis of the data

In this section, we report the analysis considering different dimensions of our participants and projects. First, we analyze how barriers reported by first-timers to OSS, in general, compare to the answers from those with prior experience in another project. Then, we present the results of the analysis considering how the characteristics of the OSS project may be related to the barriers.

4.4.1 First-timers vs. "experienced" newcomers

As it is noticeable in table 9, most of the mentions to social barriers from newcomers had been identified in first-timers' data. We understand that it occurred because,

Table 9 Barriers according to first-timers and experienced newcomers

Barrier Category	**Barrier**	**1st experience in OSS?**	
		Y **(26 newcomers)**	**N** **(9 newcomers)**
Reception issues	Not receiving an answer	• (1)	
	Delayed answers	• (2)	
	Impolite answers	• (2)	
	Receiving answers with too advanced/ complex contents	• (2)	
Newcomers' communication behavior	Not sending a meaningful/ correct message		
	English level		
	Shyness		
	Making useless comments in the mailing list/forum		
	Low responsiveness		
	Not acknowledging/thanking answers received		
Newcomers' orientation	Difficulty finding a mentor	• (4)	• (2)
Cultural differences	Some newcomers need to contact a real person	• (1)	
	Message is considered rude	• (1)	• (1)

as reported by Dagenais et al. (2010), newcomers are like explorers in an unknown landscape . Thus, during their first contact with the OSS world, they are susceptible to all kinds of threats. Specifically related to social barriers, we observed that the first-timers are sometimes unaware of community protocol, language, or even the most appropriate communication means to use. This is clear if we observe the *reception issues*, since all of them were reported by first-timers. Even for the *cultural differences* barriers, we found that they are mainly reported by first-timers. The only mention by an experienced newcomer points out to an issue that occurred when he was starting his journey in OSS. There are two other points that we can observe in the table. Firstly, newcomers did not report any problem related to their communication behavior. Secondly, a significant number of newcomers (both first-timers and experienced) reported as a barrier the difficulty in finding someone to help during their first steps.

4.4.2 Projects' characteristics

We grouped the projects according to certain characteristics and observed whether the barriers occurred in projects within the characteristics. We analyzed size in Lines of Code (LoC), programming language, age, and forge where the project is hosted (at the time of data collection). In Table 10, we present these characteristics and the number of participants per project that mentioned each social barrier. The information used to characterize the projects was gathered from OpenHub. [2]

With the exception of *not receiving an answer*, all the other barriers under the reception category were identified in at least four projects. Only participants related to JabRef reported *not receiving an answer* as a barrier for newcomers, including a core member and a student attempting to contribute to the project. It is possible that the

[2] `http://www.openhub.org`

Table 10 Barriers according to projects characteristics (the number in parenthesis represents the number of projects with that characteristic)

	Size (KLoC)				Main programming language							Project Age			Hosting forge			
	>1,000	**200–500**	**100–200**	**>100**	**Java**	**C++**	**PHP**	**Javascript**	**Ruby**	**C**	**Multiple**	**>10 yrs.**	**6–10 yrs.**	**<6 yrs.**	**GitHub/GitLab**	**SourceForge**	**Own Hosting**	**LaunchPad**
# of projects –>	**5**	**4**	**6**	**3**	**9**	**4**		**1**	**1**	**1**	**1**	**8**	**9**	**2**	**6**	**5**	**6**	**1**
Barrier																		
Not receiving an answer			2(6)		2(9)							2(8)				2(5)		
Delayed answers		1(4)	2(6)	1(3)	3(9)				1(1)			2(8)	2(9)		2(6)	2(5)		
Impolite answers	2(5)	1(4)		1(3)				1(1)	1(1)		1(1)	1(8)	2(9)	1(2)	2(6)		2(6)	
Receiving answers with too advanced/ complex contents	2(5)	1(4)		1(3)	1(9)	3(4)						3(8)	1(9)		1(6)		2 (6)	
Newcomers do not acknowledge/thank answers received			1(6)		1(9)							1(8)				1(5)		
Shyness	1(5)	1(4)							1(1)				2(9)		1(6)		1(6)	
English level	2(5)	1(4)	1(6)		2(9)	2(4)						4(8)				2(5)	2(6)	
Making useless comments in the mailing list/forums	1(5)												1(9)				1(6)	
Low responsiveness			1(6)		1(9)							1(8)				1(5)		
Not sending a meaningful/correct message		1(4)	1(6)		2(9)							2(8)				2(5)		
Finding a mentor	5(5)		3(6)	1(3)	4(9)	2(4)	1(1)	1(1)			2(1)	7(8)	2(9)	1(2)	1(6)	3(5)	6(6)	
Some newcomers need to contact a real person	1(5)	1(4)							1(1)				2(9)		1(6)		1(6)	
Message received is considered rude	2(5)					1(4)						1(8)	1(9)				2(6)	

problem was related to the JabRef community's structure, which was a bazaar with contributions coming from many volunteers, no sponsors, and no paid contributors.

When grouping the projects by characteristics, we noticed that reception issues crosscut all kinds of projects. One exception was found with regard to age. Only 1 citation to reception issues (*impolite answers*) came from participants from projects less than 5 years old.

An interesting finding is that the number of barriers under this *newcomers' communication behavior* category grew according to the size of the project and age. These are characteristics of established projects, which possibly require more intensive study and commitment from newcomers. Moreover, the experienced members of such projects possibly had interacted with a larger number of newcomers interested in joining the projects.

Difficulty finding a mentor was repeatedly reported, appearing in six projects. This barrier was identified in different projects, regardless of size, age, or programming language used. The only difference we found in this category concerns the hosting forge. We identified mentions to this barrier in the data obtained from participants from three projects hosted on their own forges and two hosted by Sourceforge. On the other hand, we found only one mention for a project hosted on GitHub (Etherpad). This may be attributed to the social coding paradigm introduced by GitHub, which offers a number of social features that make unique information about users and their activities visible within and across projects (Tsay et al., 2014).

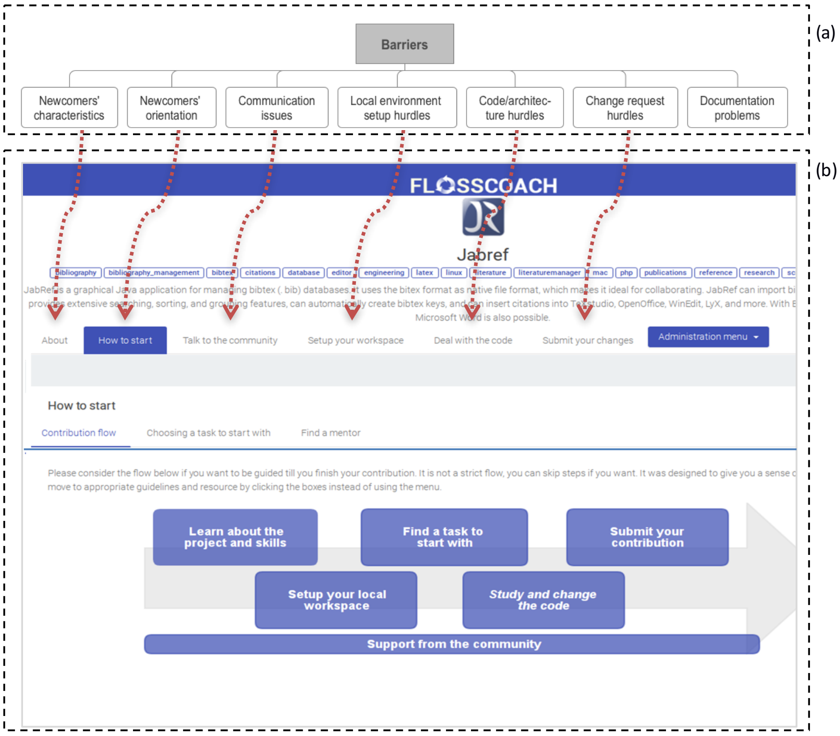
In summary, we found indications that some barriers are related to projects' specifics or singularities. The most distinctive characteristics identified in our data were projects' age and size. *Projects with a larger and older codebase seem to present more, and higher, barriers, which can be clearly related to the Lehman Laws* (Lehman, 1996). Some interesting findings concern project hosting site. We found indications that projects hosted on GitHub present fewer barriers, mainly related to *newcomers' communication*. As the focus of our study was not to observe such differences, future work may uncover these differences, especially regarding social coding environments.

