A Theory of the Engagement in Open Source Projects via Summer of Code Programs

Jefferson Silva
silvajo@pucsp.br
Pontifical Catholic Univ. of São Paulo
Brazilian Network Information Center
São Paulo, SP, Brazil

Igor Wiese
igor@utfpr.edu.br
Univ. Tecn. Federal do Paraná
Campo Mourão, PR, Brazil

Daniel M. German
dmg@uvic.ca
University of Victoria
Victoria, BC, Canada

Christoph Treude
christoph.treude@adelaide.edu.au
University of Adelaide
Adelaide, SA, Australia

Marco Aurélio Gerosa
Marco.Gerosa@nau.edu
Northern Arizona University
Flagstaff, AZ, USA

Igor Steinmacher
igor.steinmacher@nau.edu
Northern Arizona University
Flagstaff, AZ, USA

ABSTRACT

Summer of code programs connect students to open source software (OSS) projects, typically during the summer break from school. Analyzing consolidated summer of code programs can reveal how college students, who these programs usually target, can be motivated to participate in OSS, and what onboarding strategies OSS communities adopt to receive these students. In this paper, we study the well-established Google Summer of Code (GSoC) and devise an integrated engagement theory grounded in multiple data sources to explain motivation and onboarding in this context. Our analysis shows that OSS communities employ several strategies for planning and executing student participation, socially integrating the students, and rewarding student's contributions and achievements. Students are motivated by a blend of rewards, which are moderated by external factors. We presented these rewards and the motivation theory to students who had never participated in a summer of code program and collected their shift in motivation after learning about the theory. New students can benefit from the former students' experiences detailed in our results, and OSS stakeholders can leverage both the insight into students' motivations for joining such programs as well as the onboarding strategies we identify to devise actions to attract and retain newcomers.

CCS CONCEPTS

• Software and its engineering → Open source model; **• Human-centered computing → Open source software**.

KEYWORDS

Motivation, Onboarding, Engagement, Mentoring, OSS, Process Theory, Summer of Code

ESEC/FSE '20, November 8–13, 2020, Virtual Event, USA

ACM ISBN 978-1-4503-7043-1/20/11…$15.00
https://doi.org/10.1145/3368089.3409724

ACM Reference Format:
Jefferson Silva, Igor Wiese, Daniel M. German, Christoph Treude, Marco Aurélio Gerosa, and Igor Steinmacher. 2020. A Theory of the Engagement in Open Source Projects via Summer of Code Programs. In *Proceedings of the 28th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE '20), November 8–13, 2020, Virtual Event, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3368089.3409724

1 INTRODUCTION

Motivation to contribute to OSS and onboarding of new developers are often studied in the literature (e.g., [11, 17, 20, 35, 46]). However, these studies focus on software developers in general. Analyzing what motivates students to participate in OSS and how to onboard them is underexplored in the literature. Fostering the participation of students can increase the OSS workforce at the same time that it can benefit students, since potential employers increasingly consider online contributions when making hiring decisions [15, 34].

This paper aims to help understand why students participate in an OSS community through a summer of code (SoC) program. Summer of code programs provide a path towards joining open source projects, connecting projects with new contributors, typically students [42, 51]. Examples of such programs include Google Summer of Code (GSoC),[1] Rails Girls Summer of Code,[2] Julia Summer of Code,[3] and Outreachy.[4] The programs offer a variety of benefits, such as career building, an entry gateway to OSS projects, peer recognition, mentorship, stipends, and intellectual stimulation [43]. Previous work identified outcomes of summer of code programs [50, 51] and student retention [38], usually in a few projects. With this paper, we extend the existing literature by explaining onboarding strategies and motivations to participate in SoC programs. We answer the following research questions:

RQ1: How do OSS projects onboard students participating in summer of code programs?
RQ2: What motivates students to participate in a summer of code program?

[1] https://developers.google.com/open-source/gsoc/
[2] http://railsgirlssummerofcode.org/
[3] https://julialang.org/soc/archive.html
[4] https://www.outreachy.org/

To answer our RQs, we analyzed the well-established Google Summer of Code program and built an integrated theory grounded in multiple data sources: students' and mentors' answers to open-ended questions, student interviews and survey responses, a literature review, and OSS projects' applications to join GSoC. Our engagement theory has two components. The first is a grounded theory that explains the actions OSS projects perform in SoC programs to onboard students. This component can be considered a process theory [32, 33], since it explains how OSS projects adapt. The second component of our engagement theory explains how students are motivated to join and contribute to OSS projects via SoC programs. We identify several aspects that influence students' motivation, such as individual differences, external factors, and participation rewards.

Our theory contributes to enriching the state-of-the-art in several ways: (i) our engagement theory structures the existing knowledge about the understudied phenomenon of engaging in SoC programs [16]; (ii) new students can benefit from the experiences of former SoC participants to learn about motivations to join SoC programs; (iii) OSS projects can leverage the understanding of how the variety of rewards influence participants' motivation and how to onboard students in OSS to devise strategies to attract and retain contributors; (iv) program organizers can better support the involved communities; and (v) finally, our theory offers a foundation for researchers interested in building a *variance theory* [32, 54], which could, for instance, predict the actions that OSS projects need to take to retain students.

2 CONTEXT: GOOGLE SUMMER OF CODE

We study Google Summer of Code (GSoC), which is a worldwide Google program that offers students a stipend to write code for OSS projects for three months. We chose to study GSoC because (i) it is best-known compared to other SoC programs, (ii) it has been in operation for over a decade (since 2005), (iii) a large number of globally-distributed students participate in it, and (iv) it provides students with a comprehensive set of rewards, including participating in a well-known large company's program, community bonding, skill development, personal enjoyment, career advancement, peer recognition, status, and a stipend [51]. Google opens an annual call for proposals aimed at OSS projects interested in participating in the program.

3 RESEARCH DESIGN

We built two separate but interrelated theories (one for each RQ) grounded in multiple data sources, as discussed in the following.

3.1 Phase I: Building the Onboarding Theory

To understand how OSS projects onboard students in SoC programs (RQ1), we searched for data that could show us the strategies that OSS projects adopt to onboard students. OSS projects must submit an application to join GSoC. We used the Google search engine to find application forms that OSS projects made publicly available. Using the questions from the application forms as a search query (e.g., *"How will you keep students involved with your community after GSoC?"*), we were able to collect applications from 88 distinct OSS projects. We analyzed 25 applications randomly selected before reaching saturation of information. The complete list of projects and the documents we analyzed are available in the replication package[5]. We also analyzed the GSoC mentor guide [48], which includes suggestions on how to engage students.

To analyze our data, we used coding, which consists of assigning words or phrases to portions of unstructured data [37]. We followed Charmaz's constructivist approach [8] to divide the process into three steps: (*i*) *initial coding*, (*ii*) *focused coding* and *categorizing*, and (*iii*) *theory building*. As a result, we obtained 34 concepts, 13 categories, and 2 major categories, which we organized to create the onboarding theory (Figure 1), the first component of our engagement theory.

3.2 Phase II: Building the Motivation Theory

To understand what motivates students to participate in a summer of code program (RQ2), we investigated multiple empirical data sources. First, we reanalyzed the data[6] that we collected in a previous work [43], with a focus on theory building. In this phase, we also combined our previous empirical results with relevant literature. We reviewed works that targeted SoC programs and motivation [26, 39, 40, 42, 43, 50, 53, 55].

For all grounded theory procedures in this study, the first author performed the open coding. The next steps involved two other authors, who discussed until reaching mutual agreement.

3.3 Phase III: Perceptions about the Theory

In Phase III, we aimed to understand the effects of presenting our theory to students who had no previous participation in SoC programs.

Data collection. In a questionnaire[7], we asked students whether they had heard of GSoC or similar programs. For the students who had heard of such a program, we asked them to describe it and tell us whether they had considered joining it. Next, we asked students to read an explanation of GSoC (from the program website).[8] After that, we asked them to list, in order of importance, what SoC students gain by participating in GSoC.

