Engagement in Open Source Software Projects via Summer of Code Programs

Jefferson de Oliveira Silva

Text submitted
to the
Institute of Mathematics and Statistics
of the
University of São Paulo
for the
Doctoral Degree in Science

Program: Computer Science

Advisor: Prof. Dr. Marco Aurélio Gerosa

Co-Advisor: Prof. Dr. Igor Fábio Steinmacher

São Paulo, October, 2019

Engagement in Open Source Software Projects via Summer of Code Programs

This version of the thesis contains the changes suggested by the Committee Members during the public defense of the original version of this work, which occurred in Dec 3rd, 2019. A copy of the original version is available at the Institute of Mathematics and Statistics (IME)–University of São Paulo (USP).

Committee Members:

- Prof. Dr. Marco Aurélio Gerosa (advisor) - NAU
- Prof. Dr. Igor Fábio Steinmacher (co-advisor) - UTFPR
- Prof. Dr. Fabio Kon - IME-USP
- Prof. Dr. Cleidson de Souza - UFPA
- Prof. Dr. Gregorio Robles - URJC
- Prof. Dr. Paulo Meirelles - UNIFESP
- Prof. Dr. Cesar França - UFRPE

Acknowledgements

Undertaking this Ph.D. was a life-changing experience for me. To become a researcher, made me feel as if I had developed a new pair of eyes to scan the world. It would certainly not be possible to conclude the program without the support and guidance of many people.

I would like to start by thanking Marco Gerosa, my advisor, for his vision, knowledge, and guidance. Marco has been a real example of an outstanding researcher. He raised the bar on my previous academic standards, relentlessly encouraging me to improve my work. I am also very thankful to Igor Steinmacher, my co-advisor, for kindly tutoring me, patiently revising my work, and always being available for clearing doubts. Igor is quite easygoing, and it was a pleasure to work with him.

I am profoundly grateful to Christoph Treude for his reviews, who always showed me how to transform a mass of cluttered ideas into elegant, simple reasoning. Without his touch, I would likely not have published several results timely. I would like to express my sincere gratitude to Daniel German for our enlightening conversations, which always guided me into being more objective. So many valuable insights he gave me.

Many thanks to Yorah Bosse, Igor Wiese, and Italo Vega for all their support and encouragement when things got tough. Their valuable assistance inspired me to overcome several hurdles. I like to think of them as dear friends. I am grateful to Daniel Gatti, Eduardo Savino, Talitha Nicoletti, and Vagner Diniz for giving me the time to research during working hours.

I am very grateful to the members of the qualifying committee and the final examining committee. It was a pleasure to receive feedback from such distinguished researchers: Fabio Kon, Paulo Meirelles, Cleidson de Souza, Cesar França, and Gregorio Robles. I extend this gratitude to all the anonymous reviewers of the scientific community that gave valuable feedback to preliminary versions of this work reported in scientific papers. I also gratefully acknowledge the funding received from CNPq (430642/2016-4) and FAPESP (Grant 2015/24527-3).

I am eternally grateful to all my family. It would be unthinkable to have undergone the program without the assistance of my parents, Gilson Silva and Gicelia Silva, who sustained me in every possible way. Also, I cannot imagine how I could have passed through this Ph.D. without the caring and love of my wife, Francine Nagy, who embraced my dream and moved the world to make it real.

Finally, I dedicate this work to my beloved children, Raphael and Felipe, the two little superheroes with the superpower of making me never give up, even without being aware of it. Everything is for you two.

Abstract

SILVA, J. O. **Engagement in Open Source Software Projects via Summer of Code Programs**. 2019. 122f. Tese (Doutorado) - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2019.

Several Open Source Software (OSS) projects need a constant influx of newcomers, as member disengagement poses sustainability challenges. A significant hurdle for OSS projects is to discover ways to attract and retain newcomers. Some OSS projects expect to onboard newcomers and receive contributions by participating in Summer of Code programs. Such programs aim at fostering software development by typically matching OSS projects with students. Nevertheless, despite their growing popularity, there is little information about what makes students engage in OSS by participating in such programs and how it can benefit OSS projects. Summer of Code programs are typically laborious and time-consuming, and several OSS projects do not have enough empirical evidence to make a well-informed decision about the actual costs of participating in such programs, which could hinder the projects' capacity to address their needs timely. In this thesis, our goal is to reveal the students' engagement in OSS projects via Summer of Code programs. We focused on Google Summer of Code as our case study, as it is best-known, with several developers from across the globe, and in operation for more than a decade. We employed multiple empirical methods such as interviews, surveys, and quantitative analysis of data mined from source code repositories. We claim two main novel contributions: the empirical identification of the tasks that OSS projects accomplish to motivate students in such programs, and; the empirical identification of factors that influence students to engage in OSS via Summer of Code programs. With our results, OSS projects have a concise and structured description of what makes students engage in OSS projects when they participate in such programs; new students can benefit from former students' experiences, and; researchers have a theoretical foundation to devise methods and models. In this way, we believe that OSS projects can ultimately gain more contributions from students interested in short and long-term collaboration.

Keywords: Summer of Code · Open Source Software · Process Theory · Newcomer · Retention · Sustainability · Joining Process.

Contents

List of Abbreviations

CSCW	Computer-Supported Cooperative Work
GSoC	Google Summer of Code
GT	Grounded Theory
LPP	Legitimate Peripheral Participation
ICOSS	International Conference on Open Source Systems
ICSME	International Conference on Software Maintenance and Evolution
JSS	Journal of Systems and Software
OSS	Open Source Software
PT	Process Theory
SBES	*Simpósio Brasileiro de Engenharia de Software*
SBSC	*Simpósio Brasileiro de Sistemas Colaborativos*
SCM	Source Code Management System
SDT	Self-Determination Theory

List of Figures

List of Tables

Chapter 1

Introduction

Open source software (OSS) projects that mostly depend on a volunteer workforce face the constant challenge of engaging enough newcomers to replace the productive members who eventually quit [TCKH14]. Engaging newcomers is challenging for several OSS projects, mainly because newcomers may not be strongly committed, which makes them susceptible to quit [FN09].

Newcomers also quit when they face difficulties in placing their first contributions to OSS projects [Ste15]. By expecting newcomers to learn to contribute by themselves [Sca02], OSS projects may be creating a barrier that is too high for newcomers to cross. Assisting newcomers to pass through such barrier is essential to their motivation, engagement, and retention in order to promote a sustainable number of developers in an OSS project [QF10].

Several OSS projects join Summer of Code programs expecting to onboard newcomers [SWG$^+$17, TCKH14]. Summers of Code are programs that foster software development by matching students (typically) with OSS projects. Some 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]

Engaging in OSS via Summer of Code programs may be more attractive to some students than engaging in OSS projects directly. Often, such programs are usually sponsored by well-known organizations, such as Facebook, Debian, and Google [TCKH14, TCH14]. For example, students that joined a Summer of Code that offered stipends would add two entries to their CV (the names of the OSS project and the sponsoring organization), besides earning the payment. We believe that students that joined Summer of Code programs are likely to have personal goals beyond becoming active OSS project contributors.