5 Phase II - FLOSSCOACH: A portal to support newcomers' first contributions

In this kind of research, which generates a theoretical model, it is important to verify how the theory can influence the practice. Therefore, in the second phase of the study, we investigated whether the proposed model can be used to organize information for newcomers and reduce barriers. We were also interested in identifying inconsistencies in terminology and ambiguities in the model. For that purpose, we developed FLOSScoach, a portal to guide OSS project newcomers' first contributions, following the categorization presented in the barriers model to organize the portal. In each portal section, we provided information and links to help newcomers overcome the barriers related to a given category. To populate the portal, we used information provided by our interviewees and manually collected from the project websites. We organized the information into sections and subsections, in accordance with the barriers model, as shown in Figure 5.

After developing the portal, we evaluated it by conducting a study relying on qualitative data from diaries (Naur, 1983; Jepsen et al., 1998). The diary study allows access to everyday behavior in a relatively unobtrusive manner, which affords access to the experience's immediacy, and also provides accounts of phenomena over time (Symon, 2004). The diary study was conducted in two iterations and our participants were undergraduate students. Students are potential OSS project contributors and several programs (e.g., Google Summer of Code, Facebook Open Academy) currently focus on attracting them. The students chosen for this study had enough knowledge to fix small bugs in software projects and were motivated to contribute (since it was part of a course assignment). They joined a real project with real issues, and they interacted with the actual code and community. We highlight that similar assignments had been applied to evaluate the students in previous editions of the courses in which the study was conducted. Thus, the task is part of the course syllabus and it will continue to exist in following editions. Moreover, the grades were not related to the contribution itself, but to the process followed by the students (reported by means of shared diaries). They were also informed that their diaries would be analyzed as part of a research and that they could opt-out at any moment.

We split the participants into two groups. In order to split the students into case (those that had access to FLOSScoach) and control (those that did not receive credentials to FLOSScoach), we: (i) matched students and projects according to the expertise in the programming languages; (ii) ranked students according to their previous experience in OSS and industry (in the case of ties, we ranked them randomly); and



JabRef is a graphical Java application for managing bibtex (. bib) databases. It uses the bitex format as native file format, which makes it ideal for collaborating. JabRef can import bi
provides extensive searching, sorting, and grouping features, can automatically create bibtex keys, and can insert citations into TeXstudio, OpenOffice, WinEdit, LyX, and more. With E
Microsoft Word is also possible.

Please consider the flow below if you want to be guided till you finish your contribution. It is not a strict flow, you can skip steps if you want. It was designed to give you a sense c
move to appropriate guidelines and resource by clicking the boxes instead of using the menu.

Fig. 5 Barrier categories mapped to the FLOSScoach sections.

(iii) followed the ranked list (considering the project matching from the first step) to accommodate the individuals in control and case groups, aiming to balance the groups. For both iterations (each one in a different university), we first profiled the participants to verify their experience level in software development, OSS, and programming languages. All the participants were asked to contribute with code to a given project, by fixing bugs or implementing new features according to what was already reported in the projects' issue trackers. They had one month to deliver the task.

Iteration 1. The first iteration was conducted with fourteen students attending a Software Engineering course (3rd year) at Federal University of Technology - Paraná (UTFPR), Brazil. All participants were newcomers to software development. Only five of them had worked in industry, but for less than one year. Only one participant reported previous experience contributing to OSS projects. We directed the participants to two specific OSS projects, LibreOffice and JabRef, which were part of our previous phase; moreover, this study's first author is a JabRef contributor.

Iteration 2. The second iteration was conducted during a Software Engineering course for Computer Science majors at the University of São Paulo (USP), Brazil. The initial number of participants was 51. Their profile was a little different from the participants of Iteration 1. The main difference is that the participants had more industry experience. As we were counting on a larger number of newcomers, we prepared FLOSScoach with information from four other OSS projects: Amarok, Empathy, Vim, and Audacity. We chose projects written in C/C++ to match the participants' language skills.

We allowed our participants to work whenever they desired, conducting the work at a chosen time and place (as usually occurs in volunteer OSS contributions). We adopted unstructured diaries, in which participants might provide open-ended, stream-of-consciousness narratives about their activities and experiences, which can result in rich and detailed accounts (Palen and Salzman, 2002). The participants were trained to write diaries in a shared document logging their activities, issues, and everything that they did while working on the assignment.

Regular interaction between the investigator and participants is key to ensuring the subjects understand the diary entries' importance and provide the desired level of details. Therefore, we used shared documents to keep the diaries. We constantly followed the entries and provided prompt feedback to the subjects via annotated comments. We asked for additional details and clarifications, such as how they achieved something, why they made a given decision, where they found a piece of information, and what their feeling was regarding what happened.

To complement and clarify some information found in the diaries, we conducted a debrief session with the participants after the assignment deadline. The participants answered open questions that were asked according to a structured questionnaire.

We qualitatively analyzed the entries and the data collected in the debrief session using open coding and axial coding procedures. The analysis was conducted by one researcher and reviewed by a second researcher. The review was followed by a discussion phase involving both researchers. In the debrief sessions, we also collected suggestions for improving the portal and its organization. We used this feedback to refine the portal and the model after each iteration.

5.1 Diary Analysis

From the initial 65 participants, we considered only 44 for analysis. We dismissed subjects if fewer than three diary entries were written or the contribution was not a code contribution. Thus, from the 34 participants in the control group (not using FLOSScoach), 19 were considered (56%) in the analysis. From the 31 participants assigned to use FLOSScoach, 24 were considered (77%). This may indicate that FLOSScoach fostered or facilitated assignment completion. In addition, three participants originally assigned to the portal group informed us that they did not use FLOSScoach. Therefore, we excluded them from the analysis. To facilitate reporting our results, in this section we identify the participants as follows: C1-XX: participant from Control group, in Iteration 1; C2-XX: participant from Control group, in Iteration 2; P1-XX: participant from Portal group, in Iteration 1; and P2-XX: participant from Portal group, in Iteration 2. Since all the diaries were written in the students' native language (Brazilian Portuguese), the quotes presented to ground our findings are free translations from the excerpts.

Table 11 Interaction with the community per channel

	Mailing list	IRC channel	Issue Tracker	Contact with Mentor
Case Group	4 students: FC1_07, FC2_05, FC2_09, FC2_18	3 students: FC1_07, FC2_01, FC2_17	11 students: FC1_01, FC1_02, FC1_07, FC2_02, FC2_03, FC2_05, FC2_08, FC2_09, FC2_10, FC2_11, FC2_15	1 student: FC2_17
Control Group	4 students: C2_03, C2_05, C2_06, C2_13	3 students: C1_07, C2_06, C2_10	3 students: C2_06, C2_07	

5.1.1 Results

Fifteen (out of 43) participants interacted (or attempted to interact) with the community during our study. It was common to find mentions in the diaries of IRC chats, mailing list messages, and issue tracker entries. Seven participants of the control group and 8 from the portal group interacted with the community in different ways. We noticed only a small number of social interactions between newcomers and communities during our study. Two participants used the issue tracker (Bugzilla) to discuss a change request (issue) and one participant joined IRC sessions and the mailing list, as well as used the issue tracker to clarify doubts.

During Iteration 1, among the four people (3 from FLOSScoach group) who interacted with the community, one participant explicitly mentioned (twice) that he was following FLOSScoach's suggestion to interact with the community: "I will explore FLOSScoach a little more. As it suggested, I will seek for help in the community. I will write a comment at the Bugzilla, to confirm my understanding about the bug and ask if I can assign the task to myself." and "I sent an email with my doubt to the mailing list... I opened my inbox... there are many answers to my answer, I am impressed!" - FC1_07

During Iteration 2, we noticed a higher number of subjects interacting with the community. It was common to find mentions of IRC chats, mailing list messages, and entries in the issue tracker. Once again, we noticed more interactions from those students with access to FLOSScoach than those from the control group. Considering both iterations, 13 students (54%) that had access to FLOSScoach reported interaction with the community versus 7 from the control group (37%). We report the channel used by the newcomers in Table 11.

In the case group, it is possible to observe that most newcomers interacted with the community via issue tracker. A possible explanation is that people followed the instructions presented in the portal: "Use the comments to inform the community you are working on that task or to get support/ask for help regarding the task!" This was clearly reported by FC2_09: "As suggested in FLOSScoach, I commented on the bug telling the community I am working on this." In addition to the means of communication, we observed that the students received timely and welcoming answers. However, we found three cases in which the newcomers did not receive any answer from the community, which lead to frustration.

> "This was supposed to be a moment of excitement and anxiety, but the silence of the community made me feel anesthetized. On the one hand, this is an activity that does not bother me, on the other hand, it did not lead me to an effusive state as I imagined at the beginning of the task. I am blasé!" - C2_06.

> "I asked this question after searching for a certain period. I posed the question politely, as required by the community. I received no answer. . . " - FC2_09.

> "Let's see how friendly these guys are. I added a ticket, asking for help. . . " [4 days later] "I did not receive any answer" - FC2_10.