Then, we instructed the students to watch a 7.5-minute explanatory video[9] that the authors prepared to summarize the theory. We decided to use an explanatory video instead of text to facilitate the students' participation. Afterward, the students answered a final questionnaire,[10] in which we asked what about their perception of GSoC changed, what they had learned, whether our results would influence their decision to participate, how GSoC contributes (or not) to attracting new contributors to OSS projects, what could motivate other people to contribute to OSS projects through SoCs, what SoC students gain by participating in GSoC in order of importance, and demographics.

Data analysis. To analyze students' answers, we applied descriptive statistics and grounded theory procedures [8]. In response to the question of what students gain by participating in GSoC,

[5] https://figshare.com/s/f5a9f70a82d600b4c949
[6] The data is publicly available at https://figshare.com/s/e0fdcfa581b638bd3ded
[7] https://pt.surveymonkey.com/r/HX9H7FX
[8] See the **What is Google Summer of Code?** section: https://google.github.io/gsocguides/student/
[9] https://figshare.com/s/88704bc89fac722ac073
[10] https://pt.surveymonkey.com/r/HN82XYJ

students provided a list of rewards. We classified the rewards according to Silva et al.'s motivation scheme [43]. We discarded all unclear rewards. For example, when a student listed *"experience,"* we opted to discard it because it was not clear whether it referred to the experience in contributing to OSS projects or experience in the CV (or both). Moreover, we discarded all rewards that could not be classified according to Silva et al.'s motivation scheme [43], such as *"maturity,"* and *"organization."* Although we discarded rewards, we maintained the rewards' rank positions. For example, our analysis of a possible answer that listed *"1. Career building; 2. Maturity, and; 3. Stipends"* would discard *"Maturity,"* but would still rank *"Career building"* and *"Stipends"* as first and third positions, respectively. Thus, to obtain a score for each reward, for all students in our sample (38), we applied the formula: $s = \log_b(\sum_1^{38} b^{(b-r+1)}) \times 100$, where b is the number of possible categories in [43] (i.e., $b = 7$), r is the rank of a reward in an answer, and s is the final score of a reward.

Sampling. As the authors are professors, we invited our students to participate in the survey. We emailed ≈130 survey invitations to Brazilian and Chinese students. The Chinese students were enrolled in an OSS class and received grade incentives to participate. No incentives were offered to Brazilian students.

A total of 41 respondents completed all three steps (18 Brazilian and 23 Chinese). After a preliminary analysis, we observed that some Chinese respondents had already participated in GSoC (2) or a similar program (1). We excluded these answers from our analysis. Therefore, our working sample comprises 38 students (18 Brazilian and 20 Chinese).

4 RESULTS

As previously described, our engagement theory is divided into two interconnected components: onboarding and motivation theory. In this section, we explain these theories, which comprise concepts, categories, and major categories. A category is a group of concepts, and a major category is a group of categories. We present concepts in SMALL CAPITALS, categories in *italics*, and major categories in **boldface**.

4.1 The Onboarding Theory (RQ1)

To obtain empirical data on which strategies OSS projects employ to onboard students in SoCs, we analyzed the OSS projects' applications for GSoC. We found a significant number of strategies that OSS projects propose to engage students (Figure 1). OSS projects' strategies were grouped into four categories. While **planning** and **execution** follow the GSoC timeline, *integration* and *rewarding* can be performed before, during, or after the program.

Planning. As GSoC is competitive, OSS projects are required to carefully plan their participation in the program to increase their odds of selection. Thus, we grouped into this major category the actions that OSS projects do before GSoC kickoff (Figure 1). Although GSoC program administrators advise that the program *"is about building the student's experience"* and that *"getting code in [the] project is a nice side effect"* [48], OSS projects work to establish a contribution context that encourages students toward becoming contributors. As an example, Apache Software Foundation leverages the program to *"draw attention and new talent to many of its projects,"* which *"benefit from contributions and galvanize new community members by mentoring students."*[11]

When applying to GSoC, OSS projects typically start by COLLECTIVELY FORMULATING AN IDEAS LIST, which can also be used to assess the project's strengths and weaknesses and help the community decide to apply for the Summer of Code, as in the case of Debian.

Accepted OSS projects worry about fairness in ranking students' proposals, which leads them to devise and employ *applicants' proposal acceptance criteria*, such as only accepting proposals that were CHECKED BY MENTORS or that contained SOLUTIONS THAT COULD BE REFINED later by other members. Complementarily, some OSS projects employ *students' selection criteria*, deciding to only accept applicants WITH GOOD RELATIONSHIPS WITH POTENTIAL MENTORS and WITH PREVIOUS CONTRIBUTIONS TO CODEBASE. One problem with this strategy is that it can potentially harm underrepresented groups [59].

In several applications, good communication was described as key to successful participation, and several projects define a *communication policy*. We observed three types of communications: student-community, mentor-mentor, and mentor-student. The OSS projects' preferred way of communicating to students is to employ the SAME CHANNEL USED BY OTHER MEMBERS. In some cases, mentors use dedicated MENTORS' COMMUNICATION CHANNELS to talk to more experienced members. GSoC program administrators advise mentors to employ multiple methods in the STUDENTS' COMMUNICATION CHANNELS [48]. Also, a *communication policy* defines the FREQUENCY OF UPDATES students should provide, which was used to not only manage the OSS projects' expectations towards the project's completion, but also to identify student drop-outs [48].

OSS projects employ *mentors' selection criteria* to identify mentors with a good fit for the students. OSS projects define that mentoring should be performed IN PAIRS ONLY, with INEXPERIENCED MENTORS PAIRED WITH EXPERIENCED ONES, ideally with PREVIOUS EXPERIENCE IN GSoC, and performed by KNOWN MEMBERS of the community.

OSS projects may face difficulties in deciding when to accept students' work [48]. GSoC program administrators recommend that OSS projects define *work acceptance criteria* upfront. Some OSS projects define criteria such as accepting code that was MERGED INTO THE CODEBASE ONLY. Additionally, to keep track of students' work, some OSS projects establish MONITORING TOOLS, and a *review process of students' work* such as CODE INSPECTION.

Execution. We grouped into this category the OSS projects' strategies intended to coordinate and mentor students during GSoC. The *mentoring actions* consisted of REVIEWING/TESTING CODE; frequently GIVING FEEDBACK; ENCOURAGING STUDENTS when they are demotivated; IDENTIFYING BARRIERS TO WORK COMPLETION such as checking that students have appropriate working conditions or whether students have enough time to complete the tasks; MANAGING THE OSS PROJECTS' EXPECTATIONS, such as when students should complete the development of a feature; FINDING ALTERNATIVE SOLUTIONS TO PROBLEMS, especially when primary goals cannot be reached, and; INVITING STUDENTS TO TEAM MEETINGS. Moreover, several OSS projects institute *progress monitoring actions*