On the other hand, mentoring is a labor-intensive and time-consuming endeavor. Providing students with high-quality mentoring could be considered a good investment of efforts provided that they become long-term contributors or develop meaningful contributions. Otherwise, mentoring could be not only a waste of energy but also hinder mentors from addressing urgent project needs [SWG$^+$17, TCKH14]. Moreover, high-profile software companies attract a high number of

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

applicants, which creates several tasks that OSS projects need to tackle even before the program's kickoff [TCKH14].

Despite Summers of Code apparent growing popularity, there is little information for the OSS projects that desire to make a well-informed decision about participating in such programs. The OSS projects—especially first time participants—that hope to engage newcomers through such programs can acquire some idea about the engagement process by reading guides.[5] Although guides contain valuable tips, they do not detail specific tasks OSS projects need to accomplish in such programs.

Moreover, the literature offers scant evidence to the ones interested in understanding how engagement via such programs benefits OSS projects. For example, in two different studies, Trainer and colleagues [TCKH14, TCH14] identified a few outcomes of GSoC for scientific projects only. Schilling et al. [SLW12] identified objective measures to predict the engagement and retention of former GSoC students, but their scope is limited to the investigation of the KDE project. These studies offer a valuable contribution to the literature, but they do not explain what the engagement via such programs entails for OSS projects. Moreover, the literature lists a few factors that seem to be related to engagement and retention in OSS projects but offers little to nothing concerning how these factors motivate students to place contributions. As a consequence, some OSS projects may face a dilemma about whether engaging in OSS via Summer of Code programs is indeed beneficial.

Thus, in this thesis, we focus on the engagement in OSS via Summer of Code programs for OSS projects and students. With a more in-depth understanding of the outcomes of this type of engagement, researchers and OSS projects can invest their efforts in devising methods, ultimately gaining more meaningful contributions and loyal contributors.

1.1 Research Goals

The goals of this research are twofold. First, we target at identifying the tasks students accomplish in Summer of Code programs and understanding how accomplishing such tasks can benefit OSS projects. Second, we aim to understand what motivates students to engage in OSS projects when they participate in such programs.

"How does participation in Summer of Code programs benefit OSS projects?"

To guide our answer, we have defined specific research questions (RQ):

RQ1. How do OSS projects mentor students in Summer of Code programs?

RQ2. How do Summer of Code programs motivate students to engage in OSS projects?

[5] https://developers.google.com/open-source/gsoc/resources/guide

1.2 Research Design

This thesis uses multiple empirical methods to address the research problem and achieve its goals. More specifically, this research is a result of the analysis of surveys, interviews, and students' contributions to software repositories. The research design comprises three phases, as presented in Figure 1.1, and described in the following.

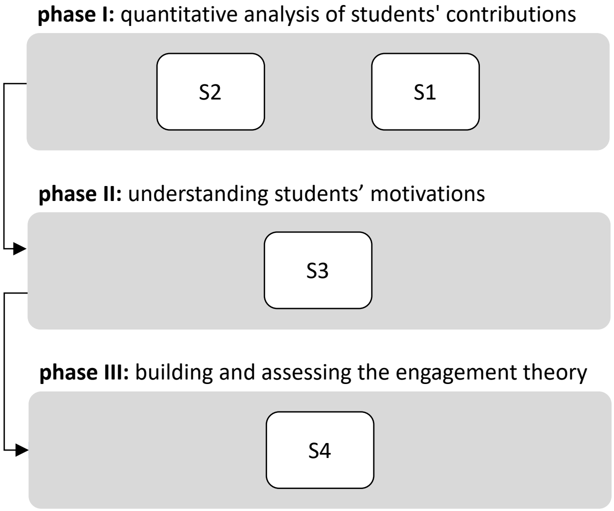

phase I: quantitative analysis of students' contributions
S2
S1
phase II: understanding students' motivations
S3
phase III: building and assessing the engagement theory
S4

Figure 1.1: *Research design*

1.2.1 Phase I: Quantitative analysis of students' contributions

We identified several information sources relevant for answering our RQs: source code management systems (SCM), issue trackers, mailing lists, Summer of Code students and mentors, OSS projects' applications for Summer of Code programs, and students who never engaged in OSS via such programs. Therefore, we designed studies to explore each data source in varying degrees. In this phase, we started by conducting S1, which consists of a preliminary exploration of SCMs, issue trackers, and mailing lists. Next, we conducted S2 for obtaining objective measures of students' contributions to OSS projects in a broader scope.

Study 1 (S1): We investigated contributions (i.e., commits) and retention of 33 students who engaged in OSS via Google Summer of Code (GSoC) 2014 to work in Apache Software Foundation projects. In S1, we analyzed students' contributions in three data sources: an SCM (GitHub), issue trackers, and mailing lists. We chose to investigate GSoC since it is a well-established program, running for more than a decade, with several developers from all over the world. Moreover, other Summer of Code programs use the same GSoC eligibility criteria to select their students, such as Julia Summer of Code, Season of KDE, and Tor Summer of Privacy (check Appendix A for more information on Summer of Code programs). Similarly, we chose to study the Apache Software Foundation for its projects' diversity and because it has been frequently targeted in empirical works (e.g., [GPVF15, SWCG12]). Additionally, Apache projects adhere to process standards, which increases reliability in data collection and analysis. We found that the students of our sample did not keep contributing to their assigned projects after GSoC, except two students, who were already project contributors before GSoC kickoff [SWSG15]. Furthermore, S1 contributed to establishing the research method employed in S2.

Study 2 (S2): To objectively understand how long and how much Summer of Code students contribute to OSS projects, we quantitatively analyzed participation (in terms of the number of commits, code churn, and commit date intervals) of 367 GSoC students by mining software repositories. We limited the analysis to the GSoC 2013-2015 editions. We split students into two groups: newcomers (i.e., inexperienced) and students with some experience in their assigned GSoC projects. We also split students' participation into three periods: before, during, and after GSoC. We found that ≈82% of the OSS projects merged at least one students' commit. When only newcomers are considered, ≈54% of OSS projects merged at least one students' commit. Moreover, we found that ≈23% of the newcomers who contributed before kickoff started contributing before knowing they would be accepted. Furthermore, we found that the number of commits and code churn of experienced students made before kickoff was strongly associated not only with how much code they produced but also with how long they remained in their assigned projects during and after GSoC. After conducting S1 and S2, we defined the scope of the thesis and started phase II.

1.2.2 Phase II: Understanding students' motivations

In this phase, we conducted S3. We aimed at investigating students' motivations to engage in OSS via Summer of Code programs. We designed S3 for qualitatively exploring the answers to surveys (students and mentors) and interviews (students).

Study (S3): To understand what motivates students to engage in OSS via GSoC, we combined a survey (students and mentors) and interviews (students). Our findings show that students enter GSoC for an enriching experience, not necessarily to become frequent contributors. Our data suggest that while the stipends are an essential motivator, the students participate for work experience and the ability to attach the name of the supporting organization to their resumés. After these studies, we started the next phase.

1.2.3 Phase III: Building and assessing the usefulness of the theory of the engagement in OSS projects via Summers of Code

This phase consists of S4, in which we build and assess the usefulness of our theory.

Study 4 (S4): Grounded on multiple data sources, we built a theory that reveals the engagement process via Summers of Code. Our findings show that, when OSS projects participate in such programs, their tasks converge towards planning their participation, socially integrating the students, rewarding contributions and achievements, and management. Moreover, the interplay of OSS projects' engagement in Summers of Code and students' contribution produces outcomes, some of which either increase students' intrinsic motivation or are instrumental for achieving their goals.