An important finding is that the newcomers who reported receiving responses in their diaries did not mention any cases of receiving improper answers. All the participants who communicated with the community mentioned that they received welcoming messages and proper orientation: "I surprisingly received an answer to my email few hours after sending it to the community" - FC2_15; "The community members were very friendly and welcoming. I will probably become a regular contributor." - FC2_17; and "I joined the IRC channel of the LibreOffice developers, and they helped me a lot." - C2_10.

One participant mentioned another benefit of using FLOSScoach: that the message template provided by the portal was helpful. He reported that the template helps newcomers to be clearer, more concise, and to reduce the shyness: "I liked the message template, showing how to introduce myself and to present the problems I am facing. Even having proficiency in English, I did not know the the most polite way of asking for help. This example helped to be clear, concise in presenting the message objective, and also to reduce shyness" - FC2_01.

During the debrief session we focused on understanding why the participants had not tried to interact with the community members. Five participants mentioned two reasons, already cataloged on our barriers model: English proficiency and fear. Four students mentioned the need to use English as the reason why they avoided approaching the community. Two students shared that they did not feel confident enough to send their questions, saying they were afraid to do so. Their fear was related to lack of self-esteem and shyness, and can be verified in the following quotes:

> "I did not try. I was afraid... of sending a newbie question, I don't know. Fear of being repressed. Because of the English, too. But, I think that even if it was in Portuguese I would not send." - C1_07.

> "I feel a little annoyed to talk about it, but I was afraid of the community. What if I ask something that is too simple. I think the community is something beautiful, that everybody is there to help, that they will welcome me because they need more contributors. . . I don't know, it is a little scary. What if I am not good enough to do what was proposed. . . and. . . the community members do not like me. . . It is a problem with me." - C1_06 (debrief).

One participant reported that the content of FLOSScoach was complete enough, and in his case talking to the community was not necessary. "I did not need to talk to them. The tool was very clear. It is very easy, [the portal] is very good." - FC1_02. This participant effectively contributed to JabRef.

In summary, we found evidence that the portal reduced the need for communication and that the portal supported newcomers who needed to interact with the community. In addition, we found evidence that FLOSScoach supported social interactions by guiding the newcomers and pointing them to the right communication means; however, in some cases, it was not enough to reduce the barriers related to, for example, a limited background in English. Finding better ways to foster communication with the community is a topic that deserves further investigation.

5.1.2 Updating FLOSScoach and the barriers model

We updated FLOSScoach after Iteration 1 according to the feedback received from the students, which we asked for during the debrief sessions. Two participants did not understand why repeated information appeared in different places. This repetition occurred because we built the first version of the portal by directly mapping the categories onto the portal. This led us to provide a "documentation" section; however, the information under documentation already appeared in other categories. Another issue regarding the portal's organization concerned the order of the categories presented to the user. One participant suggested presenting the categories in the same order as they appear in the contribution flow. In addition, he suggested avoiding the use of submenus, making all possible categories only one click away. We accommodated and rearranged the portal structure for Iteration 2.

After concluding the study using FLOSScoach, we also updated the barriers model using the feedback from this study as input. Regarding social barriers, a category called *communication* was created to accommodate problems related to *newcomers' communication* and *community reception issues*. We observed that the barriers under these two categories were closely related and categorized them under a single category. In addition, *cultural differences* barriers were included under the *communication* category; *message is considered rude* was merged with *reception issues/send a message that is considered impolite*; and *some newcomers need to contact a real person* was moved under *newcomers' communication*. This change was made in response to the barriers' proximity to the mentioned categories. We present the post-rearrangement model in Figure 6.

6 Discussion

This study uncovered empirical evidence of the barriers faced by newcomers to OSS projects when placing their first contribution. This empirical evidence is important, as many studies are motivated by or deal with anecdotal evidence. This paper brings evidence from real contexts, which are rarely precisely documented. The model presented in Figure 6 groups social and non-social barriers into five categories, illustrating the high number of technical barriers compared to social barriers. This imbalance can be explained by the characteristics of OSS communities, which demand contributors and tasks with specific technical skills and knowledge requirements.

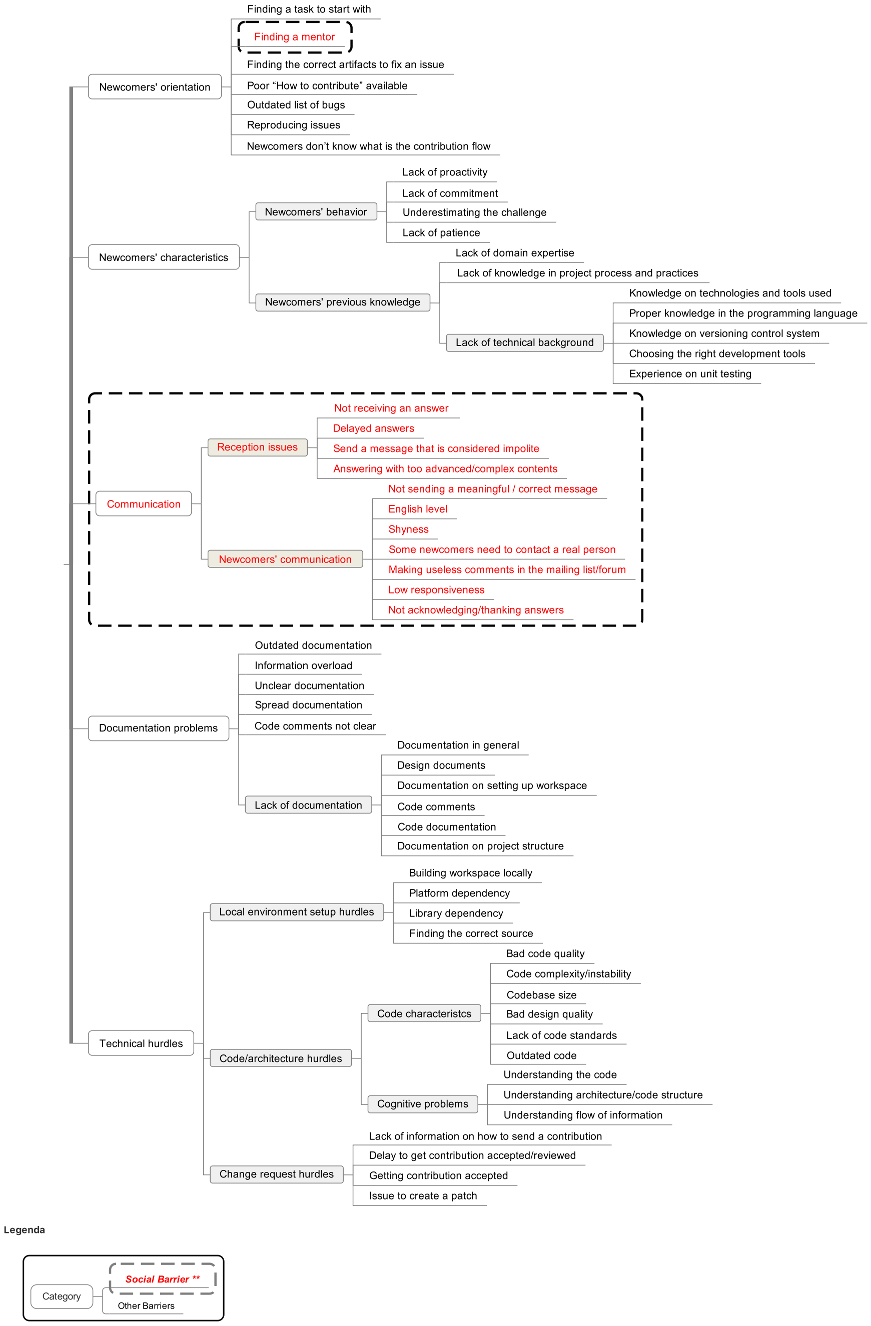

Newcomers' orientation
Finding a task to start with
Finding a mentor
Finding the correct artifacts to fix an issue
Poor "How to contribute" available
Outdated list of bugs
Reproducing issues
Newcomers don't know what is the contribution flow
Newcomers' characteristics
Newcomers' behavior
Lack of proactivity
Lack of commitment
Underestimating the challenge
Lack of patience
Newcomers' previous knowledge
Lack of domain expertise
Lack of knowledge in project process and practices
Lack of technical background
Knowledge on technologies and tools used
Proper knowledge in the programming language
Knowledge on versioning control system
Choosing the right development tools
Experience on unit testing
Communication
Reception issues
Not receiving an answer
Delayed answers
Send a message that is considered impolite
Answering with too advanced/complex contents
Newcomers' communication
Not sending a meaningful / correct message
English level
Shyness
Some newcomers need to contact a real person
Making useless comments in the mailing list/forum
Low responsiveness
Not acknowledging/thanking answers
Documentation problems
Outdated documentation
Information overload
Unclear documentation
Spread documentation
Code comments not clear
Lack of documentation
Documentation in general
Design documents
Documentation on setting up workspace
Code comments
Code documentation
Documentation on project structure
Technical hurdles
Local environment setup hurdles
Building workspace locally
Platform dependency
Library dependency
Finding the correct source
Code/architecture hurdles
Code characteristcs
Bad code quality
Code complexity/instability
Codebase size
Bad design quality
Lack of code standards
Outdated code
Cognitive problems
Understanding the code
Understanding architecture/code structure
Understanding flow of information
Change request hurdles
Lack of information on how to send a contribution
Delay to get contribution accepted/reviewed
Getting contribution accepted
Issue to create a patch
Legenda
Category
Social Barrier **
Other Barriers

Fig. 6 Barriers Model after the study with students

We related our findings to those reported in the broader CSCW literature, especially Wikipedia (Bryant et al., 2005; Choi et al., 2010; Halfaker et al., 2013, 2011). We observed that the social barriers identified are fairly similar to those evidenced in other domains. Thus, some of the solutions and mitigation strategies used in those contexts could be tried on OSS communities and vice versa.