[11] https://blogs.apache.org/foundation/entry/success-at-apache-google-summer

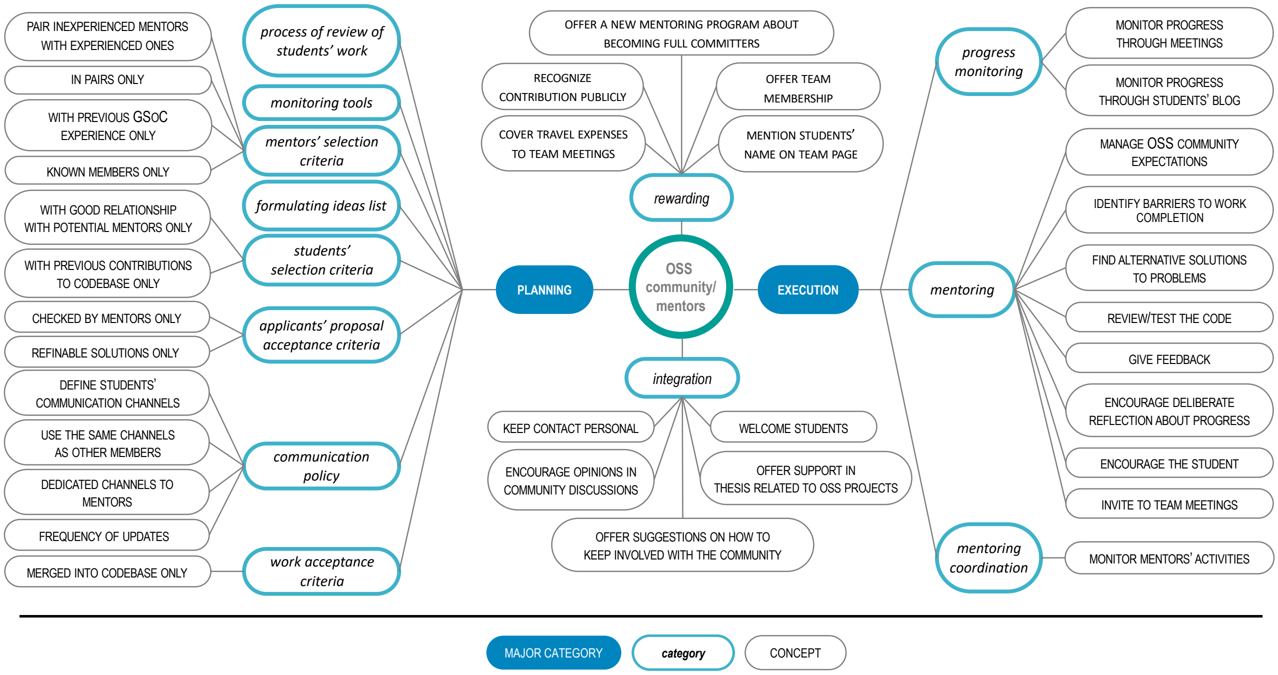


Figure 1: The onboarding theory

such as MONITORING STUDENTS' PROGRESS THROUGH MEETINGS and MONITORING PROGRESS THROUGH STUDENTS' BLOG posts. Furthermore, mentors can face problems during mentoring. Thus, some OSS projects adopt *mentoring coordination actions* such as MONITORING MENTORS' ACTIVITIES as a strategy to reduce the odds of failure.

An interesting approach for keeping students involved in the project during GSoC is to ENCOURAGE DELIBERATE REFLECTION ABOUT PROGRESS. The strategy consists of encouraging students to ponder: *"what is the plan for this day?"*; *"was the plan accomplished?"*; and *"what is the plan for tomorrow?."*

Integration. When applying for GSoC, OSS projects are required to detail their plans to keep students involved during and after the program. Often, the actions OSS projects take aim at integrating students into their social structure. An integration motif present among OSS projects (13) was to WELCOME THE STUDENTS to break the ice. As an OSS project explained: *"(...) we embrace you [the student] warmly and without condition."* Actions to keep the students involved after the program included KEEPING PERSONAL CONTACT, OFFERING STUDENTS SUGGESTIONS ON HOW TO STAY INVOLVED WITH THE PROJECT, and OFFERING SUPPORT IN STUDENT THESES RELATED TO THE OSS PROJECT.

We acknowledge that there may not be a consensus on how to effectively socially integrate students. Furthermore, there may be a classification overlap between the actions in this category and others. For example, during planning, OSS projects may decide to use mentors with excellent social skills. OSS projects may also support students in experiencing what project members regularly do, such as ENCOURAGING OMITTING OPINIONS IN DISCUSSIONS, which would occur during mentoring. Such overlap organically happens due to the cross-cutting nature of integration strategies. Although any action can ultimately be considered an act to integrate students, we grouped the ones that directly aim at diminishing social distance among members. As described in the mentor guide: successful participation in SoC programs depends mainly on the social bonding students create with the community [48].

Rewarding. We grouped into this category the OSS projects' strategies that acknowledge the merit of students' contributions. For example, an OSS project stressed the importance of RECOGNIZING STUDENTS' CONTRIBUTIONS PUBLICLY, especially to other members. Another strategy consists of MENTIONING STUDENTS' NAMES ON THE TEAM PAGE to increase their exposure, both internally and externally, especially to support their careers. Some OSS projects COVER TRAVEL EXPENSES TO TEAM MEETINGS, which aims not only at deepening ties with other members, but also at providing students a chance to network. For students who perform well, OSS projects offer TEAM MEMBERSHIP. Moreover, an OSS project offers a specific MENTORING PROGRAM ABOUT BECOMING A FULL COMMITTER to high achieving students.

Answer for RQ1: To onboard students, OSS projects propose a variety of strategies that go well beyond providing the students with the practical knowledge necessary for contributing to OSS projects. Although strategies differ from project to project, they converged towards planning and executing their participation, socially integrating the students, and rewarding contributions and achievements.

4.2 The Motivation Theory (RQ2)

As aforementioned, we started building the motivation theory based on data that we collected in a previous work [43]. Figure 2 shows the intersection between the motivational theory and the onboarding theory (*OSS community actions to plan, execute, integrate, and reward*). These boxes condense the actions shown in the onboarding theory (Figure 1).

The motivation theory, depicted in Figure 2, describes how a set of participation rewards influences students' interest in contributing to OSS projects via SoC programs. We adopted the construct *reward* because it is frequently used in the psychology literature to refer to what individuals expect to receive in exchange for carrying out a certain behavior [9]. Here, *participation rewards* refer to what students expected to receive when they participated in GSoC for the first time.

We found that some participation rewards refer to motives related to the feelings that their contribution to OSS projects evoked in students such as ENJOYMENT and FUN. Some students reported participating in GSoC for INTELLECTUAL STIMULATION. In other cases, the rewards concerned the effect that participation would have on students' careers such as CV BUILDING and on LEARNING, which was often linked to the increase of job prospects. We also found that some students consider developing useful PROJECT CODE a reward. Several students were interested in rewards typically linked to traditional OSS developers' motives, such as having a CONTRIBUTING-TO-OSS EXPERIENCE, PEER RECOGNITION, IDEOLOGY ACHIEVEMENT, and developing INTERPERSONAL RELATIONS. Students also participate in GSoC for ACADEMIC ACCOMPLISHMENTS. Furthermore, students indicated different reasons for their interest in the STIPEND, such as paying tuition, living expenses, or simply financial gain [43].

Typically, each student is interested in a different set of rewards. For example, while some students are mostly interested in rewards related to participating in OSS projects, such as acquiring CONTRIBUTING-TO-OSS-EXPERIENCE, others are mostly interested in CAREER PORTFOLIO BUILDING, such as participating in a Google program and contributing to a well-known OSS project. Additionally, while virtually every student considered the practical LEARNING essential for participating in GSoC, few students considered PEER RECOGNITION as critical. This finding suggests that participation rewards influence students' interests to different degrees. We used the generic verb *influence* to indicate how the students' interest and contributions are affected by external factors, because more research is needed to understand the specific type of influence rewards have on the students. Understanding the precise nature of the influence of external factors on students' interests and contributions comprises a gap that future research can explore.

While participation rewards seem to increase students' interest, their level of KNOWLEDGE and SKILLS seem to moderate their interest in contributing to OSS projects, at least in the case of students with more development experience [43]. For example, students with 2-3 years of experience in software development reported being more interested in participating in summer of code programs and becoming frequent contributors than students with ten years or more. Our data also suggests that DEADLINES have a moderation effect, with several students (9) reporting that without them they would have contributed to the projects at a slower pace, explaining that the STIPENDS prompted them to meet agreed timelines. In some cases, family, friends, and acquaintances influenced students' interest to join GSoC.

As aforementioned, we also searched for relevant literature to integrate in our theory. Although understudied, some studies targeted different aspects of *engagement in SoC programs*. Trainer et al. [50] conducted a case study to investigate the *outcomes* of GSoC for one OSS project. Through interviews, the authors identified that some GSoC contributions were merged in the projects' codebases, the students gained NEW SOFTWARE ENGINEERING SKILLS, and the students leveraged their participation for CAREER ADVANCEMENT. The authors also found that mentors faced several challenges, including HELPING A LARGE NUMBER OF APPLICANTS WRITE PROPOSALS during the application process and MAINTAINING AVAILABILITY, since mentors are often volunteers working in their spare time. Trainer et al. [51] also analyzed 22 GSoC projects in the scientific domain to understand GSoC *outcomes* and the *underlying practices that lead to them*. They found that GSoC facilitated the CREATION OF STRONG TIES between mentors and students, reporting that some students became mentors in subsequent years.

