Increasing the self-efficacy of newcomers to Open Source Software projects

Igor Steinmacher, Igor Wiese
Department of Computing (DACOM)
Federal University of Technology, Paraná (UTFPR)
Campo Mourao, PR, Brazil
{igorfs, igor}@utfpr.edu.br

Tayana Uchoa Conte
Institute of Computing (IComp)
Federal University of Amazonas (UFAM)
Manaus, AM, Brazil
tayana@icomp.ufam.br

Marco Aurelio Gerosa
Institute of Mathematics and Statistics (IME)
University of Sao Paulo (USP)
Sao Paulo, SP, Brazil
gerosa@ime.usp.br

***Abstract*— Community-based Open Source Software (OSS) projects are usually self-organized and dynamic, receiving contributions from distributed volunteers. These communities' survival, long-term success, and continuity demand a constant influx of newcomers. However, newcomers face many barriers when making their first contribution to an OSS project, leading in many cases to dropouts. Therefore, a major challenge for OSS projects is to provide ways to support newcomers during their first contribution. In this paper, our goal was to evaluate how the newcomers' perceived efficacy is influenced by the use of an environment that organizes the project information for developers who want to place their first contribution to an OSS project. To accomplish this goal, we created FLOSScoach, a portal aiming to support newcomers to OSS projects, which was implemented based on a model of barriers proposed in previous research. Then, we conducted a study, in which 46 students, split in case and control group, were asked to contribute to an OSS project. We assessed the newcomers' self-efficacy by conducting a questionnaire before and after the assignment. We found that presenting the information according to the model of barriers had a positive influence on newcomers' self-efficacy, empowered the newcomers, making them more confident and comfortable during the contribution process. However, there is also some indication that FLOSScoach did not lower the technical barriers.**

Keywords—*newcomers, open source, self-efficacy, contribution barriers, onboarding*

I. INTRODUCTION

Open Source Software (OSS) projects have risen to great prominence within the last several years [1]. In OSS projects, the source code is licensed to make it freely available to anyone who wishes examine it or change it for their own purposes, redistribute copies and to run the program.

Many OSS projects leverage contributions from geographically distributed volunteers and require a continuous influx of newcomers for their survival, long-term success, and continuity. According to Qureshi and Fang [2], it is essential to motivate, engage, and retain new developers in a project in order to promote a sustainable number of developers. Furthermore, some studies report that newcomers are a source of innovation for new ideas and work procedures that the group needs [3].

However, newcomers usually face many difficulties when making their first contribution to a project. OSS project newcomers are usually expected to learn about the project on their own [4]. Dagenais et al. [5] compare them to explorers in a hostile environment who need to orient themselves. Thus, a major challenge for OSS projects is providing newcomer support.

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 [6–10]. An understudied aspect of the OSS joining process is what happens during the period after a newcomer decides to participate and before their first code contribution is accepted and included in the shared repository. This period is particularly relevant to OSS projects, as many newcomers do not want to join or remain at the project, only to post a single contribution (e.g., a bug correction or a new feature). What happens in this period affects, for example, students in computer courses whose assignments include OSS project contribution, and professional developers who find a bug or wish to customize a particular software product. During this learning period, newcomers face barriers that can result in their decision to give up contributing. Thus, as Karl Fogel [11] states, "if a project doesn't make a good first impression, newcomers may wait a long time before giving it a second chance."

In a previous work by Steinmacher et al. [12], [13], a preliminary barriers model was proposed to help identifying and better understanding the barriers faced by newcomers to OSS. The 58 barriers presented in the model, are organized in 6 different categories: cultural differences, newcomers' characteristics, reception issues, newcomers' orientation, technical hurdles, and documentation problems.

Our goal in this paper is to evaluate how the newcomers' self-perceived efficacy is influenced by the use of an environment that organizes the project information for developers who want to place their first contribution to an OSS project. To achieve this goal we proposed FLOSScoach, a portal to support the first contributions of newcomers to OSS projects. The portal was built based on already existing information and strategies of 6 OSS projects (Amarok, Audacity, Empathy, JabRef, LibreOffice and Vim) and organized according to the model of barriers proposed previously [12]. To assess the influence of FLOSScoach in newcomers' self-efficacy, we conducted a study with students, and applied a self-efficacy instrument [14] before and after they attempted to contribute to an OSS project.

The rest of this paper is organized as follows. In Section II, we present the related work. In Section III, we present FLOSScoach, a portal conceived to support newcomers to OSS. In Section IV we present the research method followed in this study, while in Section V we bring the results. Threats to validity are presented in Section VI and, in Section VII, we present the conclusions.

II. Related Work

Newcomers' onboarding is not an issue exclusively faced by OSS. Many studies in the literature deal with newcomers joining process in collective production communities, including studies on Wikipedia [15], [16] and on OSS projects [17–21]. Dagenais et al. [5] and Begel and Simon [22] present studies regarding newcomers joining process in software projects, but their focus is in industrial settings.

Von Krogh et al. [20] analyzed interviews with developers, emails, source code repository, and documents of the FreeNet project. The authors proposed a joining script for developers who want to take part in the project. Nakakoji et al. [23] studied four OSS projects to analyze the evolution of their communities. They presented eight possible roles for the community members and structured them into a model composed of concentric layers, like the layers of an onion. Although these papers deal with the evolution of members' participation in OSS communities, they focus on newcomers after the onboarding.

Some researchers tried to understand the barriers that influence the retention of newcomers. Zhou and Mockus [10] worked on identifying the newcomers who are more likely to remain in the project in order to offer active support for them to become long-term contributors. Jensen et al. [18] analyzed mailing lists of OSS projects to verify if the emails sent by newcomers are quickly answered, if gender and nationality influence the kind of answer received, and if the reception of newcomers is different in users and developers lists. Steinmacher et al. [24] used data from mailing list and issue tracker to study how reception influences the retention of newcomers in an OSS project.

There are also some studies presenting tools to support newcomers' first steps. Čubranić et al. [25] presented Hipikat, a tool that supports newcomers by building a group memory and recommending source code, mails messages, and bug reports to support newcomers. Wang and Sarma [21] present a Tesseract extension to enable newcomers to identify bugs of interest, resources related to that bug, and visually explore the appropriate socio-technical dependencies for a bug in an interactive manner. Park and Jensen [26] show that visualization tools support the first steps of newcomers in an OSS project, helping them to find information more quickly.

Mentoring is also explored as a way to support newcomers. Malheiros et al. [19] and Canfora et al. [17] proposed different approaches to identify and recommend mentors to newcomers of OSS projects by mining data from mailing lists and source code versioning systems.

As listed, there are some efforts to study newcomers to OSS. However, we could not find any study focused on identifying and organizing the barriers faced by newcomers to OSS. In previous work, we report some results of this research, culminating with the model of barriers for newcomers presented in [12], [13]. The model of barriers presents 58 barriers, organized in 6 categories. It was created based on empirical evidences from qualitative analysis of interviews, questionnaires and feedback from practitioners from different OSS projects.