It is worth mentioning that the studies S2, S3, and S4 correspond respectively to Chapters 3, 4, and 5. For each of these Chapters, we define more specific research questions to guide the studies better.

1.3 Scope

In this section, we detail the scope of this research and define some terms used in this thesis. When we refer to OSS projects, we are looking at community-based OSS projects [CFM08], that is, the projects that rely on the dedication of volunteers to be maintained, regardless of code license. Consequently, we *are not* talking about the OSS projects that mostly rely on companies' employees. In community-based OSS projects, mailing lists, issue trackers, and source code—in software repositories—are publicly available.

In this thesis, we are interested in Summer of Code programs. As mentioned, such programs occur in a few months during the summer vacations (North-hemisphere). The students' engagement via such programs does not count as an internship in most colleges. In this context, quitting such programs does not typically affect students' performance in their colleges. A variation of Summer of Code programs is the so-called Semester of Code programs. The most noticeable difference is that Semester of Code programs are accomplished in credit-bearing courses. In Semester of Code programs, failing the program may mean failing a class. (See Appendix A.1 for information on Summers of Code.)

In this thesis, we focus on the study of GSoC. We chose to study GSoC because it is best-known compared to other programs; it has been in operation since 2005; every year a large number of students from all over the world participate in it, and; it provides students with a comprehensive set of rewards, including participating in a well-known company's program, community bonding, skill development, personal enjoyment, career advancement, peer recognition, status, and a stipend [TCKH14].

Additionally, other Summer of Code programs use the same eligibility criteria as GSoC, such as Julia Summer of Code—which was part of GSoC in 2014, 2016-2019—, and Season of KDE, which gives preference to students who already applied for GSoC. In GSoC, OSS projects designate mentors to assist students, who must be enrolled in accredited schools, to reduce the learning curve associated with first contributions and help students to develop their GSoC project.

Furthermore, we noticed that the term engagement is not used precisely or consistently, even in the specific psychology literature [Sch13]. For this thesis, we adopted the following definition. Engagement is a broad construct that researchers study in three domains: cognitive, emotional, and behavioral engagement [NCM18]. When it comes to students, cognitive engagement refers to their internal investment in the learning process [NCM18]. Emotional engagement concerns questions regarding their feelings of belonging in a community such as a school or a workgroup [NCM18]. In this thesis, we focus on students' behavioral engagement, which refers to *active participation* concerning task accomplishments, following norms, and obeying rules [NCM18]. We refer to the behaviors that keep students actively involved with the tasks as engaging and disengaging otherwise.

Finally, we understand motivation as the psychological state [RHS06] that energizes individuals toward an end [RD00]. It is only after being motivated that students can engage in the tasks accomplished in Summer of Code programs such as coding to meet deadlines, post weekly updates in a blog, or communicate regularly with mentors. In this sense, students' motivation antecedes their engagement in the tasks.

1.4 Main Contributions

We claim that this thesis adds to the literature two novel contributions, which map onto the research questions presented in Section 1.1, and are related to the artifacts we crafted, presented in Section 1.2. The contributions mentioned earlier are:

- **Empirical identification of tasks that OSS projects accomplish to motivate students in Summer of Code programs**. Despite its practical relevance, to the best of our knowledge, no previous research has targeted such tasks to motivate students in Summers of Code. We collected several OSS projects' applications for being accepted in Google Summer of Code. For understanding what OSS projects accomplish in Summers of Code, we applied grounded theory procedures and proposed categories, which can be used for OSS projects to make a well-informed decision about joining such programs. Here, our novel contribution is a theory that empirically identifies the tasks OSS projects accomplish before, during, and after Summer of Code programs.
- **Empirical identification of factors that influence students' motivation, in general, to engage in OSS via Summer of Code programs**. Regarding engagements in OSS via Summer of Code programs, the literature has focused on quantitative evaluations of the contributions made by the students of a single OSS project [SLW12], and on the outcomes for the students who contributed to a few scientific OSS projects only [TCKH14]. In contrast, our study offers a unique emphasis on identifying empirically internal and external factors that influence students' motivation to engage in OSS via Summers of Code, including students who never engaged in OSS via such programs. We collected data from multiple data sources and employed several research techniques: surveys with various stakeholders, semi-structured interviews, and; quantitative assessments. Our main contribution is the empirical identification of the factors that motivate students to engage in OSS via Summer of Code programs.

1.5 Other Results

This research has resulted in scientific publications. In the following subsections, we present these results.

1.5.1 Published papers

During the Ph.D. program, we published papers related to this thesis' topics.

Paper 1. Silva, Jefferson O.; Wiese, Igor ; Steinmacher, Igor ; Gerosa, Marco A.: Pagamento Atrai Colaboradores de Longo Prazo ou Prestadores de Serviço? Uma Investigação Inicial sobre o Google Summer of Code 2014. In: XII Simpósio Brasileiro de Sistemas Colaborativos, SBSC. Salvador; BA, 2015, S. 8 (Qualis B3)

Paper 2. Silva, Jefferson O.; Steinmacher, Igor ; Gerosa, Marco A.: Os Programas de Engajamento em Software Livre Atraem Estudantes à Colaboração Voluntária? Um Estudo Empírico do Google Summer of Code. In: XIII Simpósio Brasileiro de Sistemas Colaborativos, SBSC. Porto Alegre; RS, 2016, S. 15 (Qualis B3)

Paper 3. Hammouda, I.; Lundell, B.; Madey, G.; Squire, M. (Eds.) In: Proceedings of the Doctoral Consortium at the 13th International Conference on Open Source Systems, Skövde University Studies in Informatics 2017:1, ISSN 1653-2325, ISBN: 978-91-983667-1-6, University of Skövde, Skövde, Sweden (Qualis B2)

Paper 4. Silva, Jefferson O.; Wiese, Igor ; German, Daniel m..; Steinmacher, Igor ; Gerosa, Marco A. How Long and How Much: What to Expect from Summer of Code Participants?. In: 33rd International Conference on Software Maintenance and Evolution, ICSME. Shanghai; China, 2017 (Qualis A1)

Paper 5. Silva, Jefferson O.; Wiese, Igor ; Steinmacher, Igor ; Gerosa, Marco A. Engajamento de Estudantes em Projetos de Software Livre: Uma Análise do Google Summer of Code. In: 31th Brazilian Symposium of Software Engineering, SBES; Fortaleza; CE, 2017 (Qualis B2)

Paper 6. Silva, Jefferson O.; Wiese, Igor ; Steinmacher, Igor ; Gerosa, Marco A. Google Summer of Code : Student Motivations and Contributions. Journal of Systems and Software (JSS), 2019 (Qualis A1)

1.5.2 Submitted papers

Paper 7. Silva, Jefferson O.; Wiese, Igor ; Steinmacher, Igor ; Gerosa, Marco A. Towards a Theory of the Engagement in Summer of Code Programs. Computer Supported Cooperative Work (CSCW) (Qualis A1)

1.5.3 Advising

Capstone Project. Emerson Ueti e Leandro Cabral. Uma Investigação Quantitativa da Retenção no Projeto Airavata da Apache Software Foundation. 2017. Trabalho de Conclusão de Curso—Pontifícia Universidade Católica de São Paulo, PUC-SP. Advisor: Jefferson de Oliveira Silva

1.6 Funding

Project. Software engineering newcomers barriers and motivations to onboard open source software projects. FAPESP. Nº do processo FAPESP: 2015/50084-1. Coordinators: Marco Aurélio Gerosa, Daniel M. German

1.7 Thesis Organization

This thesis is organized as follows: in Chapter 2, we present an overview of the state-of-the-art of the self-guided contribution to OSS projects and contributions to OSS projects via Summer of Code programs. In Chapter 3, we present our quantitative study on several aspects of the students' contributions to their assigned GSoC projects. In Chapter 4, we present our exploration of the students' motivations to engage in OSS via Summer of Code programs. In Chapter 5, we present our theory of engagement in OSS projects via Summer of Code programs. In Chapter 6, we highlight the main findings and offer some guidelines to OSS projects and Summer of Code administrators. In Chapter 7, we conclude this research and present some thoughts on future works.