Some barriers are well reported and analyzed by the literature; the most explored barriers are those related to reception issues. For example, not receiving an answer is well evidenced quantitatively in both OSS (Steinmacher et al., 2013b) and Q&A literature (Arguello et al., 2006), and these results were in line with our results. Receiving improper answers was studied in the CSCW literature, mainly on the analysis of reverts (Halfaker et al., 2011) and use of bots (Tsvetkova et al., 2017) in Wikipedia. The proposed strategies of automated answers and feedback, as used in Wikipedia can be adapted and then evaluated in the OSS context. On the other hand, some barriers identified by our study are neglected in the literature. One of them is *receiving answers with too advanced/complex contents*. We could not identify any study dealing with this barrier. There are opportunities to investigate this kind of barrier in different domains to provide a deeper understanding of this issue.

The barriers related to newcomers' communication behavior are also understudied. These barriers were evidenced by the community members' perspective, which is rarely investigated. Sometimes, the newcomers themselves create the barriers when they post useless comments or do not acknowledge an answer (Joyce and Kraut, 2006; Burke et al., 2007), which heightens the need to better understand what the community expects from the newcomers, and how these expectations affect or impact the newcomers' reception issues.

Cultural differences also deserve to be highlighted here, since they also appear in CSCW literature (Preece, 2001; Koh et al., 2007; Ji et al., 2010; Nguyen and Fussell, 2013). There are some possibilities related to this barrier that can be followed up on different CSCW domains. For example, how do collaborators observe cultural differences? Do they consider such differences when dealing with their peers? Do these barriers cause issues related to trust, misunderstandings, etc.?

Moreover, we understand that the roles of the people involved in OSS can be analyzed from different perspectives, including economic (Ostrom, 2000), sociological (Lave and Wenger, 1991; Wenger et al., 2009) and psychological (Thompson and Fine, 1999). We acknowledge the existence of studies building upon some theories to understand specific aspects of developer onboarding Fang and Neufeld (2009) – like using Legitimate Peripheral Participation (LPP) theory (Lave and Wenger, 1991). Therefore, we point that analyzing the nature of the barriers from these perspectives is a potential future direction towards understanding the social construction of OSS.

In the second phase of our study, we built FLOSScoach, a portal to support newcomers during their first steps, and evaluated how it would affect the newcomers' perception of social barriers. By using FLOSScoach, we observed that some social barriers, mainly related to communication, were softened by reducing the need for community interaction, as reported by one participant: "I did not need to talk to them. The tool was very clear. It is very easy, very good." - FC2_02. Based on our work, we can identify various features that have the potential to help OSS project newcomers overcome social entry barriers. Regarding communication, providing a message template facilitates and encourages newcomers to interact with the community, and helps them send meaningful and complete messages. Moreover, it seems important to explicitly tell newcomers that it is important to find answers to their questions in the mailing list archives before posting a question to the list. We helped this by embedding a custom search box (which enables newcomers to search the mail archives) on the initial page of the *Communication* portal section.

Furthermore, in our experience, providing a clear contribution flow helps newcomers gain confidence about what to do and in what order. This reduces the need to communicate, since the first steps are clearly presented to those who are not accustomed to the process of contribution. Aligned with this, newcomers can benefit from a specific page that contains only the resources they need and not flood them with every possible resource, since too much information can confuse them. By using the model of barriers to place the proper signs and maps, we helped newcomers orient themselves, and alerted them about the obstacles they might face.

6.1 Guidelines for OSS Communities to avoid/reduce social barriers

Based on our results, we present a set of guidelines for communities that want to offer appropriate newcomer support and for newcomers who want to contribute to an OSS project. All the guidelines presented here are backed by evidence we collected and/or are supported by the literature.

Answer quickly. Some of our interviewees reported that they were never replied to: "They never answered our forum post. We spent a lot of effort on something that was already being done. . . " An interview with a core member also revealed this challenge: "In my opinion, the first [barrier] is not getting any reply." Newcomers are mostly volunteers and the community should not let their motivation wane by making them wait or leaving them without an answer. Automatic greetings could be used to help (Preece, 2004), at least to say that someone will answer quickly or to guide newcomers to the appropriate communication channel. Leaving a good first impression is very important, as stated by Fogel (2013): "if a project doesn't make a good first impression, newcomers may wait a long time before giving it a second chance."

In addition to providing an answer, it is important to *be kind and make newcomers feel part of the team*. A community can make newcomers feel welcome and keep them motivated by treating them as potential contributors and showing them that the community cares about them. Sending thankful, welcoming messages helps in dealing with cultural differences and misunderstandings. It is known from the literature (Singh, 2012) that receiving impolite answers demotivates newcomers. In our study, an experienced member reported that ". . . some developers may not be suitable for receiving newcomers, they may get angry pretty quickly and kill the interest of the newcomers. Very few of the newcomers know how to behave against this kind of tough developers." Designating selected members to deal with new members or setting a code of conduct are possible solutions to such reception issues.

Identify mentors or experts. As mentioned by a newcomer, "some basic handholding would help." Difficulty in finding a mentor was identified by both newcomers and core members as a barrier. In addition, mentorship can play a role in keeping newcomers motivated and helping them overcome potential barriers. Large OSS projects (e.g., Apache, LibreOffice, and Mozilla) already provide mentoring programs. In Mozilla projects, for example, some bugs are mentored. On Mozilla's page, they are presented as bugs that "have a mentor who commits to helping you every step of the way. Generally, there should be enough information in the bug to get started..." Apache also offers a mentoring program that focuses on providing mentors for anyone interested in contributing effort to an Apache Software Foundation (ASF) project. It is a formal program where newcomers can subscribe to receive a mentor. The LibreOffice community provides a wiki page called "Find the Expert," where they list a set of developers who are experts in specific knowledge areas within the project. In addition, Google's Summer of Code program provides scholarships for students interested in writing code for OSS projects, and the selected projects assign mentors to support the students during their scholarship period.

Create a newcomer-specific page. Give the newcomers every resource they need, and only the resources they need. Do not flood newcomers with every possible resource, since too much information can confuse them. Show only what is important for newcomers' first steps, like how the project is organized, and what/where are the important resources (code repository, mailing lists, issue tracker, IRC channel, code review tool). Keep the page clean, organized, up-to-date, and easy-to-follow. Make this space a kind of *new developers' guidelines* section.

6.2 Guidelines for newcomers to reduce social barriers

Be proactive. Try to overcome the barriers you face by searching for solutions yourself. This is expected by the community, as stated by one core member we interviewed: "You cannot wait for other people. You have to be willing to study new stuff by yourself." A member of another project mentioned that "I think the only requirement is that when a newcomer asks, we want to see that he or she did some research before asking." Therefore, for someone willing to contribute to a project, we suggest searching the mailing list archives, other resources made available by the community, and specialized forums to solve their problems before contacting that community. It is possible that their question has been previously answered.

Do not be afraid of the community. If newcomers could not solve problems by themselves or by using the resources made available, it is important to identify the appropriate communication means and talk to the community. We found that shyness is a barrier; however, communicating is necessary in collaborative environments such as OSS projects. It is important to observe that newcomers need to be careful about how to interpret the answers. Cultural differences may impact the way people communicate, as one experienced member highlighted: "a guy sent me a rude message, but, you know, we ... are not used to the '[nationality] way of talking directly."'

Keep the community informed about your decisions. If you choose a task to work with, send a comment in the issue you are working on informing others that you are trying to address it. In the same way, inform them whenever you give up or find any problem related to that task. This way, you are contributing to the community and avoiding concurrent work.