In this paper, we build upon this model. We created a portal that organizes the information of OSS projects according to the barriers model aiming at supporting newcomers' first steps. Finally, we analyzed how this organization influenced newcomers' self-perceived efficacy.

III. FLOSScoach: A Portal to Support Newcomers to OSS Projects

We developed the FLOSScoach[1] portal based on the barriers model presented in [12], and on information collected from project members. The mentioned barriers model was composed of the following six categories of barriers:

- **Newcomers' orientation**. Newcomers often face rugged and unfamiliar landscapes when onboarding to an OSS project. They need proper orientation to find their way and correctly make their contributions. Examples of barriers under this category includes difficulty to find a mentor, and poor "How to contribute" available.
- **Newcomers' characteristics**. This category comprises the barriers related to the experience and behavior of the newcomers regarding the project and the way they show their knowledge and interact when joining the projects It includes barriers related to newcomers' behavior and newcomers' previous knowledge.
- **Reception Issues.** This category comprises the barriers related to the interactions that occur between newcomers and the community. A breakdown during these social interactions can lead to demotivation, and even result in newcomers' dropping out. These barriers include not receiving an answer to a message and receiving impolite answers.
- **Cultural differences.** Cultural differences can result in interaction problems. Two barriers were reported in the barriers model: need to be in contact with a real person and messages considered rude.
- **Documentation problems**. This category refers to the need to learn the project's technical and social aspects in order to be able to contribute. The barriers under this category define which documentation problems have been evidenced as possible barriers to newcomers to OSS projects, and include: outdated documentation; unclear code comments; information overload; and lack of documentation.
- **Technical hurdles.** This category consists of the project's technical barriers that arise when newcomers are dealing with the code. All the problems newcomers face when dealing with source code were placed under a single category and split into three subcategories: code/architectural hurdles; change request hurdles; and local environment setup hurdles.

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

As our goal was to organize the information reflecting the categorization of the barriers model, we created a portal in which each section presented information aimed at helping newcomers to overcome the barriers related to a given category. Therefore, the portal building process included two steps:

Collection: we collected already-in-place strategies and information aimed to support newcomers overcoming the identified barriers. The collection step included the analysis of existing data collected during the interviews conducted to identify the barriers [12], [13]. We had included questions related to strategies used by the projects, meaning information that would be useful for newcomers as well as suggested contribution process (steps to be followed by newcomers). As a result, we gathered a set of information and strategies that newcomers could use to overcome or lower specific projects' barriers. Furthermore, to gather more information that could be useful to newcomers, we talked to members of LibreOffice and JabRef. We chose these projects because they presented the highest number of subjects that participated of the previous step of this research [12], [13], and because of the facility in accessing project members to gather more information. We presented the barriers model to them and asked them to point to information that the project had in place to support newcomers overcoming each barrier. In addition, we conducted a manual inspection in the projects' web pages to find other possible information. We found a few other resources, which we sent to the developers via email asking them to confirm that those resources would be useful for newcomers. The output was a list of information mapped into the related barrier categories that the support might help newcomers overcome. The information obtained included links to project pages, ways to access the communication channels, documents generated by the project, videos, list of skills needed, and a suggested contribution process.

Organization: we organized the information gathered by splitting the portal in accordance with the barriers model. After collecting the information, we organized it by splitting the portal into sections and subsections, in accordance with the barriers model. However, while analyzing the structure of the portal, we found that information related to communication was spread across newcomers' behavior and cultural differences categories. Therefore, we decided to merge these solutions into one single category that provided information regarding communication. In Figure 1 we show how the barriers model was mapped into the portal sections.

To illustrate another section of the tool, in Figure 2 we present the "How to Start" section, highlighting the contribution flow created to guide the newcomers. The newcomer can use the flow to access the other sections of the tool once the box that represents the steps is clickable.

After organizing the information of LibreOffice and JabRef, we collected information from other 4 OSS projects: Amarok, Audacity, Empathy and Vim. Differently from the two first projects, the information from these projects was collected manually by the researchers, based on the knowledge acquired previously.

After developing this portal, we started our experimental study with students using FLOSScoach. In the following section, we explore the method and results of such study.

IV. RESEARCH METHOD

The goal of the current research is to verify how the organization of information based on the theoretical model created previously influences newcomers' self-perceived efficacy. Since we are dealing with a software development topic related to newcomers' onboarding, we built a newcomers' portal to apply the model, and to evaluate the influence in a practical scenario.

To guide our research towards this study's objective, we defined the following research question:

Does the use of the portal impact newcomers' self-efficacy?

Based on the research question, we defined the iterative research method. We followed an approach similar to action-research [27], in which the model and the portal could evolve after each iteration. In each iteration, we administered pre- and post- study self-efficacy questionnaires and conducted a debrief session with the participants. In the following subsections, we presented the subjects, the task they had to accomplish and details about self-efficacy model and the post-study feedback.

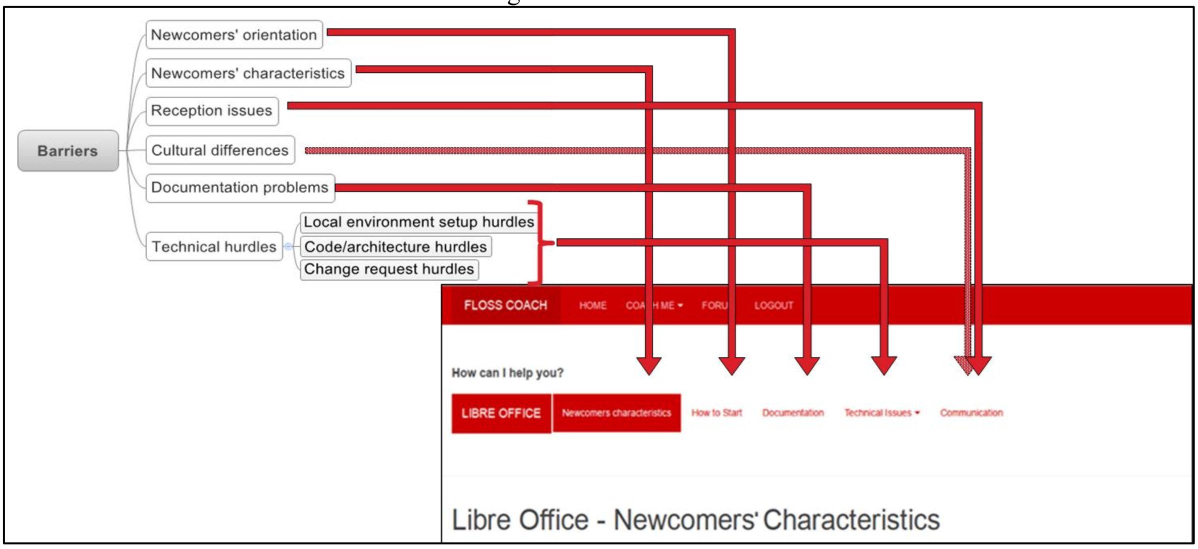

Figure 1. FLOSScoach page with information about newcomers' characteristics for LibreOffice.