Schilling et al. [39] focused on the applicants' fit to the job and with the team. The authors used the concepts of Person-Job (i.e., the congruence between an applicant's desire and job provides) and Person-Team (i.e., the applicant's interpersonal compatibility with the existing team) from the recruitment literature to derive objective measures to predict the retention of 80 former GSoC students in the KDE project. Using a classification schema of prior contributions to this project, they found that intermediate (4-94) and high (>94) numbers of commits were strongly associated with *retention*. Aligned with these results, Silva et. al [42] found that 82% of OSS projects in their sample merged at least one commit from GSoC participants into the codebase. The authors found that the number of commits and code of the students with GSoC experience strongly correlated with how much code they produced and how long they remained.

Silva et al. [43] focused on studying the *students' motivations to enter GSoC*, combining surveys (students and mentors) and interviews (students). They found that, while the STIPENDS are an important motivator, students participate in GSoC for the practical knowledge and the ability to attach the name of organizations (e.g., Google) to their resumés.

Motivation: In relation to works that focus on motivation, self-determination theory [40] is often used to explain the nature of motivation of volunteer contributors (see [55] for a summary). Typically, motivation is organized into intrinsic, extrinsic, and internalized-extrinsic components. Intrinsic motivation refers to performing an activity to satisfy psychological needs for autonomy, competence, and relatedness [40]. Intrinsically motivated behaviors are performed out of interest, requiring no reward other than the enjoyment of performing them [40]. Extrinsically motivated behaviors are instrumental in obtaining external rewards [40]. It is also possible for individuals to internalize extrinsic motivations, which means that although individuals act towards obtaining external rewards (external regulation), their behaviors are driven by internal forces (i.e., self-regulated) [40].

We changed the term *interest*, grounded in the questionnaires, to *motivation*, which is the construct typically used in the literature as the psychological state that antecedes a certain behavior [35, 36, 40]. The literature reports intrinsic motivation for OSS developers to contribute as volunteers to OSS (e.g., [6, 20]), and the students' motivation for entering the program includes enjoyment and fun. Moreover, the literature documents several extrinsically motivated behaviors of OSS developers (e.g., [49]). Most of the rewards in our theory can be considered extrinsically motivated components of participating in the program. Finally, by planning a unique and rich contributing experience, OSS projects strive to convert students into members. We understand this effort as an attempt to make students internalize OSS projects' culture and values. As one project put it: *"The more they [the students] practice, the more it [OSS project's philosophy] becomes part of their philosophy and way of thinking."*

Engagement: Typically, the term *engagement* is not used precisely or consistently, even in the psychology literature [53]. Engagement is a broad construct that researchers study in three domains: cognitive, emotional, and behavioral engagement [26]. In this study, we focus on students' behavioral engagement, which refers to their participation concerning task accomplishments, following norms, and obeying rules [26]. We refer to the behaviors that show the students' positive involvement with tasks as *engaging*.

Outcomes and Stimuli. We used the term *outcomes* in the motivational theory instead of *participation rewards*. While the term *participation rewards* refers to positive outputs students expect to receive, the term *outcomes* allows for positive, neutral, or negative results that may or may not be expected by students. In addition, we split outcomes into *intrinsic outcomes* and *extrinsic outcomes*.

Intrinsic outcomes refer to the outcomes of contribution to OSS projects that become *internal stimuli* to the feelings of autonomy, competence, and relatedness of students' intrinsic motivation [40]. For example, a contribution to OSS projects that does not lower *contribution barriers* [45] may negatively affect students' feelings of autonomy and competence, diminishing their intrinsic motivation. On the other hand, extrinsic outcomes refer to outcomes that can become *external stimuli* to students' extrinsic motivation [40]. For example, we considered the *stipend* an external outcome because it is external to the action of contributing to OSS projects in the context of SoC programs. Students can interpret an outcome in different ways. For example, while some students negatively interpreted the *stipends*, others more constructively framed the reward [43]. We employ the term *functional significance* [9] to refer to the interpretation that students assign to outcomes and external factors.

Knowledge and skills. The literature on contribution to OSS projects considers *knowledge* and *skills* among the main drivers of participation [20]. It is one's set of motivations, combined with

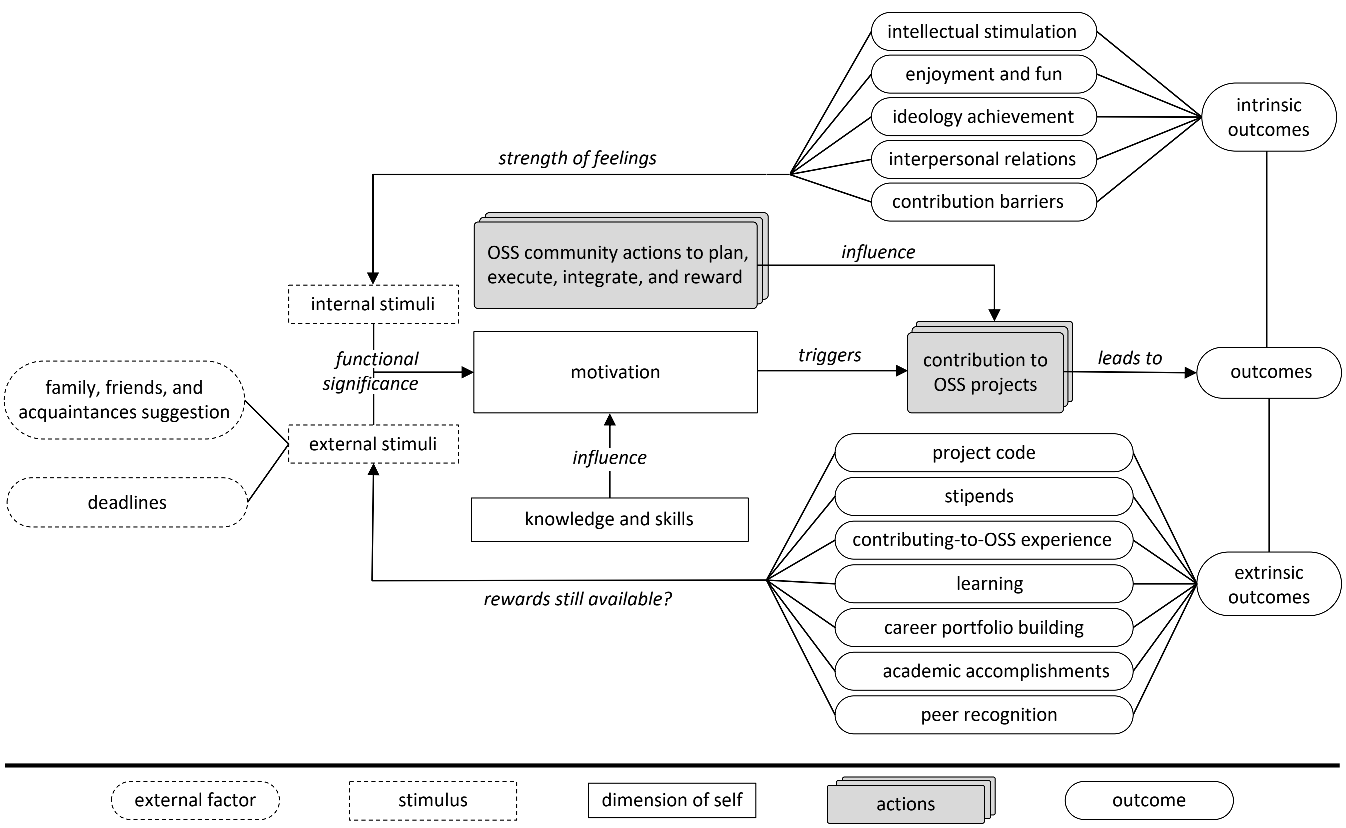


Figure 2: How Summer of Code programs motivate students to contribute to OSS projects (Motivational Theory)

knowledge and skills that trigger one's behavior [24]. In this research, several students and mentors equated participating in GSoC with the pursuit of knowledge and skills.