Send a kind and meaningful message. Some tips on how to send a message to the community: be kind, present yourself by mentioning your skills and your goals, ask your question clearly and objectively, and show what you have tried to do to solve your problem before referring to the community. While evaluating FLOSScoach, we found that using a template was helpful ("I liked the message template, showing how to introduce myself and to present the problems I am facing."). The mentioned template is presented below.

> Hello,
>
> My name is [your name] and I am a newcomer trying to place my first code contribution to [project]. I am facing problems [during my first steps/finding a task/setting up my workspace]. Can someone help me [clarifying some questions / mentoring me]?
>
> I have already [mention the things you have done already to try to solve your problem] [If you are getting an error, include it in the message] [Mention the OS you are working on and the tools you are using] Thanks in advance
>
> [YOUR SIGNATURE]

6.3 Implications to research

Our preliminary investigation of FLOSScoach revealed that it is possible to make use of the barriers model to support newcomers overcoming social barriers. By including the context of CSCW and related research literature, we provide a starting point to conceive of such support. More work is necessary on specific research topics, such as understanding (and creating ways to measure) the influence of the barriers in newcomers' experience, identifying and creating different strategies to lower each barrier, and proposing metrics to grade the support offered for each barrier. Researchers can also adapt the strategies used in FLOSScoach and the existing strategies used in other domains to design their own tool-based support.

Moreover, by substantiating and characterizing the barriers faced by newcomers in a particular domain, we believe that the CSCW researchers can benefit from our research. Future research ought to focus on searching for commonalities and differences among barriers faced in different domains in order to build models and theories about joining processes in open collaboration communities.

7 Limitations and threats to validity

Although we analyzed data from a variety of sources and from different projects, we likely did not discover all possible barriers or provide full explanations of the barriers. We are aware that each project has its singularities and that the OSS universe is massive, meaning the level of support and the barriers can differ according to the project or the ecosystem. Our strategy to consider different projects and different developer profiles aimed to alleviate this issue, identifying recurrent mentions of barriers from multiple perspectives.

In our systematic literature review, we considered any study that empirically evidenced barriers faced by newcomers during their joining process. We did not constrain our search to those papers that focused on the very first contribution of newcomers, since they do not clearly state which phase was analyzed. Therefore, there could be studies focusing on different phases of the joining process.

Another threat to the results' validity is the data classifications' subjectivity. We grounded all our results in the data collected. Additionally, we discussed the analysis process along with two other researchers to foster a better validation of the interpretations through mutual agreement.

During interviews, experienced members were asked to answer questions regarding barriers faced by newcomers when they were seeking to place their first contribution, but due to memory effects, they may have referred to the entire joining process. To avoid this kind of situation, we reinforced the focus of the research and verified some answers.

Since we sent open invitations to a mailing list, sampling bias affects our interviewees and open question respondents, namely self-selection bias and social desirability bias. However, we counteracted this effect by seeking out different sources and analyzing the answers in context to identify specifics.

We understand that the use of textual chat as the interview means can be considered a threat. The possibility of context change and the execution of parallel activities that distract the interviewees can be a negative aspect of using this means. The use of Instant Messengers has been discussed in the social sciences (Opdenakker, 2006; Hinchcliffe and Gavin, 2009), where authors underscore positive effects of using these tools. In our case, we chose to use this means once the participants are accustomed to the environment (they could choose the IM that they were more familiar with), and electronic means are the default (and preferred) form of communication in OSS projects.

Diary studies can also introduce concerns related to validity. First, diary entries are inherently filtered through participants' subjective experiences. Second, diary studies follow a case-study approach. We aimed for in-depth understanding, rather than statistical validity; we read students' entries consistently and frequently requested additional information or explanation to attempt to counteract withholdings. Third, since the researchers interacted with a high number of diaries daily, there could be some cases in which questions and/or answers inadvertently directed students' responses, hiding potential barriers faced or steps followed by the students. However, by analyzing the data, we did not notice this issue.

The use of students could have affected the results' generalizability. Moreover, most of our participants in the second phase were novices to software development in real settings (with no previous industry experience), and thus it is possible that some barriers they faced are not specific to OSS development. However, we highlight that students are potential contributors to OSS projects. During debrief sessions, which took place after the assignment, students may have felt that they needed to provide positive feedback. To avoid this, we emphasized to both groups that the feedback had no bearing on their grade and that it would be analyzed after grading.

Finally, this research was conducted between 2013 and 2015. When it was started, social coding environments such as GitHub were not as prominent as they have become more recently. It is possible that the communities evolved and lowered barriers for newcomers (Yu et al., 2015). It is a potential future direction to investigate how the barriers evolved with the growth of GitHub and how migration of the projects like JabRef to GitHub impacted the barriers.

8 Conclusion

Numerous communities are maintained by volunteers, who can easily drop out (Burke et al., 2007). An explicit effort is necessary to mitigate obstacles and problems in these communities (Wang et al., 2012a). In this study, we identified, organized, and discussed social barriers that hinder newcomers' first contribution to OSS projects. The barriers emerged from data collected from multiple sources. This study differs from the existing literature by focusing on the social barriers newcomers face when attempting to place their first contribution to a project, rather than focusing on the entire joining process. Additionally, our study followed a qualitative approach, which complements the existing dominant quantitative-based research on newcomers' joining process (Halfaker et al., 2011, 2013; Jensen et al., 2011; Wang et al., 2012b; Zhu et al., 2012; Zhou and Mockus, 2012).

In Phase I of this work, our goal was to identify the barriers faced by OSS newcomers. The main result of this phase was a preliminary model comprising 58 barriers grouped in six different categories. We observed that less than 30% of the barriers (17 barriers) appeared in the literature. In particular, some social barriers, such as cultural differences, receiving answers with too advanced/complex contents, and newcomers' shyness, were evidenced by the practitioners, but not found in the systematic literature review. These barriers certainly warrant further investigations.

By stratifying the data according to newcomers' experience, we found that newcomers with no previous experience in OSS projects reported most of the social barriers. We believe that this occurs because first-timers are not aware of the social protocol, or the appropriate way to communicate in this kind of environment. By analyzing the barriers according to the characteristics of the projects, we could observe some specificities but barriers recur regardless of the analyzed project characteristics. We found indications that the older and larger the projects are the more barriers they present, and that projects hosted by so-called social coding environments (GitHub and GitLab) present fewer social barriers.

In Phase II, we found that organizing existing information according to the barriers model reduced the need for communication and indicated that the portal supported the newcomers who needed to interact with the community. This indicates that providing a friendly and organized environment to newcomers can support newcomers, avoiding or overcoming social barriers.

Our results and guidelines may provide insights for communities that want to smooth newcomer onboarding and lay a foundation for building better onboarding support tools. In addition, researchers can use the model to plan further qualitative and quantitative studies to investigate specific barriers, their interplay, and in-practice implications. OSS projects can benefit from additional contributions if they offer the right support to newcomers who are trying to contribute to the project for the first time. A smooth first contribution may increase the total number of successful contributions made by single contributors and, hopefully, the number of long-term contributors.

Acknowledgments

We would like to thank each of the students and developers who participated in our research. We also thank the reviewers for their valuable comments. This work is supported by the CNPq (Grant #430642/2016-4); and FAPESP (Grants #2015/24527-3 and 2014/21899-4).

References

Anthes, Gary (2016). Open Source Software No Longer Optional. *Communications of ACM*, vol. 59, no. 8, pp. 15–17.

Arguello, Jaime; Brian S. Butler; Elisabeth Joyce; Robert Kraut; Kimberly S. Ling; Carolyn Rosé; and Xiaoqing Wang (2006). Talk to Me: Foundations for Successful Individual-group Interactions in Online Communities. In: *CHI'06. SIGCHI Conference on Human Factors in Computing Systems, Montréal, Québec, Canada, 22-27 April 2006*. ACM: New York, NY, USA, pp. 959–968.

Balali, Sogol; Igor Steinmacher; Umayal Annamalai; Anita Sarma; and Marco Aurélio Gerosa (2018). Newcomers' Barriers...Is That All? An Analysis of Mentors' and Newcomers' Barriers in OSS Projects. *Computer Supported Cooperative Work (CSCW)*, vol. PP.

Bonaccorsi, Andrea; and Cristina Rossi (2004). Altruistic individuals, selfish firms? The structure of motivation in Open Source software. *First Monday*, vol. 9, no. 1, pp. [online].

Bryant, Susan L.; Andrea Forte; and Amy Bruckman (2005). Becoming Wikipedian: Transformation of Participation in a Collaborative Online Encyclopedia. In: *GROUP'05. 2005 International ACM SIGGROUP Conference on Supporting Group Work, Sanibel Island, Florida, USA, 6-9 November 2005*. ACM: New York, NY, USA, pp. 1–10.