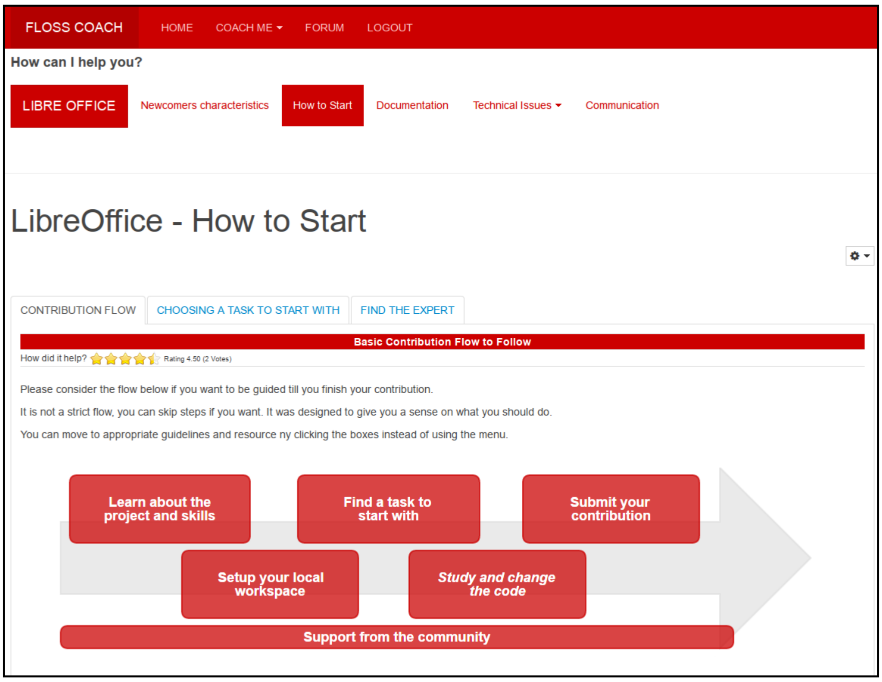

Figure 2. FLOSScoach "How to Start" page presenting the suggested contribution flow.

A. Subjects and assignment

Our subjects were undergraduate students. This population is a sample of potential OSS project contributors, which is why there are currently several programs (for example, GSoC, Facebook Open Academy) focusing on attracting them. Additionally, according to a study by Höst et al. [28], students may provide an adequate model of the professional population and point to a set of benefits that researchers gain from empirical studies. Furthermore, Runeson [29] identified a similar trend when comparing freshman, graduate, and professional developers.

The students chosen for this study had enough knowledge to fix small bugs in software projects and were motivated to contribute (since their grade depended on it). Furthermore, they joined a real project with real issues, and they interacted with the actual code and community.

All the participants received an assignment, requesting them to contribute to a given OSS project. The contribution(s) might be fixing bug(s) or implementing new feature(s) according to what was already reported in project's issues tracker. A contribution was considered complete once the code was accepted by the project members and included in the main trunk of the project. They had one month to deliver the task.

Iteration 1. Students attending a Software Engineering course (3rd year) at Federal University of Technology – Paraná (UTFPR) received a graded assignment. All participants were newcomers to software development in general. Only five of them worked in the industry, but for less than a year. Only one participant presented previous experience in OSS project contributing. We directed the participants to specific OSS projects: LibreOffice and JabRef. These two projects were part of our previous studies, presenting a high number of participants and communities receptive to our research.

We split the participants and directed them to two different projects and assigned the students according to their experience into case (received access credentials to FLOSScoach) and control (did not receive credentials to FLOSScoach).

Iteration 2. It was conducted during a Software Engineering course of a Computer Science major at University of São Paulo (USP). The initial number of participants was 51 students. The profile was a little different from those participants of Iteration 1. The main difference is that the participants had a slightly more industry experience: nine presenting from 1 to 3 and seven presenting 3 to 5 years of experience.

As we were counting on a higher number of newcomers, we assigned the subjects to four other OSS projects: Amarok, Empathy, Vim, and Audacity. We chose projects used by general audience and written in C/C++, to match the participants' skills.

Once again, considering the programming language background and previous experience of the participants, we split them first by project and, then, into case and control groups.

Iteration 2 started with 51 subjects. However, we considered only 32 to analyze the results. We dismissed subjects who: gave up of the assignment; or did not fill out the post-study questionnaire.

We would like to highlight that similar assignment had been applied to evaluate the students in previous editions of the courses where the study has been conducted. Thus, the task is part of the dynamics of the course and will continue to exist in following editions. Moreover, the grades were not related to the contribution itself, but the process followed by the students (reported by means of shared diaries). We emphasized it to all the students before the assignment (as it had been done in the previous editions of the course). The assignment was mandatory, but filling the questionnaires was not. In the surveys welcome page there was a written notice saying that the survey was part of a research, and would be used in a study. It was also written that the students would not be evaluated by their answers and that the survey was not mandatory for the course.

B. Self-efficacy model

To answer the research question, we administered pre- and post- study self-efficacy questionnaires to the subjects, which were quantitatively analyzed. Self-efficacy is a measure of the confidence in the participants' perceived ability to perform a task, which can impact one's actual ability to complete a task [14]. It is correlated with the willingness to stick with a learning task, and has been studied in computer science education [30] and OSS [31] contexts. Based on previous work that applied self-efficacy in OSS research [31], [32], we prepared a questionnaire with 10 items related to self-efficacy of contributing to OSS on a five-point Likert scale ("extremely disagree" (1) to "extremely agree" (5), with an neutral option (3)), as presented in Table 1. The items were elaborated to cover the activities that are related to the barriers identified in the barriers model [12].

Table 1. Items on self-efficacy toward OSS activities.

Sentence
1. I feel comfortable asking for help from the community using electronic communication means
2. I can write my doubts and understand answers in English
3. I am good in understanding code written by other people
4. I have pretty good skills to write and change code
5. I feel comfortable with the process of contributing to an Open Source project
6. I think that contributing to an open source software project is an interesting activity
7. I feel I can set up and run an application if a set of instructions is properly given
8. I am pretty good on searching for solutions and understanding technical issues by myself
9. I can choose an adequate task to fix if a list of tasks is given
10. I can find the piece of code that need to be fixed given a bug report presenting the issue

We asked the participants to answer it immediately before they started their assignment, and also immediately after concluding it. We aimed to discover whether the person had success performing the tasks (resulting in an increase in self-efficacy), or faced unexpected problems or failures (resulting in a decrease in self-efficacy)[33].