Chapter 2

Related Works

Communities that depend on collaboration require a pool of volunteers to function adequately. A challenging task for most communities is to engage new volunteers to replace the old ones, who stop contributing eventually. When dropout rates are higher than new volunteers' engagement rates, communities face crucial sustainability issues. Such communities need a steady stream of volunteer newcomers to maintain service levels. Although engaging newcomers is a formidable task for several communities, some factors are known to motivate newcomers, making the service experience more attractive.

In this Chapter, we present an overview of the existing literature on the factors related to newcomers' engagement in communities in general, online communities, and Open Source Software (OSS) communities. To ease their identification, for each work, we highlighted the factors related to newcomers' engagement in *italics*.

2.1 Newcomers' Engagement in Communities

Failing to engage newcomers is a widespread drawback that affects several communities, including community organizations, faith-based groups, and government volunteering agencies around the globe [Wit13]. The United States Bureau Labor of Statistics provided some evidence on this phenomenon by showing that the volunteering efforts of the general North-American population declined steadily from 2011 to 2015. As a countermeasure, researchers often study the factors that contribute to increasing newcomers' engagement levels.

For example, in 2013, the Youngstown's fire department (LA, USA) experienced a shortage of volunteers firefighters to the point that it became difficult to respond to emergency calls.[1] To understand what motivated new volunteer firefighters in suburban fire departments to engage and keep volunteering, Smith [Smi14] conducted a qualitative case study. The author [Smi14] investigated fire departments in three towns. Smith [Smi14] found that *offering an environment with excellent interpersonal relationships*, reinforcing their *desire to serve and help people*, and *reducing the bureaucracy of departmental requirements* were the key influencers of volunteering new firefighters.

[1] http://kpel965.com/youngsville-looking-for-firefighters

In many countries, a significant challenge is to keep a stable national blood supply in order to maintain blood stocks at safe levels [Don13]. For example, in the United Kingdom, The Guardian UK news reported that 40% fewer newcomers came forward to give blood in 2015 compared with the decade ago, which left health chiefs worried about blood shortages.[2] For mitigating this problem, Charbonneau et al. [CCC16] surveyed blood donors to understand the reasons they lapsed or reduced their donations' frequency. The authors [CCC16] found that, besides the widely known claims of lack of time and fear of needles, several blood donors mentioned the annoyance at the need to complete a health questionnaire at each donation. The authors [CCC16] reasoned that the *institutional simplicity for volunteer blood donations* should be addressed for engaging new donors.

In some cases, even the aids of governments and the private sector may not suffice to engage newcomers. As an example, The Guardian newspaper reported that despite substantial investments from the English government and private sector, many communities—in this case, charities—struggled to engage newcomers.[3] For addressing such problems, Veitch et al. [VHH$^+$99] studied an approach to engage and retain a qualified medical workforce in rural, remote communities in Australia. The authors [VHH$^+$99] developed collaborative planning that integrated local community support with government and professional strategies. They concluded, among other things, that *community members' attitudes and expectations* regarding medical services played an essential role in the engagement of medical practitioners.

In this thesis, we are interested in the engagement of newcomers in OSS communities, which are instances of Open Collaboration Communities [FL13]. Choi et al. [CAKL10] claim that, although online groups are similar to offline ones in many aspects, one cannot automatically generalize findings from offline groups to online groups due to their distinct nature. While face-to-face interactions tend to strengthen social ties among members in offline communities, the distributed nature of peer production in online communities tend to cause the opposite effect, weakening social ties among members [CAKL10]. Moreover, Open Collaboration Communities depend on volunteer contributions, which makes it easy for volunteers to leave.

Identifying factors related to newcomers' engagement in online Open Collaboration Communities is still a relevant problem that needs to be further explored. In this sense, the next Sections present an overview of the body of literature on the factors related to newcomers' engagement in online communities (Section 2.2) and, more specifically, engagement in OSS communities (Section 2.3).

2.2 Newcomers' Engagement in Online Communities

In face-to-face settings, several researchers identified a diverse set of motivations for newcomers' engagement in such offline communities. As with offline groups, online communities provide their members with opportunities for information sharing, learning, social support, and producing knowledge for the public good [RK09].

The motives for engaging in online communities are as diverse as of offline ones [Pre00]. New-

[2] http://www.theguardian.com/uk-news/2015/jun/05/sharp-drop-new-blood-donors-uk-stocks-at-risk
[3] http://www.theguardian.com/society/2008/sep/10/voluntarysector.volunteering

comers engage in online communities by self-interest, for example, seeking *reputation enhancement* [CSK99], or to *gain personal benefits* [FHFM04]. In some cases, newcomers act selflessly and appear to engage in the online communities by *altruism* [PS07]. Moreover, when newcomers identify closely with an online community, they feel an obligation to *give back to other community members* [WF05]. Contrarily, in some cases, despite long-term affiliation with an online community, participants feel no obligation to other members [PNA04]. Furthermore, *low barriers for placing contributions* played an important role for the newcomers' engagement in communities [Cif03].

Online communities need to worry about newcomers' engagement in the communities from the beginning. Despite the limited direct control of individual members' actions, online communities should consider factors in their design to achieve established goals, which include attracting and engaging newcomers [RK09]. Resnick and Kraut [RK09] presented a schema comprised of eight categories, each containing several factors that allegedly increase newcomers' engagement. We present them as follows:

(*i*) Community structure, including *community size*, *degree of homogeneity of member interests*, whether there is a *subgroup structure*, and *whether membership is recruited through existing social ties*;

(*ii*) Content, tasks, activities in the community, which includes opportunities for *self-disclosure* (e.g., in user profiles), whether there are *newcomers' welcoming activities and safe spaces for newcomers' exploration*;

(*iii*) Select, sort, and highlight things, which concerns with *newcomers finding things that are best for them*;

(*iv*) External communication, which represents the facilities provided so that *newcomers can invite friends or forward content to them*;

(*v*) Feedback and rewards, which includes *providing feedback*—tell people whom others reacted to their engagement in the community; and *rewarding and sanctioning newcomers*, giving or taking away something newcomers value, in response to actions they take;

(*vi*) Roles, rules, policies, and procedures, which includes *articulating different roles*, such as 'welcomers' or dispute handlers; *providing guidelines about how members should behave*; *establishing procedures for decision-making and conflict resolution*;

(*vii*) *Access controls*, which places limits on who can join the community and what actions they can take;

(*viii*) Presentation and framing, which includes *communication choices*, *ways of framing what the community is* and *what happens in the community*;

Farzan et al. [FDKP11] compared alternative ways of designing online communities to increase member retention. By conducting two experiments within a Facebook game application, the authors found that online communities designed for *social presence*-the virtual representation of other members on the screen—that encouraged the formation of shared-identities and common-bonds, in isolation or combination, helped to increase the return rate of members.