Answer for RQ2: A summer of code program stimulates students' motivation in three ways. First, it enhances students' sense of competence, autonomy, and relatedness (i.e., intrinsic motivation). Second, it drives engagement, which is instrumental in achieving students' goals. Finally, when students internalize OSS projects' culture and values, they may be more likely to voluntarily contribute after the program.

4.3 The Perception of Potential Participants

We analyzed the motivational theory in light of the perceptions of college students who had never participated in SoC programs. Most students in our sample were between 18-25 years old (Figure 3(a)). While in Brazil, 50% of our respondents declared themselves as females; in China, only 5% of the respondents self-declared as females (Figure 3(b)). No one self-declared as other.

As can be seen in Figure 3(c), most students had not heard of GSoC before participating in the study (Chinese: ≈95%; Brazilian: ≈70%). The only Chinese student that had heard of the program before described it accurately, claiming that his knowledge came from his efforts to join the program. On the other hand, the Brazilian students' descriptions (2) of GSoC were at best simplistic or inaccurate.

We showed the GSoC description to the students and asked them to rank the participation rewards listed in Silva et al. [43] (Figure 4). Most students (20) ranked LEARNING as the most important reward, and several others (8) ranked it second. Similarly, several students (10) ranked CONTRIBUTION TO OSS first, some (5) ranked it second, and a few (2) ranked it third.

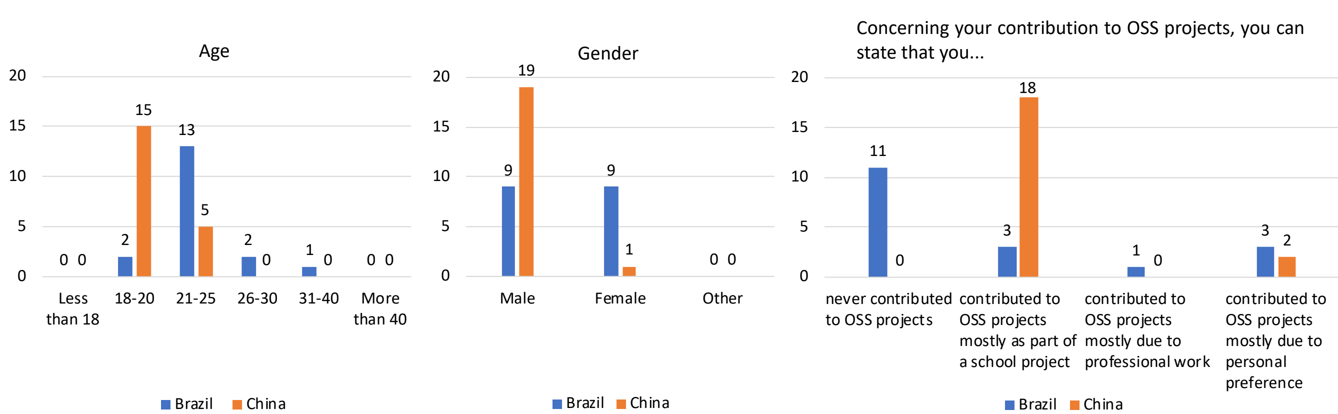
After watching an explanatory video about the theory, most students reported changes in their perception of GSoC. For instance, P10 was surprised by *"being able to develop projects [with] values that I [she] believe[s]."* Encouragingly, several students learned about OSS. As P38 said: *"Participants gain invaluable experience working directly with mentors on OSS projects, and earn a stipend upon successful completion of their project."*

Figure 4 shows the changes in students' perceptions about the importance of the rewards. Except for LEARNING and CONTRIBUTION TO OSS, which remained stable, the other rewards varied greatly. For example, we noticed that the Chinese students did not consider GSoC for CAREER BUILDING (R3), ACADEMIC (R4) concerns, earning STIPENDS (R5), PEER RECOGNITION (R6), or INTELLECTUAL STIMULATION (R7) (Figure 4c). Similarly, Brazilian students did not consider GSoC for some rewards (R6 and R7) (Figure 4b).

Even when students considered the rewards, Figure 4 shows that after being presented with the theory, their perception of the importance of most rewards increased. Figure 4 also shows that the students reprioritized the importance of several rewards. When we observe all participants' rankings, we can see that ACADEMIC concerns ranked last despite their increase in score (Figure 4a). Nevertheless, Figure 4b and Figure 4c suggest that there may be differences among countries that should be further explored. For example, while for Brazilian students, CAREER BUILDING (R3) seems to be more important than STIPENDS (R5), Chinese students seem to think otherwise.

In addition, several students (22) answered that our results influenced their decision to engage in GSoC. However, we noticed that in several cases, students did not feel confident enough in their programming skills to participate in an SoC despite their will to do it. We noticed a pool of potential contributors who need proper encouragement and further guidance to contribute to OSS projects beyond the existing means. Future research could investigate other ways of matching OSS projects with students with low confidence in their programming skills.

Summary of Phase III: By analyzing the students' perceptions, we could observe that the motivational theory broadened their understanding of GSoC and how such a program could assist them in achieving their goals, especially related to their career, inspiring them to engage in such programs in the future.



(a) (b) (c)

Figure 3: Phase III - Student demographics

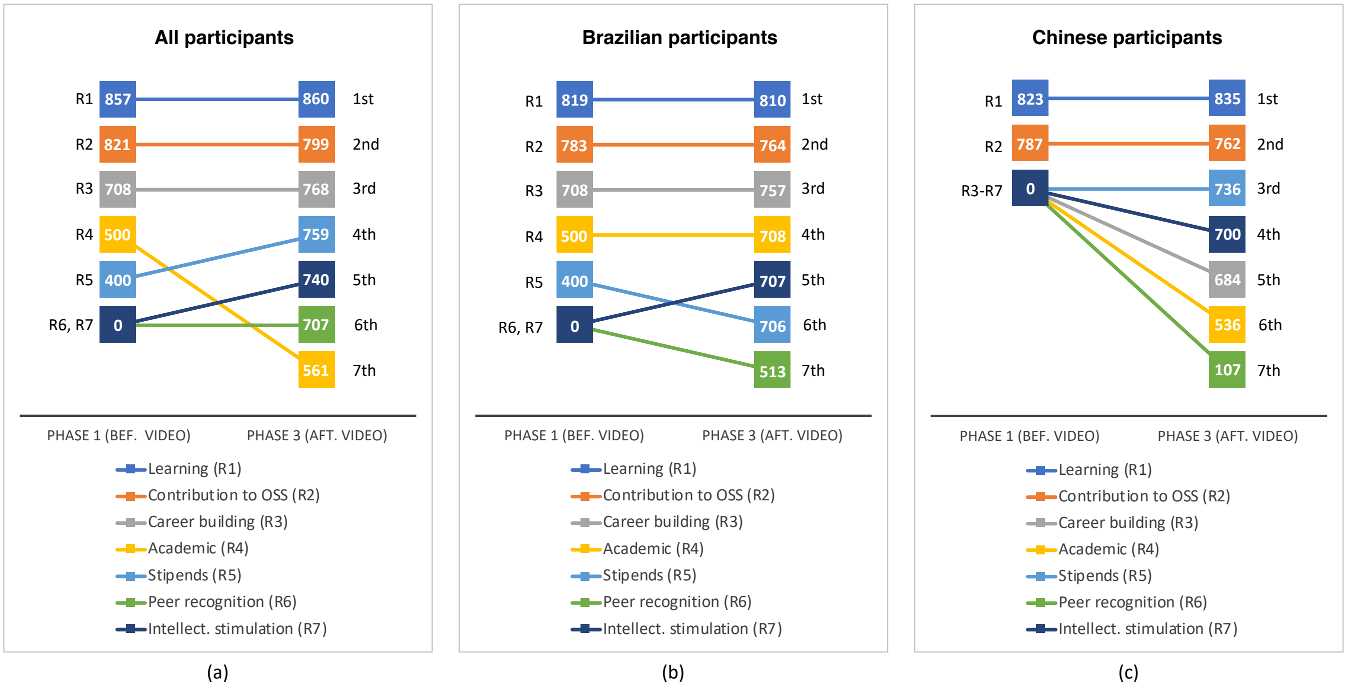


Figure 4: Change in students' perceptions of the importance of rewards

5 DISCUSSION

Our study proposes an integrated theory. The onboarding theory (Figure 1) describes strategies proposed by projects, which converge towards planning and executing, socially integrating students, and rewarding students' contributions and achievements. We noticed that onboarding is labor-intensive and time-consuming and OSS projects should have an adequate structure to provide support for onboarding students. This may be particularly problematic for small communities. Future research can develop onboarding tools specific for students, which could be deployed as software bots [58], for example. Future research can also investigate the effectiveness of each strategy and the context in which they should be employed.