C. Post-study feedback

We conducted a quick post-study debriefing session with the participants, aiming to get their feedback regarding the contribution process and the use of FLOSScoach (for the case group). The method used for collecting such data was different for each iteration.

For Iteration 1, we conducted debrief sessions by means of semi-structured interviews, following the script:

- What were your overall impressions of the experience, considering the positive and negative points of the contribution process?
  - How do you feel about the outcome of this activity?
- How was your contact to the community? Did you talk to them? Using which mean?
- What were the main barriers you faced during the process? (Detail the issues)
- What would you suggest to the community to improve the newcomers' experience? (strategies, information, tools, etc.)
- How would you describe the role of FLOSScoach during this process?
  - What are the main benefits and drawbacks of the portal?
  - What are the weaknesses or elements that can be improved in the portal?

We conducted the interviews one day after the students' deadline. The interviews duration ranged from 9 to 16 minutes, with an average of 12 minutes. Afterwards, we transcribed the interviews in order to conduct the qualitative analysis. In Iteration 2, we conducted an online open questionnaire on the day the participants finished the assignment. We used the same questions as in the Iteration 1 script above. The feedback data was analyzed using procedures of Grounded Theory [34].

To verify how the use of FLOSScoach influenced participant's self-efficacy, we analyzed the variation of the pre- and post-study questionnaire answers for each iteration. In the following subsections we present the results of the analysis.

V. RESULTS

In this section we report the results related to the self-efficacy questionnaire and feedback from the subjects. To facilitate reporting our results, in this section we identified the participants as following:

- **CXX_IT1**: participant from Control group, on iteration 1;
- **CXX_IT2**: participant from Control group, on iteration 2;
- **FCXX_IT1**: participant from Case group, on iteration 1;
- **FCXX_IT2**: participant from Case group, on iteration 2.

Before reporting the results, we would like to highlight that, from the initial 65 subjects we considered only 46 for analysis. From the 34 subjects assigned to the control group (not using FLOSScoach), only 19 were considered (56%), as 15 did not complete the assignment (44%). From the 31 subjects assigned to use FLOSScoach 24 were considered (77%), only 4 did not complete the assignment (13%), and 3 decided not to use the portal (10%). This may indicate that FLOSScoach fostered or facilitated the assignment completion.

A. Results for Iteration 1

Figure 3 presents each participant self-efficacy pre- and post-study results. In Figure 3(a), the data is related to the participants who used the portal to contribute. In Figure 3(b), we show the results for the participants who did not have access to the portal. In both figures, we highlight the project to which the participants were asked to contribute (presenting dashed boxes), and we represent the last step the participant reached: (1) workspace setup; (2) finding a task; (3) finding the piece of code to work on; (4) fixing the bug/submitting the fix; (5) fix accepted.

As shown in Figure 3(a), the self-efficacy of six out of seven participants who used the portal during Iteration 1 increased. This shows that most part of the participants finished the study more confident than when they began it. The participant FC6_IT1 is the one that did not made use of the portal, and also presented an increase in self-efficacy. This increase can be explained by his previous experience, which influenced the way he approached the barriers. For participant FC1_IT1, we can see a strict increase.

This can be explained by the fact that this participant was able to very quickly contribute to the project.

Decreases were analyzed in detail, and the self-efficacy score was influenced by the answers to Q6 and Q10. On Q6, the values pre- and post-study were 5 and 3, while for Q10 the values were 4 and 1, respectively. These two questions explain 5 out of the 6 points difference on pre/post analysis. A possible explanation for this is that his self-confidence decreased. During the debrief session, this participant was very upset about his performance, because he was stuck on finding the right artifact to change. He reported spending 2 weeks to find it, and could not complete the assignment, since he was not able to test his changes.

Analyzing Figure 3(b), we can see a slightly different scenario. Most part of the participants that did not have access to the portal (4 out of 7) presented a decrease in self-efficacy. In accordance with the results presented by Davidson et al. [31] in their study of older adults, we attribute the decreases in self-efficacy to the idea that "you don't know what you don't know." If participants experienced unexpected barriers, their self-efficacy significantly decreases.

Notably, two of the participants whose self-efficacy increased were contributing to JabRef (C1_IT1 and C2_IT1). Therefore, all the five participants assigned to contribute to JabRef in Iteration 1 increased their self-efficacy score. We attribute this behavior to the complexity of the project, which exposed the newcomers to lower barriers, mainly related to workspace setup and understanding code.

To better understand the different behaviors observed, Figure 4 shows the median of the answers per question. We can thereby observe trends that encouraged self-efficacy up, and others that pulled it down. First, the scores for questions related to the OSS contribution process (Q5 and Q6) increased for the participants that used the portal, and decreased for those who did not. The decreasing trend observed for Q10 represents the newcomers' difficulty in finding the artifact they need to change in order to work on a selected task, reported by newcomers in both groups. Another interesting behavior was observed for Q8 and Q9 (mostly on Q9), showing their self-confidence to choose a task to work on, even when not using the portal. We did not find any explanation for the variations in Q1 and Q2, since there was a small number of social interactions with the community reported by the participants.

1) Potential enhancements to the portal after Iteration 1

In order to prepare the portal for the second iteration, we analyzed the feedback from the participants. We hoped to find possible suggestions for improving the portal, which we asked for during the debrief session with the participants who used FLOSScoach.

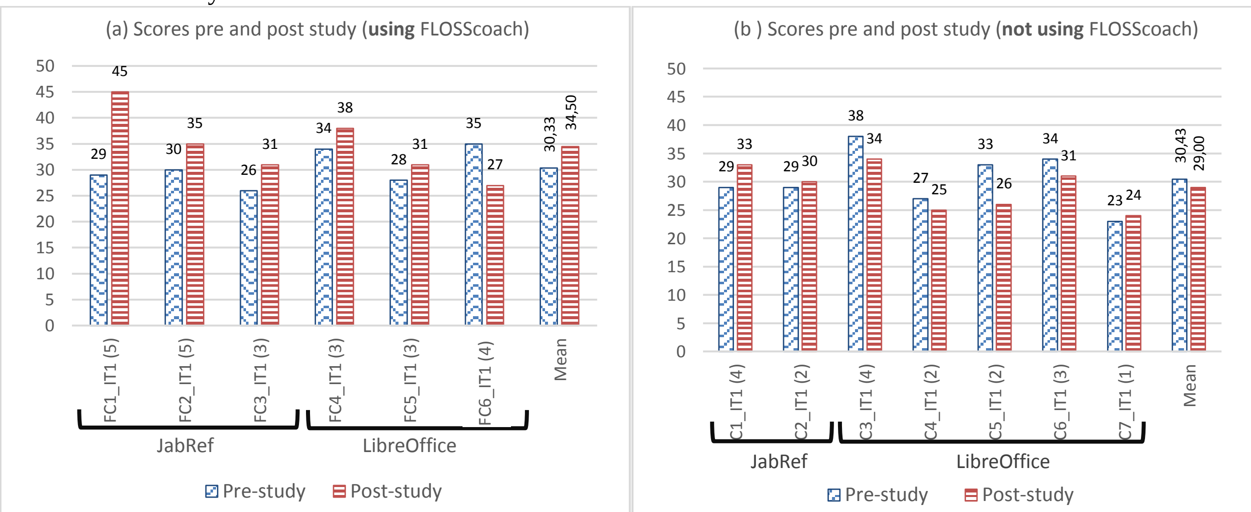

Figure 3. Self-efficacy results per subject - pre and post questionnaires (Iteration 1).