Faraj and Johnson [FJ11] investigated for the presence (or absence) of structural network exchange patterns in 5 online communities at a group level. The authors found two network exchange patterns that suggested member retention: *direct reciprocity* (i.e., replying to respondents); and *indirect reciprocity* (i.e., providing communication that supports generalized exchange).

Panciera et al. [PHT09] investigated the production of Wikipedia editors to understand how the work of the regular editors (i.e., Wikipedians) compared to the work of the less-prolific ones. The authors found out that the initial activity of Wikipedians set them apart from the majority of sporadic editors. Moreover, the authors revealed that Wikipedians do not do more work, better work, or more community-oriented work over time, suggesting that *the customization of the initial contributors' experience* can lead to higher member retention.

To investigate the process of involvement in knowledge production online communities, Dejean and Jullien [DJ15] used a survey to collect responses of 13,386 Wikipedia users (≈ 12% of regular contributors). The authors challenge the idea that becoming a core member starts with peripheral contributive activities and results from a continuous learning process, as proposed by LPP theory. More specifically, the authors investigated the link between the first contribution and the level of contribution reached (i.e., retention). They found that *future major contributors started with a 'big' contribution* (e.g., writing or rearranging an article), as opposed to minor contributors. Moreover, they found that *major contributors made their contribution within one month after discovering Wikipedia*. The authors suggested that the effort invested in the first contribution should be viewed as a sign of a willingness to join the community.

Morgan et al. [MBWS13] identified common challenges faced by Wikipedia newcomers such as isolation, intimidation, and limited opportunities for socialization. The authors [MBWS13] were particularly concerned about the potential impact of the challenges on the female editors. Thus, the authors [MBWS13] developed a website called 'Teahouse,' which was intended to be a safe space for newcomers to engage in Wikipedia. The website consisted of a simple set of tools, norms, and procedures. Within Teahouse, newcomers could have their questions answered by friendly Wikipedians. The authors [MBWS13] surveyed Teahouse users and compared the engagement levels to the engagement of regular Wikipedia newcomers. The authors [MBWS13] found that when newcomers used the Teahouse website, their engagement level was higher, meaning that they made more edits. The authors [MBWS13] highlighted two strategies for increasing newcomers' engagement: offer *early support*, such as contacting the newcomers who stopped contributing, ideally within the first 24 hours, alerting them to opportunities and human interaction; and provide newcomers with *opportunities for socialization*, such as non-confrontational forums to facilitate social or educational discussions.

Halfaker et al. [dSCK$^+$11] described the Article Feedback Tool, a system designed to elicit lightweight contributions from Wikipedia readers. The tool allowed readers to submit—via standard web interface— four types of feedback (problem, question, praise, suggestion) about the encyclopedia articles to editors. The authors [dSCK$^+$11] conducted three field experiments on the English Wikipedia. For each experiment, the tool interface was loaded on a random sample of articles. Moreover, all readers who viewed the articles were randomly assigned. Their findings [dSCK$^+$11] suggested that newcomers' engagement can be increased by *making the means of contribution more* *prominent* without sacrificing quality with simple and straightforward contribution types such as comments and feedback.

Although these studies are related to our research, they focus on online communities in general. In the next Section, we present the studies that specifically deal with newcomers to OSS projects.

2.3 Newcomers' Engagement in Open Source Software Projects

In this Section, we review works on newcomers' engagement in OSS projects. We adopted the term projects instead of communities because the literature on volunteer contributions to OSS projects usually does so. Although contributing to online communities shares many similarities with the contribution to OSS projects, there are substantial differences. Software development is a knowledge-intense activity that often requires high levels of domain knowledge, experience, and intensive learning by those contributing to it [VSL03]. When compared to online communities, the newcomers' contribution experience in OSS projects can be more arduous, since in many cases, only developers actively involved in a project's development can understand its architecture well enough to contribute code effectively [VSL03].

In the following subsections, we present two ways of contributing to OSS projects. In Section 2.3.1, we summarize the works related to the engagement in OSS projects when there are no contracts binding newcomers to the projects, which we refer to as self-driven engagement. In Section 2.3.2, we present the literature related to the engagement in OSS projects via Summer of Code programs.

2.3.1 Newcomers' self-driven engagement in OSS projects

Newcomers are often on their own to choose an OSS community, select a specific OSS project, learn contribution norms, understand the project's architecture, identify issues, besides being proactive in asking for help [Duc05]. In this context, newcomers need to possess qualities such as a good measure of self-efficacy, some socialization tactics, and persistence [Duc05]. We refer to this type of *active participation* as self-driven engagement in OSS projects.

Unveiling the nature of the self-driven engagement in OSS projects stimulated the curiosity of scholars in a variety of fields, including psychology, anthropology, economics, law, computer science [Ros04]. As a result, motivation, ideology, social ties with team members, incentives, rewards, and project characteristics are only a fraction of the perspectives adopted to study the engagement in OSS projects. We present some of the perspectives in the following.

Motivation

A conventional understanding among researchers seems to be that motivation refers to the psychological needs that require satisfaction [DKR99]. Such needs can be acquired through the influence of the environment, such as culture and lifestyle, or they can be innate [Mas12]. In general, theorists study motivation to understand its link to behavior, using several constructs to explain how motivation influences choice, persistence, and performance in a broad range of domains, including software engineering [BB10]. As with other practitioners, software engineers are influenced by their motivational state, which is determinant to the success or failure of the projects [BBH+08]. It is out of the scope of this thesis to provide an exhaustive systematic review of any motivational theories. Other motivational theories include goal-orientation theory (see [Nic84]), social exchange theory (see [Bla64]), and reinforcement theory (see [Ski53]).

Motivation is typically classified into three types: intrinsic motivation, extrinsic motivation, and internalized-extrinsic motivation [SDPR92]. (There is also a fourth type called amotivation, in which individuals have no interest in goal-directed activities. In this research, we consider that students could not be amotivated since they joined the program, and we assumed that this action was voluntary.) Although we present works using this classification, we stress that there is no single motivation for a newcomer's decision to engage in an OSS project [LW05]. In this sense, motivations with elements of both intrinsic and extrinsic are not rare [LW05]. We used the layout presented by von Krogh et al. [vKHSW12] to classify the works into each category.

Intrinsic motivation. Researchers started using the concept of intrinsic motivation to explain 'spontaneous' behaviors, which occurred with no need for reinforcement [SDPR92]. Intrinsically-motivated behaviors do not require any 'rewards' other than those obtained from the satisfaction of performing them, such as when individuals perform a task for its fun or because it is challenging [DKR99].