Burke, Moira; Cameron Marlow; and Thomas Lento (2009). Feed Me: Motivating Newcomer Contribution in Social Network Sites. In: *CHI'09. SIGCHI Conference on Human Factors in Computing Systems, Boston, MA, USA, 4-9 April 2009*. ACM: New York, NY, USA, pp. 945–954.

Burke, Moira; Elisabeth Joyce; Tackjin Kim; Vivek Anand; and Robert Kraut (2007). Introductions and Requests: Rhetorical Strategies That Elicit Response in Online Communities, In: C. Steinfield, B. Pentland, M. Ackerman, and N. Contractor (eds.): *Communities and Technologies 2007*. Springer-Verlag London, pp. 21–39.

Cai, Yuanfeng; and Dan Zhu (2016). Reputation in an open source software community: Antecedents and impacts. *Decision Support Systems*, vol. 91 pp. 103 – 112.

Canfora, Gerardo; Massimiliano di Penta; Rocco Oliveto; and Sebastiano Panichella (2012). Who is Going to Mentor Newcomers in Open Source Projects? In: *FSE'12. ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering*, *Cary, North Carolina, 11-16 November 2012*. ACM: New York, NY, USA, pp. 44:1–44:11.

Choi, Boreum; Kira Alexander; Robert E. Kraut; and John M. Levine (2010). Socialization Tactics in Wikipedia and Their Effects. In: *CSCW'10. 2010 ACM Conference on Computer Supported Cooperative Work*, *Savannah, Georgia, USA – 6-10 February 2010*. ACM: New York, NY, USA, pp. 107–116.

Cubranic, Davor; Gail C. Murphy; Janice Singer; and Kellogg S. Booth (2005). Hipikat: a project memory for software development. *IEEE Transactions on Software Engineering*, vol. 31, no. 6, pp. 446–465.

Dagenais, Barthélémy; Harold Ossher; Rachel K. E. Bellamy; Martin P. Robillard; and Jacqueline P. de Vries (2010). Moving into a New Software Project Landscape. In: *ICSE'10. Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering - Volume 1*, *Cape Town, South Africa, 1-8 May 2010*, Vol. 1 of *ICSE'10*. ACM: New York, NY, USA, pp. 275–284.

David, Paul A.; and Joseph S. Shapiro (2008). Community-based production of open-source software: What do we know about the developers who participate? *Information Economics and Policy*, vol. 20, no. 4, pp. 364–398.

Deshpande, Amit; and Dirk Riehle (2008). The Total Growth of Open Source, In: B. Russo, E. Damiani, S. Hissam, B. Lundell, and G. Succi (eds.): *Open Source Development, Communities and Quality*, Vol. 275 of *IFIP – The International Federation for Information Processing*. Springer US, pp. 197–209.

Dittrich, Yvonne (2014). Software engineering beyond the project – Sustaining software ecosystems. *Information and Software Technology*, vol. 56, no. 11, pp. 1436 – 1456. Special issue on Software Ecosystems.

Dittus, Martin; Giovanni Quattrone; and Licia Capra (2016). Social Contribution Settings and Newcomer Retention in Humanitarian Crowd Mapping. In: . *International Conference on Social Informatics*, *Bellevue, WA, USA, 11-14 November 2016*. Springer International Publishing: Cham, pp. 179–193.

Ducheneaut, Nicolas (2005). Socialization in an Open Source Software Community: A Socio-Technical Analysis. *Computer Supported Cooperative Work (CSCW)*, vol. 14, no. 4, pp. 323–368.

Fang, Yulin; and Derrick Neufeld (2009). Understanding Sustained Participation in Open Source Software Projects. *Journal of Management Information Systems*, vol. 25, no. 4, pp. 9–50.

Farzan, Rosta; and Robert E. Kraut (2013). Wikipedia Classroom Experiment: Bidirectional Benefits of Students' Engagement in Online Production Communities. In: *CHI'13. SIGCHI Conference on Human Factors in Computing Systems*, *Paris, France, 27 April - 2 May 2013*. ACM: New York, NY, USA, pp. 783–792.

Faulkner, Ryan; Steven Walling; and Maryana Pinchuk (2012). Etiquette in Wikipedia: Weening New Editors into Productive Ones. In: *WikiSym'12. Eighth Annual International Symposium on Wikis and Open Collaboration*, *Linz, Austria, 27-29 August 2012*. ACM: New York, NY, USA, p. A5.

Fogel, Karl (2013). *Producing Open Source Software: How to Run a Successful Free Software Project*. O'Reilly Media, 1st edition.

Forte, Andrea; and Cliff Lampe (2013). Defining, Understanding, and Supporting Open Collaboration: Lessons From the Literature. *American Behavioral Scientist*, vol. 57, no. 5, pp. 535–547.

Greene, Gillian J.; and Bernd Fischer (2016). CVExplorer: Identifying Candidate Developers by Mining and Exploring Their Open Source Contributions. In: *ASE 2016. 31st IEEE/ACM International Conference on Automated Software Engineering*, *Singapore, Singapore, 3-7 September 2016*. ACM: New York, NY, USA, pp. 804–809.

Halfaker, Aaron; Aniket Kittur; and John Riedl (2011). Don't Bite the Newbies: How Reverts Affect the Quantity and Quality of Wikipedia Work. In: *WikiSym'11. Proceedings of the 7th International Symposium on Wikis and Open Collaboration*, *Mountain View, California, 3-5 October 2011*. ACM: New York, NY, USA, pp. 163–172.

Halfaker, Aaron; R. Stuart Geiger; Jonathan Morgan; and John Riedl (2013). The Rise and Decline of an Open Collaboration System: How Wikipedia's reaction to sudden popularity is causing its decline. *American Behavioral Scientist*, vol. 57, no. 5, pp. 664–688.

Hannebauer, Christoph; Matthias Book; and Volker Gruhn (2014). An Exploratory Study of Contribution Barriers Experienced by Newcomers to Open Source Software Projects. In: *CSI-SE'14. First International Workshop on Crowd Sourcing in Software Engineering*, *Hyderabad, India, 2 June 2014*. ACM: New York, NY, USA, pp. 11–14.

Hannebauer, Christoph; and Volker Gruhn (2016). Motivation of Newcomers to FLOSS Projects. In: *OpenSym'16. Proceedings of the 12th International Symposium on Open Collaboration*, *Berlin, Germany, 17-19 August 2016*. ACM: New York, NY, USA, pp. 1:1–1:10.

Hars, Alexander; and Shaosong Ou (2002). Working for Free? Motivations for Participating in Open-Source Projects. *International Journal on Electronic Commerce*, vol. 6, no. 3, pp. 25–39.

Herbsleb, James D.; and D. Moitra (2001). Global software development. *IEEE Software*, vol. 18, no. 2, pp. 16–20.

Herraiz, Israel; Gregorio Robles; Juan José Amor; Teófilo Romera; and Jesús M. González Barahona (2006). The Processes of Joining in Global Distributed Software Projects. In: *GSD'06. Proceedings of the 2006 International Workshop on Global Software Development for the Practitioner*, *Shanghai, China, 23 May 2006*. ACM: New York, NY, USA, pp. 27–33.

Hinchcliffe, Vanessa; and Helen Gavin (2009). Social and Virtual Networks: Evaluating Synchronous Online Interviewing Using Instant Messenger. *The Qualitative Report*, vol. 14, no. 2, pp. 318–340.

Hoda, Rashina; James Noble; and Stuart Marshall (2010). Using Grounded Theory to Study the Human Aspects of Software Engineering. In: *HAoSE'10. Human Aspects of Software Engineering*, *Reno, Nevada, 17-21 October 2010*. ACM: New York, NY, USA, p. Article 5.

Jalali, Samireh; and Claes Wohlin (2012). Systematic Literature Studies: Database Searches vs. Backward Snowballing. In: *ESEM'12. ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*, *Lund, Sweden, 19-20 September 2012*. ACM: New York, NY, USA, pp. 29–38.

Jensen, Carlos; Scott King; and Victor Kuechler (2011). Joining Free/Open Source Software Communities: An Analysis of Newbies' First Interactions on Project Mailing Lists. In: *HICSS'10. 44th Hawaii International Conference on System Sciences*, *Kauai, HI*. IEEE, pp. 1–10.

Jepsen, Leif Obel; Lars Mathiassen; and Peter Axel Nielsen (1998). Using Diaries, In: L. Mathiassen (ed.): *Reflective Systems Development*. Aalborg University, Chapt. 3.

Jergensen, Corey; Anita Sarma; and Patrick Wagstrom (2011). The Onion Patch: Migration in Open Source Ecosystems. In: *ESEC/FSE'11. Proceedings of the 19th ACM SIGSOFT Symposium and the 13th European Conference on Foundations of Software Engineering*, *Szeged, Hungary, 5-9 September 2011*. ACM: New York, NY, USA, pp. 70–80.