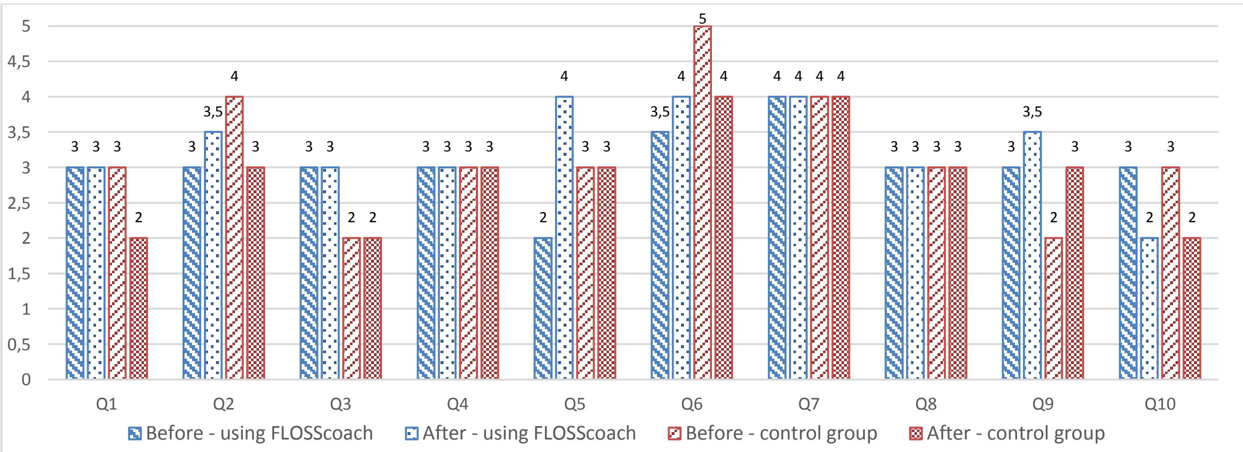

Figure 4. Self-efficacy results per question (Iteration 1).

We found that the most used feature of the portal was the contribution flow, which was shown in a specific page, the "How to Contribute" page, shown in Figure 2. However, some participants felt that it was not handy. They proposed to make the *flow accessible any time*, without having to navigate back it. Two of our participants who used the flow to guide their contribution offered this suggestion.

We received some other suggestions regarding the portal's organization. Two participants questioned 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 providing 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* that was 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.

The participants also suggested some features that might help newcomers choose a task and find the artifacts that they need to change in order fix the issues. Two newcomers suggested the *recommendation of related information* to support their choice. One asked about the possibility of recommending other issues that are related to a given task, enabling newcomers to better understand the issue's context. The second suggestion was to indicate what part of the code newcomers should change to address the issue. These suggestions are very similar to the goal of some proposed initiatives in the literature, like in Wang and Sarma [21] and Hipikat, by Cubranic et al. [25].

We accommodated the suggestions related to user interface and rearranged the portal structure for Iteration 2. The recommendation of related information suggestions were not implemented, but is a possible future direction for this research.

B. Results for Iteration 2

Following the same method as in Iteration 1, we applied the self-efficacy questionnaire before and after the assignment. In Figure 5 and Figure 6, we present the results of self-efficacy pre- and post-study for the group who used the portal and the group who did not use it, respectively. In both figures, we highlight the project to which the participants were asked to contribute (presenting dashed boxes), and we represent the last step the participant reached: (1) workspace setup; (2) finding a task; (3) finding the piece of code to work on; (4) fixing the bug/submitting the fix; (5) fix accepted.

As can be observed in Figure 5, the self-efficacy of 11 participants who used the portal increased, two remained the same, and 5 decreased. Most of the participants finished more confident than when they started the study. It is possible to observe that none of the participants with a decreasing behavior achieved a further step than finding a bug to work with. Interestingly, before and after mean values are very close to each other, possibly because of the huge decreases observed for FC16_IT2, FC17_IT2 and FC18_IT2). Another aspect that appears to be relevant is that all the participants that worked on the Vim project presented a self-efficacy decrease.

We can see three strict decreases (for participants FC16_IT2, FC17_IT2 and FC18_IT2), and two smaller decreases (FC6_IT2 and FC7_IT2). Regarding FC16_IT2's decrease, we found two main axes that contributed to the decrease: social interactions (Q1 and Q2), in which we observed a decrease of 5 points compared to the pre-study; and code issues, contributing to a 7-point decrease. More specifically, for the code issues we observed that the self-efficacy reported in questions Q3, Q4, and Q10 decreased 1, 4, and 2 points, respectively. In his feedback, he reported that he had to change his task three times, trying to understand it and find possible solutions (and complaining about his difficulty in understanding the code).

FC17_IT2 complained that the bugs were not classified by difficulty level and the community reception was poor. He tried to interact with the community, but received no answer. He also highlighted problems in understanding the code as a contribution barrier. Regarding his self-efficacy answers, the decrease behavior spread across all questions. However, the 5 points decrease in the last three questions (-1, -3 and -1, respectively) concerned finding a task and where to address a bug.