A widely known intrinsic motivation is to accomplish a task for the learning involved [SDPR92]. Learning is a hybrid form of motivation because one can "*learn for the learning sake*" or because it is instrumental to achieving external rewards [SDPR92]. In the context of self-driven engagement in OSS projects, Lakhani and von Hippel [LV03] explored how the valuable chores are organized in an Apache web project, as an attempt to understand why project participants engage in the service for free. The authors [LV03] found that 98% of the effort invested in accomplishing the chores returned to the participants in the form of *learning*. With this result, the authors [LV03] concluded that *learning* is the main driver for volunteer engagement.

The capacity for having fun while contributing to OSS projects is also a frequently studied engagement factor. For example, Luthiger and Jungwirth [LJ07] employed surveys to assess the importance of fun as a motivator for software developers to engage in OSS projects. By surveying 1,330 OSS developers, the researchers found out that *fun motives* accounted for about 27 to 33 percent of OSS developers' motivation. More specifically, their findings revealed that the *fun* factor had a significantly positive effect on both the number of hours spent on a project as well as the developers' intention to engage in the OSS projects in the future.

In some cases, volunteers are software companies' employees, not necessarily full-time volunteers to OSS projects. Even then, enjoyment plays a significant role in the engagement. For example, Lakhani and Wolf [LW05] studied OSS developers' motivations to contribute to OSS projects. The authors [LW05] surveyed 684 OSS developers in 287 OSS projects. They found that there were no significant differences in motivations between paid and volunteer contributors. Approximately 40% of their sample consisted of developers who were paid to contribute to the studied OSS project. Paid contributors consisted of the developers who received direct compensation for their work in OSS projects and the developers who coded for OSS projects with the permission of their supervisors while working as software engineers in companies. Their findings included that the *enjoyment-based intrinsic motivation*, such as *how creative a person feels when working on the project*, the *intellectual stimulation* derived from writing code, and *improving programming skills* were top motivators for engagement.

Heterogeneity of motivations has been found to be an engagement factor. For example, David and Shapiro [DS08] studied the motivation to contribute to OSS projects of developers involved in community-based production. In an extensive web survey, the authors [DS08] collected data on 1,473 respondents using a questionnaire, classifying the responses into 'motivational profiles' by hierarchical cluster analysis. Their findings provided evidence that heterogeneity of motivation was a crucial feature of OSS development. The authors [DS08] concluded that OSS projects that *mobilized individuals with different motivations*, *taking advantage of each motivational profile* had greater success engaging and retaining developers.

Furthermore, several empirical studies found intrinsic motivation factors that played a significant role in motivating OSS developers, such as: *ideology* (e.g., [DAW03, DS08, Gho05]); *altruism* (e.g., [BSS07, Gho05, HPS03]); *kinship amidity* (e.g., [DS08, HO02, HNH03]); and *enjoyment and fun* (e.g., [Ben01, Hem02, HNH03]).

Extrinsic motivation. In contrast, extrinsically-motivated behaviors are the pursuit of external rewards or the consequences derived from their performance [SDPR92]. In the OSS literature, Tirole and Lerner [TL02] proposed a generalization for OSS developers' motivation as some form of *self-interest calculus*. They studied the signaling behavior of OSS developers. The authors [TL02] used four cases and descriptive statistics to highlight the extent to which economics can explain engagement in OSS projects. They concluded that OSS developers are motivated mainly by *career concerns*, an argument that can be reconciled with the economic literature. As the authors posited: "*A programmer participates in a project, whether commercial of F/OSS, only if she derives a net benefit (broadly defined) from engaging in the activity. The net benefit is equal to the immediate payoff (current benefit minus current costs) plus the delayed payoff (delayed benefits minus delayed costs).*"

Lakhani and Wolf [LW05] examined the link between financial motives and engagement in OSS projects. They found that sponsored contributors dedicated 17.7 hours per week on all OSS projects the contributors participated in, while volunteers dedicated 11.7 per week. The difference between the groups was found to be significant. Moreover, the most commonly cited extrinsic motivation factors are *career building* (e.g., [TL02, Rie07, YJC07, LW05, Hem02, HO02, RHS06, HNH03, XJS09]), and *stipends* (e.g., [LW05, HO02, HNH03, LJ07]).

Internalized-extrinsic motivation. Individuals can undergo a motivation internalization process. More precisely, they can internalize extrinsic motivations so that their motivations are self-regulated rather than externally imposed [RHS06]. For example, Lattemann and Stieglitz [LS05] aimed at identifying factors that sustain the motivation of OSS community members over the entire life cycle of an OSS project. The authors conducted a systematic review of empirical studies to identify mechanisms that work as OSS governance tools. They found that *social control mechanisms* like *trust*, moral concepts by *rituals and ceremonies*, or *group evaluation processes*, such as *peer reviews* (i.e.,*peer reputation*) among community members become a crucial part in the government of OSS communities.

Spaeth et al. [SHK08] studied Freenet, an OSS project for peer-to-peer software. The authors [SHK08] wanted to understand why OSS developers contribute to OSS projects (i.e., public good) freely. Using concepts of resource mobilization theory along with the private-collective model, the authors [SHK08] found that the production process of knowledge in an OSS project yields communal resources as a by-product (i.e., *reputation*, *control over technology*, and *learning opportunities*) that reward its contributors. That is, although developers freely collaborate in the creation of the public good, each developer is individually rewarded by *communal resources*.

Several internalized extrinsic motivation factors were found to be important, such as *reputation* (e.g., [DR87, Gho05, HO02]); *reciprocity* (e.g., [BL01, Hem02, LW05]); *learning* (e.g., [HK03, TL02, YJC07]); and *own use value* (e.g., [LV03, OR07, Ray99, DAW03]).

Project characteristics

Zhou and Mockus [ZM12] worked on identifying the newcomers who were more likely to remain contributing to the project to offer active support for them to become long-term contributors. They found that the *individual's motivation* (*willingness*, in the authors' words) and the *project's climate* were associated with the odds that an individual would become a long-term contributor.

Chegalur-Smith, Sidorova, and Daniel [CSSD10] analyzed whether *codebase size*, *project age* and *niche size* (a measure borrowed from ecology) influenced project attractiveness. They found that these three characteristics influenced the project's ability to attract and retain developers.

Contribution barriers

Any skilled newcomer who wants to contribute to an OSS project can access the current code, choose (or report) an issue, address it, and submit a patch. For this contribution to be successfully carried out, newcomers need to overcome several hurdles. An unsuccessful first attempt to place a contribution may frustrate newcomers, making them wait a long time before a second try. Not rarely, several newcomers give up before trying again [FN09]. Such hurdles are known in the literature as contribution barriers [Ste15]. For mitigating such barriers, Steinmacher [Ste15] worked on identifying them, and then, creating a contribution barrier model. Additionally, the author [Ste15] proposed a tool called FLOSScoach,[4] to assist newcomers in achieving successful contributions.

Joining process

A common way to model the joining process in OSS communities is the so-called onion model presented by Nakakoji et al. [NYN$^+$02] and Ye and Kishida [YK03]. The onion model is a general layered structure that organizes OSS member roles. The authors in both works studied four different OSS projects and found that each member may have one of the eight roles presented in Figure 2.1. They claim that the onion's layers represent the path that a user ideally runs through to arrive at the project's core. Although they recognize that "not all members want to and will become core members," they create a process a developer needs to follow to contribute to a project. They

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

mentioned that some OSS communities must create a social mechanism that encourages *peer support* and facilitates community members to aspire to the core member's status.