Jergensen, Niels (2007). Developer autonomy in the FreeBSD open source project. *Journal of Management and Governance*, vol. 11, no. 2, pp. 119–128.

Ji, Yong Gu; Hwan Hwangbo; Ji Soo Yi; P. L. Patrick Rau; Xiaowen Fang; and Chen Ling (2010). The Influence of Cultural Differences on the Use of Social Network Services and the Formation of Social Capital. *International Journal of Human-Computer Interaction*, vol. 26, no. 11–12, pp. 1100–1121.

Joyce, Elisabeth; and Robert E. Kraut (2006). Predicting Continued Participation In Newsgroups. *Journal of Computer-Mediated Communication*, vol. 11, no. 3, pp. 723–747.

Karumur, Raghav Pavan; Tien T. Nguyen; and Joseph A. Konstan (2016). Early Activity Diversity: Assessing Newcomer Retention from First-Session Activity. In: *CSCW'16. 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, *San Francisco, California, USA, 27 February - 2 March 2016*. ACM: New York, NY, USA, pp. 595–608.

Ke, Weiling; and Ping Zhang (2010). The Effects of Extrinsic Motivations and Satisfaction in Open Source Software Development. *Journal of the Association for Information Systems*, vol. 11, no. 12, pp. 784–808.

Koh, Joon; Young-Gul Kim; Brian Butler; and Gee-Woo Bock (2007). Encouraging Participation in Virtual Communities. *Communications of the ACM*, vol. 50, no. 2, pp. 68–73.

Kraut, Robert E.; and Paul Resnick (2012). *Building Successful Online Communities: Evidence-Based Social Design*. The MIT Press.

Lakhani, Karim R.; and Robert G. Wolf (2005), *Perspectives on Free and Open Source Software*, Chapt. Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects, pp. 1–22. Cambridge, Mass.: The MIT Press.

Lampe, Cliff; and Erik Johnston (2005). Follow the (Slash) Dot: Effects of Feedback on New Members in an Online Community. In: *GROUP'05. 2005 International ACM SIGGROUP Conference on Supporting Group Work*, *Sanibel Island, Florida, USA, 6-9 November 2005*. ACM: New York, NY, USA, pp. 11–20.

Lave, Jean; and Etienne Wenger (1991). *Situated Learning : Legitimate Peripheral Participation*. Cambridge University Press, 1 edition.

Lehman, Meir M. (1996). Laws of Software Evolution Revisited. In: *EWSPT'96. 5th European Workshop on Software Process Technology*. Springer: Berlin, Heidelberg, pp. 108–124.

Musicant, David R.; Yuqing Ren; James A. Johnson; and John Riedl (2011). Mentoring in Wikipedia: A Clash of Cultures. In: *WikiSym'11. 7th International Symposium on Wikis and Open Collaboration*, *Mountain View, California, 3-5 October 2011*. ACM: New York, NY, USA, pp. 173–182.

Nakakoji, Kumiyo; Yasuhiro Yamamoto; Yoshiyuki Nishinaka; Kouichi Kishida; and Yunwen Ye (2002). Evolution Patterns of Open-source Software Systems and Communities. In: *IWPSE'02. International Workshop on Principles of Software Evolution*, *Orlando, Florida, 19-20 May 2002*. ACM: New York, NY, USA, pp. 76–85.

Naur, Peter (1983). Psychology of Computer Use, In: T. R. G. Green, S. J. Payne, and G. C. van der Veer (eds.): *Psychology of Computer Use*. London: Academic Press, Chapt. Program development studies based on diaries, pp. 159–170.

Nguyen, Duyen T.; and Susan R. Fussell (2013). Effect of Message Content on Communication Processes in Intercultural and Same-culture Instant Messaging Conversations. In: *CSCW'13. 2013 Conference on Computer Supported Cooperative Work*, *San Antonio, Texas, USA, 23-27 February 2013*. ACM: New York, NY, USA, pp. 19–32.

Opdenakker, Raymond (2006). Advantages and Disadvantages of Four Interview Techniques in Qualitative Research. *Forum: Qualitative Social Research*, vol. 7, no. 4, pp. A11.

Oreg, Shaul; and Oded Nov (2008). Exploring motivations for contributing to open source initiatives: The roles of contribution context and personal values. *Computers in Human Behavior*, vol. 24, no. 5, pp. 2055–2073.

Ostrom, Elinor (2000). Collective action and the evolution of social norms. *Journal of economic perspectives*, vol. 14, no. 3, pp. 137–158.

Palen, Leysia; and Marilyn Salzman (2002). Voice-mail Diary Studies for Naturalistic Data Capture Under Mobile Conditions. In: *CSCW'02. Proceedings of the 2002 ACM Conference on Computer Supported Cooperative Work*, *New Orleans, Louisiana, USA, 16-20 November 2002*. ACM: New York, NY, USA, pp. 87–95.

Parra, Esteban; Sonia Haiduc; and Rebecca James (2016). Making a Difference: An Overview of Humanitarian Free Open Source Systems. In: *ICSE'16. 38th International Conference on Software Engineering Companion*, *Austin, Texas, 14-22 May 2016*. ACM: New York, NY, USA, pp. 731–733.

Pham, Raphael; Leif Singer; Olga Liskin; Fernando Figueira Filho; and Kurt Schneider (2013). Creating a Shared Understanding of Testing Culture on a Social Coding Site. In: *ICSE'13. 2013 International Conference on Software Engineering*, *San Francisco, CA, USA, 18-26 May 2013*. IEEE Press: Piscataway, NJ, USA, pp. 112–121.

Pinto, Gustavo; Igor Steinmacher; Luiz Felipe Dias; and Marco Aurelio Gerosa (2018). On the Challenges of Open-Sourcing ProprietarySoftware Projects. *Empirical Software Engineering*, vol. PP.

Pinto, Gustavo; Igor Steinmacher; and Marco Aurélio Gerosa (2016). More Common Than You Think: An In-depth Study of Casual Contributors. In: *SANER 2016. IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering*, *Suita, Osaka, Japan, 14-18 March 2016*. pp. 112–123.

Preece, Jenny (2001). Sociability and usability in online communities: Determining and measuring success. *Behaviour and Information Technology*, vol. 20, no. 5, pp. 347–356.

Preece, Jenny (2004). Etiquette Online: From Nice to Necessary. *Communications of the ACM*, vol. 47, no. 4, pp. 56–61.

Preece, Jenny; Blair Nonnecke; and Dorine Andrews (2004). The top five reasons for lurking: improving community experiences for everyone. *Computers in Human Behavior*, vol. 20, no. 2, pp. 201–223.

Qureshi, I.; and Y. Fang (2011). Socialization in Open Source Software Projects: A Growth Mixture Modeling Approach. *Organizational Research Methods*, vol. 14, no. 1, pp. 208–238.

Riehle, Dirk (2015). How Open Source Is Changing the Software Developer's Career. *IEEE Computer*, vol. 48, no. 5, pp. 51–57.

Roberts, Jeffrey A.; Il-Horn Hann; and Sandra A. Slaughter (2006). Understanding the Motivations, Participation, and Performance of Open Source Software Developers: A Longitudinal Study of the Apache Projects. *Management Science*, vol. 52, no. 7, pp. 984–999.

Schilling, Andreas; Sven Laumer; and Tim Weitzel (2012). Who Will Remain? An Evaluation of Actual Person-Job and Person-Team Fit to Predict Developer Retention in FLOSS Projects. In: *HICSS'12. 2012 45th Hawaii International Conference on System Sciences*, *Maui, HI, USA, 4-7 January 2012*. IEEE Computer Society: Washington, DC, USA, pp. 3446–3455.

Seaman, Carolyn B. (1999). Qualitative methods in empirical studies of software engineering. *IEEE Transactions on Software Engineering*, vol. 25, no. 4, pp. 557–572.

Shah, Sonali K. (2006). Motivation, Governance, and the Viability of Hybrid Forms in Open Source Software Development. *Management Science*, vol. 52, no. 7, pp. 1000–1014.

Singh, Vandana (2012). Newcomer Integration and Learning in Technical Support Communities for Open Source Software. In: *GROUP'12. Proceedings of the 17th ACM International Conference on Supporting Group Work*, *Sanibel Island, Florida, USA, 27-31 October 2012*. ACM: New York, NY, USA, pp. 65–74.

Singh, Vandana; and Lila Holt (2013). Learning and best practices for learning in open-source software communities. *Computers & Education*, vol. 63 pp. 98–108.