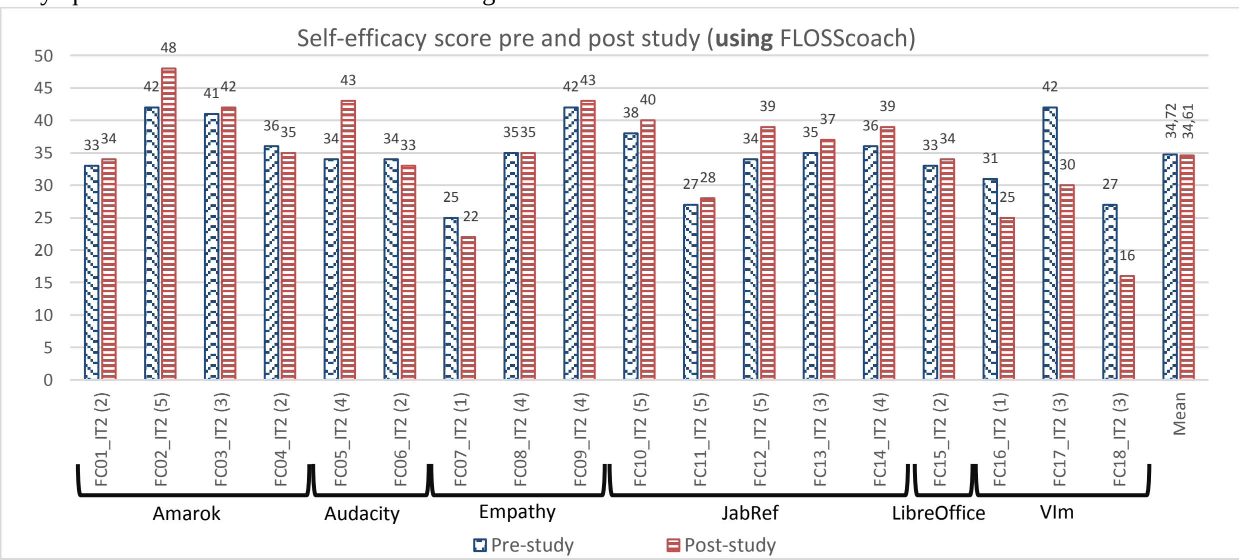

Figure 5. Self-efficacy scores for participants who used FLOSScoach (Iteration 2).

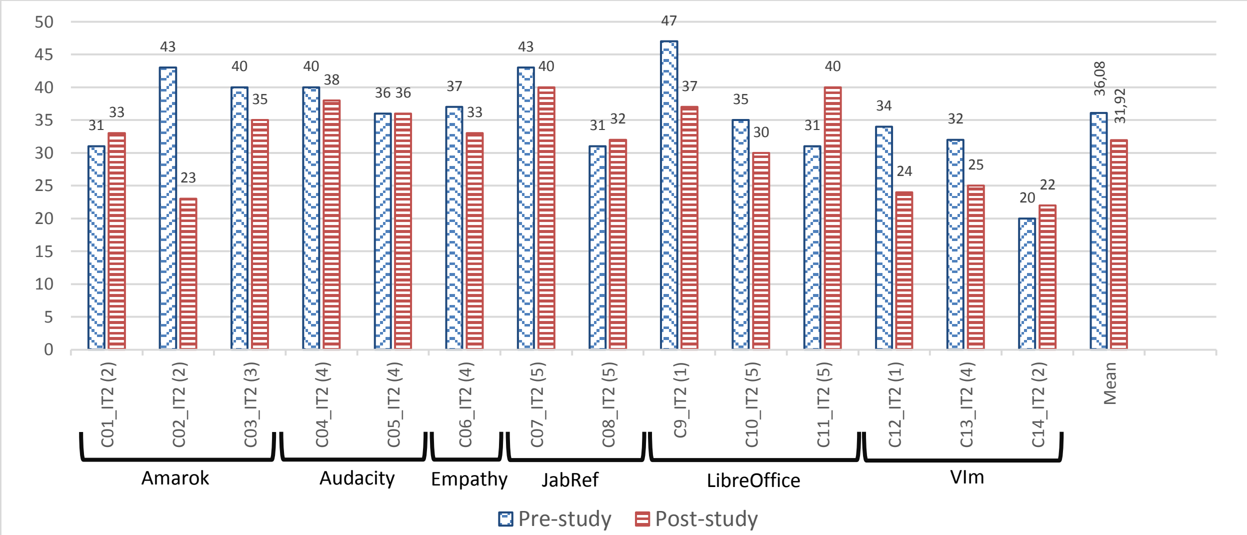

Figure 6. Self-efficacy scores for participants of the control group (Iteration 2).

For FC18_IT2 we noticed in all questions a decrease of 1 or 2 points. We could not find any evidence for this decrease in the debrief questionnaire. We contacted the participant to investigate the reasons. We found that a single cause impacted his score as a whole. In his case, the problem was finding the right artifact to work on. This can be evidenced in the quote: *"As I faced many problems to find the buggy feature, and due to external factors (time), my self-efficacy was negatively impacted."*

For the participants FC6_IT2 and FC7_IT2, we found complaints related to dealing with a large project combined with the short time period, which caused problems in understanding the code. Looking at their self-efficacy results, we found that both presented a strict decrease in the two last questions, which related to finding a task to work with and the right artifact to fix the bug.

For the participants who had no access to FLOSScoach (control group), we observe an opposite behavior (Figure 6). The self-efficacy score of nine participants decreased after the study, whereas only four increased. Among the four participants with an increasing self-efficacy, two finished their assignment with their contribution accepted by the community. We did not find any pattern for the scores of the participants with an increasing score. The most interesting behavior was evidenced for C11_IT2, presenting higher self-efficacy score in 8 out of 10 questions, while the other two scores remained the same as in the pre-study.

Significantly, five other subjects submitted or had their fixes accepted (C4_IT2, C6_IT2, C7_IT2, C10_IT2 and C13_IT2), yet presented a decrease in their self-efficacy scores. Analyzing their answers, we found that there was a decrease for each of these subjects in the code-related questions, as well as in the contribution process questions for three subjects.

Also notable is the large decrease in C2_IT2's score, which drops by 20 points. By checking the differences in each question, we observed that all the scores decreased, but the highest differences were in questions Q5, Q6, and Q7, with a 3-point decrease each. These questions concerned confidence in the contribution process and in workspace setup. This second point was highlighted by the participant in his feedback. We tried contacting the participant to better understand it, but did not receive any answer.

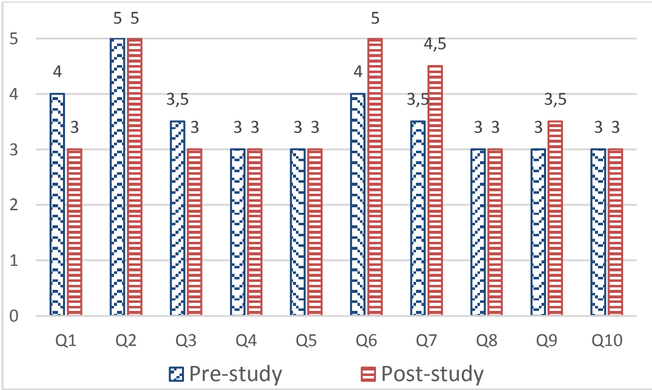

Figure 7. Median of answers per self-efficacy question - participants who used FLOSScoach (Iteration 2).