We also noticed the absence of strategies focused specifically on promoting diversity and inclusion of underrepresented populations, such as women. This is an important point to revisit given that previous research has shown that current tools and platforms in OSS are gender-biased [27].

In the motivational theory, we show that students' motivation is influenced by the *outcomes* of engaging in SoCs—which include participation rewards. Outcomes can influence students' motivation differently. Even if students enter an SoC interested in the same rewards, the outcomes will undoubtedly differ because each experience is unique due to individual differences. For example, if two students entered GSoC equally interested in the *stipends*, their response to *having to meet deadlines* may differ. While the deadlines negatively influenced some students [43], others framed it more constructively [25].

Following Ralph's advice [33], we also compare the explanatory power of our theory to others. In general, the Legitimate Peripheral Participation (LPP) theory is used to explain how newcomers engage in OSS projects [22] and become contributors [41, 46]. Newcomers begin their involvement by observing experienced project members and, after a while, they become in charge of straightforward but valuable tasks. In time, newcomers become familiar with contribution norms and take on more important tasks. This process culminates in the emergence of frequent contributors [22]. However, LPP does not describe precisely the engagement that occurs through SoC programs. Students usually do not start at the margin by observing experienced members. Instead, they are individually guided—and sponsored—to become contributors. The student-project relationship in an SoC context is mediated by a contract that binds students and mentors for three months. Therefore, our findings indicate that more research is necessary to understand how students can be legitimized as full project members in an SoC context.

In general, the results presented here help explain why new contributors participate in OSS communities, considering the comprehensive set of rewards offered by these programs. In Phase III, we observed that the motivational theory helped students change their perception about the rewards from joining SoC programs and that they became inspired to engage in such programs and OSS. Therefore, this may also help OSS projects to devise strategies to attract and retain students. Those involved in running these communities can increase their ability to attract developers, mentors, and ultimately retain participants (in some cases being hired to do the work they started as volunteers).

6 IMPLICATIONS

Research: Our theory provides an understanding of newcomer engagement in OSS projects through a corporate sponsored SoC program. As Ralph explained [33], process theories offer a foundation for the development of engagement methods. While process theories are concerned with how entities (i.e., motivation) change, methods *"prescribe practices, techniques, tools, or sequences that are ostensibly better than their alternatives"* [33, p. 20]. Researchers could extend our results by studying methods and models, taking into account OSS projects' peculiarities. Moreover, while this research is specific to open source communities, we believe that the results go beyond them, such as the motivations for students to build their résumés and to seek out challenging work as part of their early development. We also acknowledge the body of knowledge about onboarding online communities in general [18]. A future work would involve comparing and extending our theory based on this literature.

OSS projects: Understanding how to onboard students in OSS and how the variety of rewards influence students' motivation can help OSS projects to devise strategies to attract and retain students. Moreover, OSS projects can use our results to make a well-informed decision about their participation in SoC programs. While OSS projects that already participate in such programs can revise their action plans in light of our results, projects that have never applied can use them as a guide.

Students: Our theory can transfer the experiences of former participants to students who have not yet participated in an SoC program. In this way, our theory can broaden new students' perspectives, not only giving them a better understanding of SoC programs in general, but also communicating participation rewards that motivated former students. Although students generally see the benefits of getting involved in OSS projects [30], the theory may help to show the advantages of SoC programs.

Program organizers: Those running SoC programs (including many large software organizations) can leverage our results to devise guidelines for the participating projects, including how to select proposals and engage students.

7 LIMITATIONS AND THREATS TO VALIDITY

As any empirical research, our study has some limitations and potential threats to validity. We discuss them in this section.

Transferability of the results: Our results are grounded in data from GSoC. Hence, our theory may not necessarily transfer to other SoC programs. Nevertheless, we believe that GSoC motivators and the onboarding strategies can be replicated in other contexts in which engaging students is important.

Data Representativeness: Although we collected data from multiple sources for a variety of OSS projects, we likely did not find all onboarding strategies or consider all factors that motivate students to contribute. Each OSS project has its singularities, and the actions for engaging in SoC programs can differ. With more data, perhaps we could find different ways of categorizing concepts, which could increase explanatory power. Further studies are necessary to broaden the scope of our analysis.

Subjectivity of the data analysis: Another threat to the validity of our results is the data classification's subjectivity. To alleviate this threat, we employed grounded theory procedures [8], which require the complete analysis to be grounded in collected data. However, when applying grounded theory, there is always an "*uncodifiable step*," which relies on researchers' interpretations [3, 21].

Limitations from using project applications as a data source (RQ1): We are aware that OSS projects have limited space for revealing their action plans when they apply for GSoC, which potentially makes them report the actions that increase their odds of acceptance in the program. Moreover, the applications describe the plan that the projects have and not their actual onboarding process. In this way, underreporting might occlude actions that are relevant for the OSS projects' decision process of engaging in SoC programs. On the other hand, projects submit, and probably refine, these applications yearly, increasing their accuracy and completeness. Future work can gain understanding of the effectiveness of the actions by interviewing or surveying project members, mentors, and students or conducting ethnographic studies in the actual projects.

Survivability bias (RQ2): Since we could not contact applicants who were rejected from the program, the primary data that we used to build our motivation theory is from students that were accepted, and thus our theory is biased towards those accepted applicants. Future research can devise methods to reach rejected applicants. Having both types of applicants would help the theory to explain successful and unsuccessful engagement cases, thereby increasing its explanatory power.

Evaluation with students (Phase III): The students we surveyed are not necessarily representative of the intended target of the theory and do not necessarily match the actual participants of OSS or GSoC. Moreover, the sample size is small and was collected from only two countries, leveraging the authors' personal networks. The results are promising, and a large-scale study is deemed necessary. Differences among countries and other personal characteristics could also be explored in such a large-scale study. Another threat related to the evaluation is the confirmation bias. However, the changes in the ranking are less susceptible to this kind of bias and may reveal motivators of which the students are not normally aware.

8 RELATED WORK

We have already discussed more specific related work in Section 4.2. In the following, we summarize broader literature on onboarding and motivation to contribute to OSS.

8.1 Onboarding in OSS

Many studies focus on newcomers onboarding to OSS projects [29, 56, 57]. Mentoring was also explored as a way to support newcomers, and it is particularly relevant to SoC programs. In fact, the importance of mentorship as part of the knowledge acquisition process for novices is evidenced in the theory of software development expertise developed by Baltes and Diehl [3]. In closed source settings, it is common practice to offer formal mentorship to newcomers to support their first steps [5]. In the OSS domain, researchers proposed approaches to recommend mentors to newcomers [7, 23, 28, 47]. Fagerholm et al. [10] conducted a case study to assess the impact of mentoring support on developers and found that it significantly improves newcomer onboarding. Schilling et al.

[38] studied the impact of mentoring on developers' training and retention in OSS projects. In contrast, Labuschagne and Holmes [19], who studied Mozilla, evidenced that onboarding programs may not result in long-term contributors, even though mentored newcomers considered the mentorship program valuable. Balali et al. [2] analyzed challenges mentors face when onboarding newcomers as well as how they recommend tasks to newcomers [1]. To the best of our knowledge, no previous work has focused on mentoring strategies in the context of SoC programs.