Smolander, Kari; Matti Rossi; and Sandeep Purao (2008). Software architectures: Blueprint, literature, language or decision? *European Journal of Information Systems*, vol. 17, no. 6, pp. 575–588.

Steinmacher, Igor; Ana Paula Chaves; Tayana Conte; and Marco Aurélio Gerosa (2014). Preliminary empirical identification of barriers faced by newcomers to Open Source Software projects. In: *SBES'14. 28th Brazilian Symposium on Software Engineering*, *Maceio, AL, Brazil*. IEEE Computer Society, pp. 1–10.

Steinmacher, Igor; Ana Paula Chaves; and Marco Aurélio Gerosa (2013)a. Awareness Support in Distributed Software Development: A Systematic Review and Mapping of the Literature. *Computer Supported Cooperative Work (CSCW)*, vol. 22, no. 2-3, pp. 113–158.

Steinmacher, Igor; Gregorio Robles; Brian Fitzgerald; and Anthony Wasserman (2017). Free and open source software development: the end of the teenage years. *Journal of Internet Services and Applications*, vol. 8, no. 1, pp. 17.

Steinmacher, Igor; Igor Scaliante Wiese; Ana Paula Chaves; and Marco Aurélio Gerosa (2013)b. Why do newcomers abandon open source software projects? In: *CHASE'13. 2013 6th International Workshop on Cooperative and Human Aspects of Software Engineering*, *San Francisco, CA, USA, 23 May 2013*. IEEE Computer Society: Washington, DC, USA, pp. 25–32.

Steinmacher, Igor; Marco Aurélio Graciotto Silva; Marco Aurélio Gerosa; and David F. Redmiles (2015)a. A systematic literature review on the barriers faced by newcomers to open source software projects. *Information and Software Technology*, vol. 59 pp. 67–85.

Steinmacher, Igor; Tayana Conte; Marco Aurélio Gerosa; and David Redmiles (2015)b. Social Barriers Faced by Newcomers Placing Their First Contribution in Open Source Software Projects. In: *CSCW'15. Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, *Vancouver, BC, Canada, 14-18 March 2015*. ACM: New York, NY, USA, pp. 1379–1392.

Steinmacher, Igor; Tayana Uchoa Conte; Christoph Treude; and Marco Aurélio Gerosa (2016). Overcoming Open Source Project Entry Barriers with a Portal for Newcomers. In: *ICSE'16. Proceedings of the 38th International Conference on Software Engineering*, *Austin, Texas, 14-22 May 2016*. ACM: New York, NY, USA, pp. 273–284.

Stol, Klaas-Jan; Paris Avgeriou; and Muhammad Ali Babar (2010). Identifying architectural patterns used in open source software: approaches and challenges. In: *EASE'10. 14th International conference on Evaluation and Assessment in Software Engineering*, *UK, 12-13 April 2010*. British Computer Society: Swinton, UK, pp. 91–100.

Strauss, Anselm; and Juliet M. Corbin (2007). *Basics of Qualitative Research : Techniques and Procedures for Developing Grounded Theory*. SAGE Publications, 3rd edition.

Suh, Bongwon; Gregorio Convertino; Ed H. Chi; and Peter Pirolli (2009). The Singularity is Not Near: Slowing Growth of Wikipedia. In: *WikiSym'09. 5th International Symposium on Wikis and Open Collaboration*, *Orlando, Florida, 25-27 October 2009*. ACM: New York, NY, USA, pp. 8:1–8:10.

Symon, Gillian (2004). In: C. Cassell and G. Symon (eds.): *Essential Guide to Qualitative Methods in Organizational Research*. SAGE publications, Chapt. Qualitative research diaries, pp. 98–113.

Thompson, Leigh; and Gary Alan Fine (1999). Socially shared cognition, affect, and behavior: A review and integration. *Personality and Social Psychology Review*, vol. 3, no. 4, pp. 278–302.

Treude, Christoph; and Margaret-Anne Storey (2010). Awareness 2.0: staying aware of projects, developers and tasks using dashboards and feeds. In: *ICSE'10. 32nd ACM/IEEE International Conference on Software Engineering*, *Cape Town, South Africa, 1-8 May 2010*. ACM: New York, NY, USA, pp. 365–374.

Tsay, Jason; Laura Dabbish; and James Herbsleb (2014). Influence of Social and Technical Factors for Evaluating Contribution in GitHub. In: *ICSE 2014. Proceedings of the 36th International Conference on Software Engineering*, *Hyderabad, India, 31 May - 7 June 2014*. ACM: New York, NY, USA, pp. 356–366.

Tsvetkova, Milena; Ruth García-Gavilanes; Luciano Floridi; and Taha Yasseri (2017). Even good bots fight: The case of Wikipedia. *PLoS ONE*, vol. 12, no. 2, pp. 1–13.

Wang, Jianguo; and Anita Sarma (2011). Which bug should I fix: helping new developers onboard a new project. In: *CHASE'11. 4th International Workshop on Cooperative and Human Aspects of Software Engineering*, *Waikiki, Honolulu, HI, USA, 21 May 2011*. ACM: New York, NY, USA, pp. 76–79.

Wang, Loxley Sijia; Jilin Chen; Yuqing Ren; and John Riedl (2012)a. Searching for the Goldilocks Zone: Trade-offs in Managing Online Volunteer Groups. In: *CSCW'12. ACM 2012 Conference on Computer Supported Cooperative Work*, *Seattle, Washington, USA, 11-15 February 2012*. ACM: New York, NY, USA, pp. 989–998.

Wang, Yi-Chia; Robert Kraut; and John M. Levine (2012)b. To Stay or Leave?: The Relationship of Emotional and Informational Support to Commitment in Online Health Support Groups. In: *CSCW'12. ACM 2012 Conference on Computer Supported Cooperative Work*, *Seattle, Washington, USA, 11-15 February 2012*. ACM: New York, NY, USA, pp. 833–842.

Wenger, Etienne; Nancy White; and John D Smith (2009). *Digital habitats: Stewarding technology for communities*. CPsquare.

Wolff-Marting, Vincent; Christoph Hannebauer; and Volker Gruhn (2013). Patterns for tearing down contribution barriers to FLOSS projects. In: *SoMeT'13. Proceedings of the 12th International Conference on Intelligent Software Methodologies, Tools and Techniques*, *Budapest, Hungary, 22-24 September 2013*. IEEE, pp. 9–14.

Yang, Diyi; Robert Kraut; and John M. Levine (2017). Commitment of Newcomers and Old-timers to Online Health Support Communities. In: *CHI'17. Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, *Denver, Colorado, USA, 6-11 May 2017*. ACM: New York, NY, USA, pp. 6363–6375.

Ye, Yunwen; and Kouichi Kishida (2003). Toward an Understanding of the Motivation Open Source Software Developers. In: *ICSE'03. 25th International Conference on Software Engineering*, *Portland, Oregon, 3-10 May 2003*. IEEE Computer Society: Washington, DC, USA, pp. 419–429.

Yu, Yue; Huaimin Wang; Vladimir Filkov; Premkumar Devanbu; and Bogdan Vasilescu (2015). Wait for It: Determinants of Pull Request Evaluation Latency on GitHub. In: *MSR'15. Proceedings of the 12th Working Conference on Mining Software Repositories*, *Florence, Italy, 16-24 May 2015*. IEEE Press: Piscataway, NJ, USA, pp. 367–371.

Zhou, Minghui; and Audris Mockus (2012). What Make Long Term Contributors: Willingness and Opportunity in OSS Community. In: *ICSE'12. Proceedings of the*

34th International Conference on Software Engineering, *Zurich, Switzerland, 2-9 June 2012*. IEEE Press: Piscataway, NJ, USA, pp. 518–528.

Zhu, Haiyi; Amy Zhang; Jiping He; Robert E. Kraut; and Aniket Kittur (2013). Effects of Peer Feedback on Contribution: A Field Experiment in Wikipedia. In: *CHI'13. SIGCHI Conference on Human Factors in Computing Systems*, *Paris, France*. ACM: New York, NY, USA, pp. 2253–2262.

Zhu, Haiyi; Robert Kraut; and Aniket Kittur (2012). Effectiveness of Shared Leadership in Online Communities. In: *CSCW'12. ACM 2012 Conference on Computer Supported Cooperative Work*, *Seattle, Washington, USA, 11-15 February 2012*. ACM: New York, NY, USA, pp. 407–416.

von Krogh, Georg; Stefan Haefliger; Sebastian Spaeth; and Martin W. Wallin (2012). Carrots and Rainbows: Motivation and Social Practice in Open Source Software Development. *MIS Quarterly*, vol. 36, no. 2, pp. 649–676.

von Krogh, Georg; and Eric von Hippel (2003). Editorial: Special issue on open source software development. *Research Policy*, vol. 32, no. 7, pp. 1149–1157.