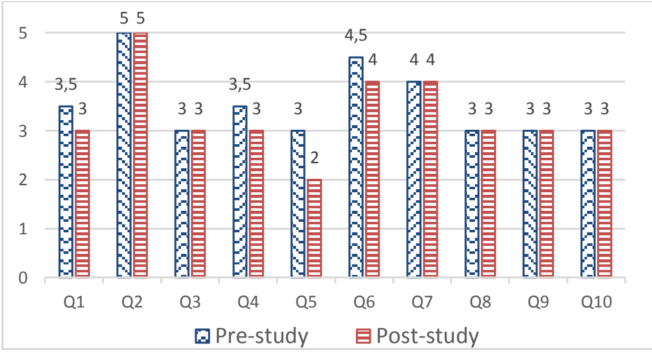

Figure 8. Median of answers per self-efficacy question - participants from control group (Iteration 2).

As in Iteration 1, we presented the median of the answers per question to provide some explanation for the variation of self-efficacy. In Figure 7 and Figure 8, we could observe trends that pushed self-efficacy up and others that pulled it down. First, in consonance with the previous iteration, the scores for questions related to the contribution process (Q5 and Q6) remained equal and increased for the participants that used the portal, whereas it decreased for those who did not. Another increasing trend relates to easiness in finding a task to work with (Q9). The difference is less pronounced, but can indicate a possible facility in finding the right task to work with for the participants using FLOSScoach.

As in Iteration 1, there is some indication of technical problems in the self-efficacy results, however they are less pronounced. This can be observed in the variations of Q3 in Figure 7 and Q4 in Figure 8. There was no evidence of difficulty finding the artifact(s) that needed to be changed to fix a bug, in contrast to Iteration 1's clear difference (refer to Figure 4).

1) Potential enhancements to the portal after Iteration 2

As in Iteration 1, we analyzed the data we gathered, searching for suggestions to enhance the portal. There was no suggestion or complaint that required changes in the barriers model.

Two participants suggested ways to help choosing a task to start with, and in consonance with the Iteration 1, two newcomers recommended providing what part of the code the newcomers should alter to address the issue. Other participants suggested some changes and new features related to making the portal a more collaborative environment.

Iteration 2's suggestions did not affect the barriers model; however, they are valuable ideas for future work.

VI. Threats to Validity

As in any empirical study, we need to discuss the limitations of the study. The perceived efficacy measures are based on self-reported questionnaire items as opposed to objectively measured ones. However, we calculated the Cronbach's Alpha for our sample, which indicated that the questionnaire items are reliable, exceeding the threshold level of 0.7 [35] (see Table 2).

Table 2. Cronbach's alpha.

	Cronbach's alpha	
	Control group	**Case group**
Before the assignment	0.8341	0.8118
After the assignment	0.7395	0.8796

Regarding our questionnaire, we acknowledge that we did not cover all the possible activities that a newcomer need to perform or barriers they need to overcome. Moreover, the questionnaire was not validated by experts, what could have revealed inconsistencies. However, we built an instrument aiming to cover the categories presented in the model of barriers, to assess different dimensions of the process. Thus, our results are valid for the set of categories covered by the questions.

Another threat to validity is our use of students, which affects the results' generalization. Most of our subjects 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, as mentioned before, Höst et al. [28] found that students perform the same as professionals on small tasks of judgment. Moreover, students are potential OSS project contributors, being a sample of the actual newcomers' population.

Additionally, our students may have felt that they needed to provide positive feedback on the surveys (especially evaluating FLOSScoach). Although we emphasized that their answers would reflect their actual perceptions, it may still have been uncomfortable for students to criticize work that was known to relate to faculty research.

Regarding the possible benefit received by students that had been assigned to use FLOSScoach, we inform that even students that were part of the control group and were not able to contribute could receive an excellent grade. We can highlight that the median grade obtained by the control group was 7.0, while the median of those from the experimental group was 7.25, and we did not find statistical difference between both means applying Mann-Whitney test (p=0.3418).

Although we used data from a variety of projects, the findings are not generalizable to all projects, nor can we provide full explanations for the lowered barriers. The results could be influenced by projects' characteristics (e.g., project size, community size, code complexity, programming language). This can be further investigated in future work.

It is likely that there are problems with the use of FLOSScoach that we did not uncover here; and there are likely other barriers that can be lowered when different individuals use the portal. We are aware that each project has its singularities and that the OSS universe is huge, meaning the level of support and the barriers can differ according to the project. Our strategy of considering different projects aimed to explore different ways to use the portal and overcome barriers.

VII. Conclusions

In this paper, we analyzed how the organization of the information that is relevant for newcomers could benefit new developers' contribution process. To organize such information we used a theoretical model proposed previously to build FLOSScoach, a portal to support newcomers to OSS.

We conducted a self-efficacy study, in which 46 students, split in case and control group, were asked to contribute to an OSS project. We assessed the newcomers' self-efficacy by conducting a questionnaire before and after the assignment. The results were very encouraging, revealing that the use of FLOSScoach had a positive influence on newcomers' self-efficacy, mainly by making them more confident and comfortable during the OSS project contribution process. However, there is also some indication that FLOSScoach did not lower some technical barriers. We conclude that providing the proper signs and maps, potentially empowers the newcomers that are starting their journey, by making them more self-confident.

Regarding the research design, it was interesting to conduct an iterative method. Firstly, it enabled us to improve the portal by getting the feedback from the subjects. Secondly, the analysis of the results obtained with students from different universities attempting to contribute to different OSS projects helped us improving the perception regarding the self-efficacy and about how the portal was used.

The next step of this research is to apply user experience assessment methods to evaluate how newcomers perceive and use FLOSScoach, and conduct other iterations of action research to keep improving the portal. Another interesting future research direction could be to use mining techniques and natural language processing to add and update project information to FLOSScoach. We also aim to have the portal evaluated by real developers. We started contacting some communities, to get it assessed by the community members, so they can indicate FLOSScoach to newcomers. Two newcomers had already used it, and provided a good feedback about the usefulness of the portal.

ACKNOWLEDGEMENTS

The authors would like to thank Fundação Araucária, CNPq (process 477831/2013-3), NAPSoL-PRP-USP, NAWEB, FAPESP and FAPEAM (processes 062.00600/2014; 062.00578/2014; and 01135/2011) for their financial support. Marco Gerosa receives individual grant from CNPq.

REFERENCES

[1] G. von Krogh and E. von Hippel, "Editorial: Special issue on open source software development," *Research Policy*, vol. 32, no. 7, pp. 1149–1157, Jul. 2003.

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

[3] R. E. Kraut, M. Burke, J. Riedl, and P. Resnick, "The Challenges of Dealing with Newcomers," in *Building Successful Online Communities: Evidence-Based Social Design*, R. E. Kraut and P. Resnick, Eds. MIT Press, 2012, pp. 179–230.

[4] W. Scacchi, "Understanding the requirements for developing open source software systems," *IEE Proceedings Software*, vol. 149, no. 1, pp. 24–39, Feb. 2002.