8.2 Motivation to contribute to OSS

Motivation can be defined as the conscientious governance process for decisions considering the existing possible forms of volunteer action [12]. Motivation is internal to the individual, can vary according to goals, has fervor and duration, and regulates human behavior [13]. Motivation is studied in a variety of areas, including software engineering. In 2006, Beecham et al. [4] conducted a systematic literature review and found 92 relevant papers about motivation in this area. This systematic literature review was later updated by França et al. [14], who analyzed the period from 2006 to 2010 and found 54 relevant studies. Specifically to OSS, Von Krogh et al. [55] reviewed the literature in 2009 and found 40 relevant studies. As aforementioned, none of these studies focus on students, which are the focus of this work. We also consider as related work studies that investigate motivation to participate in short term coding activities, such as Hackathons (e.g., [31, 44]).

9 CONCLUSION

Attracting and retaining new contributors are vital to the sustainability of OSS projects that depend on a volunteer workforce. Some OSS projects participate in summer of code programs expecting to onboard new contributors. In this study, we developed an engagement theory that explains how to onboard students and how students become motivated to participate. Our theory is grounded in multiple data sources, such as the guides provided by program administrators [48], OSS projects' applications for GSoC, surveys involving students and mentors [43], interviews with students [43], quantitative studies [42], and the literature (e.g., [38, 50–52]). We employed grounded theory procedures to merge previous research findings with our results and to explain contributions to OSS projects through SoC programs. We claim that the development of our engagement theory is a first step towards building a *variance theory* that can explain in greater detail why and when students meaningfully contribute to OSS projects. A variance theory could ultimately predict the students who are more likely to continue contributing to OSS projects after participation.

ACKNOWLEDGEMENTS

We thank all the participants of this study who volunteered to support our research. This work was partially supported by the CNPq (430642/2016-4), FAPESP (2015/24527-3), and the National Science Foundation (grants 1815503 and 1900903). This work was conducted as part of a Ph.D. dissertation in the Computer Science Department at the University of São Paulo (USP).

REFERENCES

[1] Sogol Balali, Umayal Annamalai, Susmita Padala, Bianca Trinkenreich, Marco Gerosa, Igor Steinmacher, and Anita Sarma. 2020. Recommending Tasks to Newcomers in OSS Projects: How Do Mentors Handle It?. In *16th International Symposium on Open Collaboration (OpenSym 2020)*. ACM, 14.

[2] Sogol Balali, Igor Steinmacher, Umayal Annamalai, Anita Sarma, and Marco Aurelio Gerosa. 2018. Newcomers' Barriers. . . Is That All? An Analysis of Mentors' and Newcomers' Barriers in OSS Projects. *Computer Supported Cooperative Work (CSCW)* 27, 3 (01 Dec 2018), 679–714. https://doi.org/10.1007/s10606-018-9310-8

[3] Sebastian Baltes and Stephan Diehl. 2018. Towards a Theory of Software Development Expertise. In *Proceedings of the 26th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE '18)*. ACM, Lake Buena Vista, FL, 14.

[4] Sarah Beecham, Nathan Baddoo, Tracy Hall, Hugh Robinson, and Helen Sharp. 2008. Motivation in Software Engineering: A systematic literature review. *Information and software technology* 50, 9-10 (2008), 860–878.

[5] Andrew Begel and Beth Simon. 2008. Novice Software Developers, All over Again. In *Proceedings of the Fourth International Workshop on Computing Education Research (ICER '08)*. ACM, New York, NY, USA, 3–14. https://doi.org/10.1145/1404520.1404522

[6] Jürgen Bitzer, Wolfram Schrettl, and Philipp JH Schröder. 2007. Intrinsic motivation in open source software development. *Journal of Comparative Economics* 35, 1 (2007), 160–169.

[7] Gerardo Canfora, Massimiliano Di Penta, Rocco Oliveto, and Sebastiano Panichella. 2012. 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 (FSE '12)*. ACM, New York, NY, USA, Article 44, 11 pages. https://doi.org/10.1145/2393596.2393647

[8] Kathy Charmaz. 2006. *Constructing Grounded Theory*. SAGE Publications, London, UK. 224 pages.

[9] Edward L Deci, Richard Koestner, and Richard M Ryan. 1999. A meta-analytic review of experiments examining the effects of extrinsic rewards on intrinsic motivation. *Psychological Bulletin* 125, 6 (1999), 627–668.

[10] Fabian Fagerholm, Alejandro S. Guinea, Jürgen Münch, and Jay Borenstein. 2014. The Role of Mentoring and Project Characteristics for Onboarding in Open Source Software Projects. In *ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM '14)*. ACM, 55:1–55:10. https://doi.org/10.1145/2652524.2652540

[11] Fabian Fagerholm, Alejandro Sanchez Guinea, Jay Borenstein, and Jurgen Munch. 2014. Onboarding in Open Source Projects. *IEEE Software* 31, 6 (2014), 54–61.

[12] César França and Fabio QB da Silva. 2010. Designing motivation strategies for software engineering teams: an empirical study. In *Proceedings of the 2010 ICSE Workshop on Cooperative and Human Aspects of Software Engineering*. 84–91.

[13] César França, Fabio QB Da Silva, and Helen Sharp. 2018. Motivation and satisfaction of software engineers. *IEEE Transactions on Software Engineering* (2018).

[14] César França, Tatiana Gouveia, Pedro Santos, Celio Santana, and Fabio QB da Silva. 2011. Motivation in software engineering: A systematic review update. In *15th Annual Conference on Evaluation & Assessment in Software Engineering (EASE 2011)*. IET, 154–163.

[15] Gillian J Greene and Bernd Fischer. 2016. Cvexplorer: Identifying candidate developers by mining and exploring their open source contributions. In *Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering*. 804–809.

[16] Jo E. Hannay, Dag I. K. Sjoberg, and Tore Dyba. 2007. A systematic review of theory use in software engineering experiments. *IEEE Transactions on Software Engineering* 33, 2 (2007), 87–107.

[17] Guido Hertel, Sven Niedner, and Stefanie Herrmann. 2013. Motivation of Software Developers in Open Source Projects : An Internet-Based Survey of Contributors to the Linux Kernel. *Research Policy* 32, 7 (2013), 1159–1177.

[18] Robert E Kraut and Paul Resnick. 2012. *Building successful online communities: Evidence-based social design*. Mit Press.

[19] Adriaan Labuschagne and Reid Holmes. 2015. Do Onboarding Programs Work?. In *Proceedings of the 12th Working Conference on Mining Software Repositories (MSR '15)*. IEEE Press, Piscataway, NJ, USA, 381–385. https://doi.org/10.1109/MSR.2015.45

[20] Karim R. Lakhani and Robert G. Wolf. 2005. Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects. In *Perspectives on Free and Open Source Software*. The MIT Press, Cambridge, MA, Chapter Chapter 1, 570.

[21] Ann Langley. 1999. Strategies for theorizing from process data. *Academy of management review* 24, 4 (1999), 691–710.

[22] Jean Lave and Etienne Wenger. 1991. *Situated learning: Legitimate Peripheral Participation*. Cambridge University Press, UK. 138 pages.

[23] Yuri Malheiros, Alan Moraes, Cleyton Trindade, and Silvio Meira. 2012. A Source Code Recommender System to Support Newcomers. In *Proceedings of the IEEE 36th Annual Computer Software and Applications Conference (COMPSAC '12)*. IEEE, Los Alamitos, California, USA, 19–24. https://doi.org/10.1109/COMPSAC.2012.11

[24] Terence R. Mitchell and Denise Daniels. 2003. Motivation. In *Handbook of Psychology*. John Wiley & Sons, Inc., Hoboken, NJ, USA. https://doi.org/10.1002/0471264385.wei1210

[25] Ivange Larry Ndumbe. 2016. GSoC 2016 Final Evaluation. https://ivange94blog.wordpress.com/2016/08/20/gsoc-2016-final-evaluation/

[26] Tuan Dinh Nguyen, Marisa Cannata, and Jason Miller. 2018. Understanding student behavioral engagement : Importance of student interaction with peers and teachers. *The Journal of Educational Research* 111, 2 (2018), 163–174.