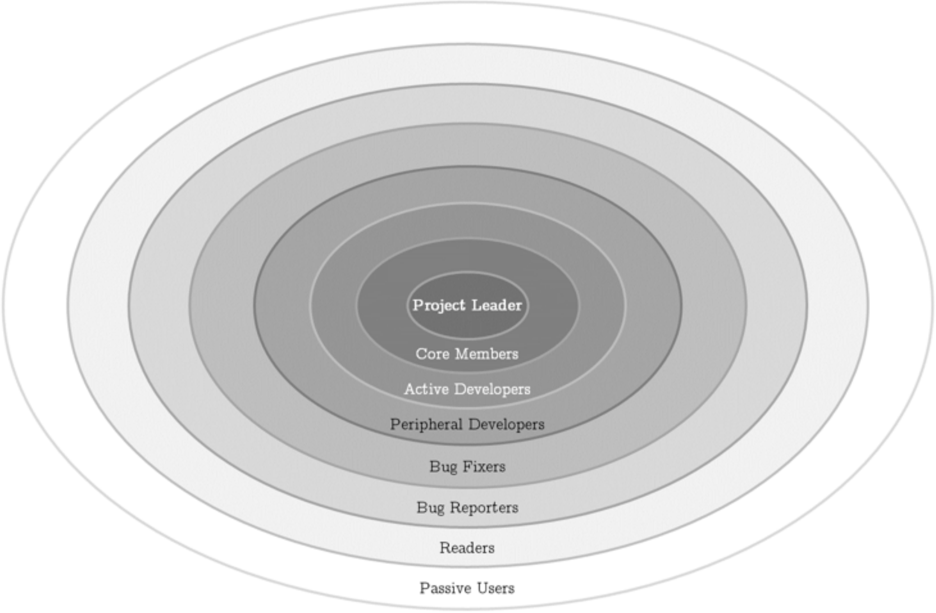


Figure 2.1: *The Onion Model [NYN+ 02]*

von Krogh et al. [VSL03] aimed at creating a theory that explains how newcomers voluntarily engage in OSS projects when contribution costs vary. The authors [VSL03] examined the joining behavior in the Freenet project and proposed a theoretical construct called 'joining script.' The authors created this construct by analyzing interviews with developers, emails, source code repository, and documents from the project. The joining script is defined as the level and type of activity a newcomer goes through to become a member of a community. They concluded that *participants who behaved according to the 'joining script'* (level and type of activity) are more likely to be granted access to the developer community than those participants that did not.

von Hippel and von Krogh [HK03] focused on understanding why OSS developers write code to the public good. The authors [HK03] proposed that volunteer contributions to OSS projects are a compound of two organizational science models. The first model, named 'private investment,' grants innovators rights to their innovations via intellectual property law mechanisms such as patents, copyrights, and trade secrets. Nevertheless, the authors [HK03] explain that although this model ensures proper incentives to innovators, society does not benefit from the knowledge yielded by these innovations. The second model, named 'collective action,' requires that contributors relinquish control of the knowledge they have developed and make it a public good. While this model ensures that society will benefit from the yielded knowledge, it generates 'free-riding' opportunities, especially when potential contributors have the option to wait for others to contribute and free-ride on what they have done. Thus, the authors [HK03] proposed that volunteer contributions to OSS can be understood by the 'private-collective' model, in which contributors retain private benefits from their work processes such as *learning*, *enjoyment*, and *status*, while simultaneously gaining benefits associated with community engagement.

Socialization

Ducheneaut [Duc05] explored the dynamic strategies and processes by which newcomers are progressively socialized (or not) into the Python community. The authors analyzed the trajectories of 284 participants and made an in-depth analysis of the socialization history of one successful newcomer. They combined ethnography and software specially designed to visualize and explore the interacting networks of human and material resources incorporated in the email and code databases of OSS. For newcomers to contribute (and keep contributing), it is necessary to follow rites of passage before proposing solutions to OSS projects, which include *building identities*, *demonstrating technical expertise*, and *actually developing the code itself*.

Project attractiveness

Santos et al. [SKKP13] defined a theoretical cause-effect model for attractiveness as a crucial construct for OSS projects. Attractiveness represents the forces that OSS projects put forward in order to foster contributions from newcomers [Ste15]. The authors [SKKP13] proposed their typical causes (*license type*, *intended audience*, *type of project*, *development status*); indicators (*hits*, *downloads*, *members*); and consequences (*# of open tasks*, *time for task completion*). They tested the model with data from more than 4,000 projects and found that project conditions and characteristics–such as *license restrictiveness* and *their available resources*–directly influence the attractiveness.

Meirelles et al. [MSM+10] built upon Santos' model, inserting source code as one of the typical causes. They observed the influence of structural complexity and software size (lines of code and number of modules) on project attractiveness. Their results indicated that *structural complexity* has a negative influence, and *software size* has a positive influence on attractiveness.

Engagement theories

The Legitimate Peripheral Participation theory is frequently used to explain the process of how apprentices become experienced members (i.e., onboard) of a community of practice [LW91]. When applied to OSS, the theory describes that newcomers begin their involvement by observing experienced members. After a while, apprentices (i.e., newcomers) become in charge of simple tasks (e.g., reporting issues or fixing bugs), which are nonetheless valuable to the community (i.e., OSS projects). This process allows newcomers to become acquainted with the OSS projects, and contribution norms, culminating in the emergence of frequent contributors.

Fang and Neufeld [FN09] built upon Legitimate Peripheral Participation theory to understand the motivation of developers to contribute sustainably. Results from qualitative analyses revealed that *initial conditions to participate* did not adequately predict long-term engagement, but that *situated learning* and *identity construction* behaviors were positively linked to sustained engagement.

2.3.2 Students' engagement in OSS projects via Summers of Code

Summer of Code programs are becoming a common initiative to bring more contributors to OSS (e.g., Google Summer of Code, Julia Summer of Code), and to increase diversity (e.g., Outreachy, Rails Girls Summer of Code). Figure 2.2 illustrates this type of engagement in OSS projects.

Given Summer of Code apparent success, some researchers targeted such programs to understand

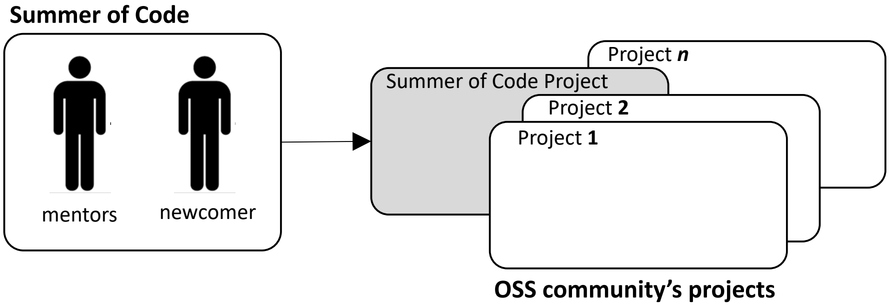


Figure 2.2: *Students engagement in OSS projects via Summers of Code*

engagement and retention. For example, Schilling et al. [SLW12] used the concepts of Person-Job (the congruence between an applicant's desire and job supplies), and Person-Team (the applicant's level of interpersonal compatibility with the existing team) from recruitment literature to derive factors that predict the retention of GSoC students in the KDE project. The authors [SLW12] worked with a sample consisting of 80 former GSoC students. Using a classification schema of prior code contributions to KDE, they found that intermediate levels (4-94 commits) and high levels (>94 commits) of *previous development experience* in KDE were strongly associated with engagement.