[5] B. Dagenais, H. Ossher, R. K. E. Bellamy, M. P. Robillard, and J. P. de Vries, "Moving into a new software project landscape," in *Proceedings of the 2010 ACM/IEEE 32nd International Conference on Software Engineering*, 2010, vol. 1, pp. 275–284.

[6] A. Hars and S. Ou, "Working for free? Motivations of participating in open source projects," in *Proceedings of the 34th Annual Hawaii International Conference on System Sciences*, 2001, pp. 1–9.

[7] C. Jergensen, A. Sarma, and P. Wagstrom, "The Onion Patch: Migration in Open Source Ecosystems," in *Proceedings of the 19th ACM SIGSOFT Symposium and the 13th European Conf. on Foundations of Software Engineering*, 2011, pp. 70–80.

[8] A. Schilling, S. Laumer, and T. Weitzel, "Who Will Remain? An Evaluation of Actual Person-Job and Person-Team Fit to Predict Developer Retention in FLOSS Projects," in *Proceedings of the 2012 45th Hawaii International Conference on System Sciences*, 2012, pp. 3446–3455.

[9] Y. Ye and K. Kishida, "Toward an Understanding of the Motivation Open Source Software Developers," in *Proceedings of the 25th International Conference on Software Engineering*, 2003, pp. 419–429.

[10] M. Zhou and A. Mockus, "What make long term contributors: Willingness and opportunity in OSS community," in *Proceedings of the 34th International Conference on Software Engineering*, 2012, pp. 518–528.

[11] K. Fogel, *Producing Open Source Software: How to Run a Successful Free Software Project*, First. O'Reilly Media, 2013.

[12] I. Steinmacher, A. P. Chaves, T. Conte, and M. A. Gerosa, "Preliminary empirical identification of barriers faced by newcomers to Open Source Software projects.," in *Proceedings of the 28th Brazilian Symposium on Software Engineering*, 2014, pp. 1–10.

[13] I. Steinmacher, T. Conte, M. A. Gerosa, and D. F. Redmiles, "Social Barriers Faced by Newcomers Placing Their First Contribution in Open Source Software Projects," in *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, 2015, pp. 1–13.

[14] A. Bandura, *Social foundations of thought and action: a social cognitive theory*. Prentice-Hall, 1986.

[15] A. Halfaker, A. Kittur, and J. Riedl, "Don't Bite the Newbies: How Reverts Affect the Quantity and Quality of Wikipedia Work," in *Proceedings of the 7th International Symposium on Wikis and Open Collaboration*, 2011, pp. 163–172.

[16] P. Vora and N. Komura, "The n00b Wikipedia Editing Experience," in *Proceedings of the 6th International Symposium on Wikis and Open Collaboration*, 2010, p. Article 36.

[17] G. Canfora, M. di Penta, R. Oliveto, and S. Panichella, "Who is Going to Mentor Newcomers in Open Source Projects?," in *Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering*, 2012, pp. 44:1–44:11.

[18] C. Jensen, S. King, and V. Kuechler, "Joining Free/Open Source Software Communities: An Analysis of Newbies' First Interactions on Project Mailing Lists," in *Proceedings of the 44th Hawaii International Conference on System Sciences*, 2011, pp. 1–10.

[19] Y. Malheiros, A. Moraes, C. Trindade, and S. Meira, "A Source Code Recommender System to Support Newcomers," in *Proceedings of the IEEE 36th Annual Computer Software and Applications Conference*, 2012, pp. 19–24.

[20] G. von Krogh, S. Spaeth, and K. R. Lakhani, "Community, joining, and specialization in open source software innovation: A case study," *Research Policy*, vol. 32, no. 7, pp. 1217–1241, 2003.

[21] J. Wang and A. Sarma, "Which bug should I fix: helping new developers onboard a new project," in *Proceedings of the 4th International Workshop on Cooperative and Human Aspects of Software Engineering*, 2011, pp. 76–79.

[22] A. Begel and B. Simon, "Novice Software Developers, All over Again," in *Proceedings of the Fourth international Workshop on Computing Education Research*, 2008, pp. 3–14.

[23] K. Nakakoji, Y. Yamamoto, Y. Nishinaka, K. Kishida, and Y. Ye, "Evolution Patterns of Open-source Software Systems and Communities," in *Proceedings of the International Workshop on Principles of Software Evolution*, 2002, pp. 76–85.

[24] I. Steinmacher, I. S. Wiese, A. P. Chaves, and M. A. Gerosa, "Why do newcomers abandon open source software projects?," in *Proceedings of the 2013 6th International Workshop on Cooperative and Human Aspects of Software Engineering*, 2013, pp. 25–32.

[25] D. Cubranic, G. C. Murphy, J. Singer, and K. S. Booth, "Hipikat: a project memory for software development," *IEEE Transactions on Software Engineering*, vol. 31, no. 6, pp. 446–465, Jun. 2005.

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

[27] S. Kemmis, R. McTaggart, and R. Nixon, *The action research planner*. Springer Singapore, 2014.

[28] M. Höst, B. Regnell, and C. Wohlin, "Using Students as Subjects – A Comparative Study of Students and Professionals in Lead-Time Impact Assessment," *Empirical Software Engineering*, vol. 5, no. 3, pp. 201–214, Nov. 2000.

[29] P. Runeson, "Using students as experiment subjects–an analysis on graduate and freshmen student data," in *Proceedings of the 7th International Conference on Empirical Assessment in Software Engineering*, 2003, pp. 95–102.

[30] S. Cassidy and P. Eachus, "Developing the computer user self-efficacy (CUSE) scale: investigating the relationship between computer self-efficacy, gender and experience with computers," *Journal of Educational Computing Research*, vol. 26, no. 2, pp. 133–153, Jan. 2002.

[31] J. L. Davidson, U. A. Mannan, R. Naik, I. Dua, and C. Jensen, "Older Adults and Free/Open Source Software: A Diary Study of First-Time Contributors," in *Proceedings of The International Symposium on Open Collaboration*, 2014, p. A5.

[32] Y. Park, "Supporting the learning process of open source novices : an evaluation of code and project history visualization tools," School of Electrical Engineering and Computer Science - Oregon State University, 2008.

[33] S. A. Smith, S. J. Kass, R. J. Rotunda, and S. K. Schneider, "If at first you don't succeed: Effects of failure on general and task-specific self-efficacy and performance.," *North American Journal of Psychology*, vol. 8, no. 1, pp. 171–182, Apr. 2006.

[34] A. Strauss and J. M. Corbin, *Basics of Qualitative Research : Techniques and Procedures for Developing Grounded Theory*, 3rd ed. SAGE Publications, 2007.

[35] J. C. Nunnaly and I. H. Bernstein, *Psychometric theory*. New York: McGraw-Hill, 1978.