[27] Susmita Hema Padala, Christopher John Mendez, Luiz Felipe Dias, Igor Steinmacher, Zoe Steine Hanson, Claudia Hilderbrand, Amber Horvath, Charles Hill, Logan Dale Simpson, Margaret Burnett, et al. 2020. How Gender-biased Tools Shape Newcomer Experiences in OSS Projects. *IEEE Transactions on Software Engineering* (2020).

[28] Sebastiano Panichella. 2015. Supporting newcomers in software development projects. In *IEEE International Conference on Software Maintenance and Evolution (ICSME 2015)*. IEEE, 586–589. https://doi.org/10.1109/ICSM.2015.7332519

[29] Yunrim Park and Carlos Jensen. 2009. Beyond pretty pictures: Examining the benefits of code visualization for open source newcomers. In *2009 5th IEEE International Workshop on Visualizing Software for Understanding and Analysis*. IEEE, 3–10.

[30] Gustavo Pinto, Clarice Ferreira, Cleice Souza, Igor Steinmacher, and Paulo Meirelles. 2019. Training Software Engineers Using Open-Source Software: The Students' Perspective. In *Proceedings of the 41st International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET '19)*. IEEE Press, 147–157. https://doi.org/10.1109/ICSE-SEET.2019.00024

[31] Arie Purwanto, Anneke Zuiderwijk, and Marijn Janssen. 2019. Citizens' motivations for engaging in open data hackathons. In *International Conference on Electronic Participation*. Springer, 130–141.

[32] Paul Ralph. 2015. Developing and Evaluating Software Engineering Process Theories. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*. IEEE, Florence, Italy, 20–31. https://doi.org/10.1109/ICSE.2015.25

[33] Paul Ralph. 2019. Toward methodological guidelines for process theories and taxonomies in software engineering. *IEEE Transactions on Software Engineering* 45, 7 (2019), 712–735.

[34] Dirk Riehle. 2015. How Open Source Is Changing the Software Developer's Career. *IEEE Computer* 48, 5 (2015), 51–57.

[35] Jeffrey Roberts, Il-Horn Hann, and Sandra Slaughter. 2006. Understanding the Motivations, Participation, and Performance of Open Source Software Developers: A Longitudinal Study of the Apache Projects. *Management Science* 52, 7 (2006), 984–999.

[36] Richard M Ryan and Edward L Deci. 2000. Intrinsic and Extrinsic Motivations: Classic Definitions and New Directions. *Contemporary Educational Psychology* 25, 1 (2000), 54–67.

[37] Johnny Saldana. 2015. *The coding manual for qualitative researchers*. SAGE Publications, Inc, London, UK.

[38] Andreas Schilling, 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 *Proceedings of the Annual Hawaii International Conference on System Sciences (HICSS)*. IEEE, Maui, HI, 3446–3455.

[39] Andreas Schilling, Sven Laumer, and Tim Weitzel. 2014. Stars matter: how FLOSS developers' reputation affects the attraction of new developers. In *Proceedings of the 52nd ACM conference on Computers and people research - SIGSIM-CPR '14*. ACM Press, New York, New York, USA, 5–10. https://doi.org/10.1145/2599990.2599991

[40] C. Scott Rigby, Edward L. Deci, Brian C. Patrick, and Richard M. Ryan. 1992. Beyond the intrinsic-extrinsic dichotomy: Self-determination in motivation and learning. *Motivation and Emotion* 16, 3 (1992), 165–185. https://doi.org/10.1007/BF00991650

[41] Dan Sholler, Igor Steinmacher, Denae Ford, Mara Averick, Mike Hoye, and Greg Wilson. 2019. Ten simple rules for helping newcomers become contributors to open projects. *PLOS Computational Biology* 15, 9 (09 2019), 1–10. https://doi.org/10.1371/journal.pcbi.1007296

[42] Jefferson O. Silva, Igor Wiese, Daniel German, Igor Steinmacher, and Marco A Gerosa. 2017. How Long and How Much : What to Expect from Summer of Code Participants?. In *Proceedings of 33rd International Conference on Software Maintenance and Evolution (ICSME)*. IEEE, Shanghai, China, 10.

[43] Jefferson O Silva, Igor Wiese, Daniel M German, Christoph Treude, Marco A Gerosa, and Igor Steinmacher. 2020. Google summer of code: Student motivations and contributions. *Journal of Systems and Software* 162 (2020), 110487.

[44] Anthony Simonofski, Victor Amaral de Sousa, Antoine Clarinval, and Benoıt Vanderose. 2020. Participation in Hackathons: A Multi-Methods View on Motivators, Demotivators and Citizen Participation. In *International Conference on Research Challenges in Information Science*.

[45] Igor Steinmacher, Tayana Conte, Marco Aurélio Gerosa, and David F. Redmiles. 2015. 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 (CSCW)*. ACM, Vancouver, BC, 1379–1392.

[46] Igor Steinmacher, Marco Gerosa, Tayana U. Conte, and David F. Redmiles. 2019. Overcoming Social Barriers When Contributing to Open Source Software Projects. *Comput. Supported Coop. Work* 28, 1–2 (April 2019), 247–290. https://doi.org/10.1007/s10606-018-9335-z

[47] Igor Steinmacher, Igor Scaliante Wiese, and Marco Aurélio Gerosa. 2012. Recommending mentors to software project newcomers. In *2012 Third International Workshop on Recommendation Systems for Software Engineering (RSSE)*. IEEE, 63–67.

[48] The Google Summer of Code Guides Authors. 2018. Google Summer of Code Mentor Guide. https://google.github.io/gsocguides/mentor/index

[49] Jean Tirole and Josh Lerner. 2002. Some Simple Economics of Open Source. *The Journal of Industrial Economics* 50, 2 (2002), 197–234.

[50] Erik H Trainer, Chalalai Chaihirunkarn, and James D Herbsleb. 2014. The Big Effects of Short-term Efforts: Mentorship and Code Integration in Open Source Scientific Software. *Journal of Open Research Software* 2, 1 (jul 2014), e18.

[51] Erik H. Trainer, Chalalai Chaihirunkarn, Arun Kalyanasundaram, and James D. Herbsleb. 2014. Community Code Engagements: Summer of Code & Hackathons for Community Building in Scientific Software. In *Proceedings of the 18th International Conference on Supporting Group Work (GROUP)*. ACM, Sanibel Island, Florida, 111–121.

[52] Erik H Trainer, Arun Kalyanasundaram, Chalalai Chaihirunkarn, and James D Herbsleb. 2016. How to Hackathon: Socio-technical Tradeoffs in Brief, Intensive Collocation. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing - CSCW '16*. ACM, New York, NY, USA, 1116–1128.

[53] Catherine Truss, Kerstin Alfes, Rick Delbridge, Amanda Shantz, and Emma Soane. 2013. *Employee engagement in theory and practice*. Routledge.

[54] Leona E. Tyler. 1947. *The psychology of human differences*. D Appleton-Century Company, New York, NY.

[55] Georg Von Krogh, Stefan Haefliger, Sebastian Spaeth, and Martin W Wallin. 2012. Carrots and rainbows: Motivation and social practice in open source software development. *MIS Quarterly* 36, 2 (2012), 649–676.

[56] Georg Von Krogh, Sebastian Spaeth, and Karim R. Lakhani. 2003. Community, joining, and specialization in open source software innovation: A case study. *Research Policy* 32, 7 (2003), 1217–1241.

[57] Jianguo Wang and Anita Sarma. 2011. 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*. 76–79.

[58] Mairieli Wessel, Bruno Mendes De Souza, Igor Steinmacher, Igor S Wiese, Ivanilton Polato, Ana Paula Chaves, and Marco A Gerosa. 2018. The power of bots: Characterizing and understanding bots in oss projects. *Proceedings of the ACM on Human-Computer Interaction* 2, CSCW (2018), 1–19.

[59] Frances Zlotnick. 2017. GitHub Open Source Survey 2017. https://doi.org/10.17605/OSF.IO/ENRQ5