Trainer et al. [TCH14] conducted a case study of a bioinformatics library called Biopython to investigate the outcomes of GSoC (for this project only). For collecting data, the authors [TCH14] interviewed the top 15 Biopython developers ranked by the number of commits to the master branch on GitHub. They identified three positive outcomes for the students: the addition of new features to the project's codebase, finding that 50% of the GSoC projects were merged to codebase; training, finding that the students learned new software engineering skills, such as testing; personal development, reporting that students use engagement in GSoC for career advancement. Moreover, the authors [TCH14] found that the *relationship of the students with their mentors* enhanced students' commitment and became a substantial factor that caused students to come back as contributors after GSoC, especially when *students receive requests from their mentors.*

Trainer et al. [TCKH14] studied to understand the range of outcomes that Summers of Code produce and the underlying practices that lead to these outcomes. The authors performed a multiple case study of 22 GSoC projects within six different scientific software projects across three different domains. Through interviews, they found that GSoC facilitated the creation of *strong ties between mentors and students* and that some students kept contributing as mentors, active contributors, and users. In particular, they mentioned that 18% of the students (n=22) became mentors in subsequent editions. We are not aware of other studies, explicitly targeting newcomers' retention in OSS projects for other Summers of Codes.

2.4 Final Considerations

In this Chapter, we presented studies on engagement in offline and online communities. We compiled factors that increased the odds of the engagement in OSS projects when newcomers decide to contribute by their means (i.e., self-driven engagement) and when they contribute via

Summer of Code programs.

We claim that the conditions for engaging in OSS projects via Summers of Code differ from the self-driven engagement. Several onboarding barriers, such as the ones presented by Steinmacher [Ste15], can be mitigated—or eliminated—in a Summer of Code setting. While in self-driven engagement, contribution to OSS projects is ultimately an individual endeavor, in Summers of Code, students are assisted by a team of allegedly experienced developers. Students are bound to Summers of Code by a contract for some time. Other than providing students with meaningful learning, Summers of Code may include other goals, such as promoting gender diversity. Furthermore, students engage in a project designed to fit their expertise level since such programs focus on their learning.

In this way, we claim that self-driven engagement and engagement via Summers of Code are fundamentally distinct since self-driven newcomers cannot rely on these—theoretically—favorable conditions. Our next study targeted at quantifying students' contribution to OSS/GSoC projects.

Chapter 3

Quantitative Analysis of Students' Contributions

In this chapter, we performed studies S1 and S2, as depicted in 1.1. In these studies, we wanted to analyze quantitatively several aspects of the contribution to OSS projects via Summer of Code programs. These studies can offer objective measures such as code churn metrics (how much they contribute to their assigned Summer of Code projects) and frequency metrics (how long they remain contributing to the projects).

The analysis of such metrics can be used to counterbalance the analysis of the inherently subjective perceptions of students and mentors, which is crucial in understanding students' motivations. For example, while students may perceive themselves as frequent contributors to their assigned Summer of Code projects, a quantitative analysis can provide us with the specific time-period (in days) that students contributed to the projects.

The content of this chapter was accepted for publication at ICSME 2017 [SWG+17]. It is worth mentioning that we made several adaptations to the published text for higher cohesion and readability.

3.1 Research Questions

We established the following research questions (RQ) to guide our answers.

RQ 3.1: How much code do Summers of Code students contribute to the codebase?

RQ 3.1a: How many commits/code churn in the codebase are contributed by the students?

RQ 3.2: How long do students contribute before and after participation in Summers of Code?

RQ 3.2a: What was the students' participation before and after GSoC?

RQ 3.2b: Is previous participation associated with students' retention?

3.2 Research Method

In this section, we present the method for data collection and analysis. For data collection, we searched for the GSoC students' assigned projects and mined repositories. For data analysis, we used descriptive statistics and statistical tests. Figure 3.1 depicts the research method we use to answer our RQs.

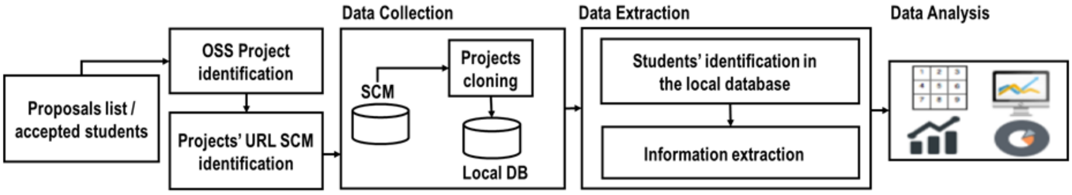

Data Collection
Data Extraction
Data Analysis
Proposals list / accepted students
OSS Project identification
Projects' URL SCM identification
SCM
Projects cloning
Local DB
Students' identification in the local database
Information extraction

Figure 3.1: *Method used to collect and analyze the students' interaction with the assigned Summer of Code projects*

Data Collection

The data collection phase involved several steps since Google only published (at the time of the data collection) the names of the organizations and accepted candidates, which made it hard to determine the specific project a given participant worked for. For example, Google informs that participant Apache Software Foundation accepted John Doe, but, generally, there is no information on which Apache project John has worked on. As the collection and verification of each student project is a laborious and time-consuming task, we limited our analysis to the GSoC 2013 to 2015 editions. We counted ≈3,100 distinct accepted students for these editions.

We randomly sampled 866 students, which offers a confidence level of 95% and a margin of error of 5%. We manually searched for the students assigned GSoC projects in source code management systems (SCM) by using their names and the project description provided by GSoC. In most cases, the projects were hosted on GitHub. We determined that we found their assigned projects when we had clear evidence linking the projects in the SCM with the students' information and the organization (e.g., when the projects' descriptions in the SCMs matched those of the GSoC's projects, or when we found web links in the students' blogs to the projects). We found the projects of 406 students (out of 866), all of which were hosted on GitHub.

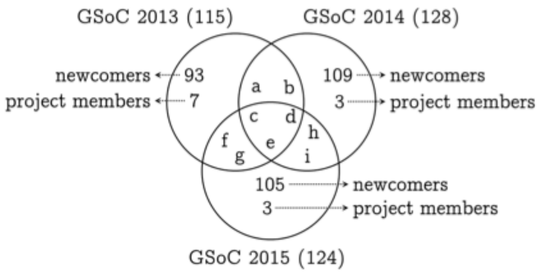
The next step was to identify the students' IDs in the project logs. First, we used CVSAnalY (MetricsGrimoire)[1] to extract information from Git repositories and store it in a local database. The database includes information not only about the project commits but also about the contributors. Second, we searched for all the IDs that students' might have used. We used the students' names and emails (or combinations) to decide if the IDs belonged to the same student. Based on this, to identify the students, we applied common disambiguation heuristics, such as the ones presented by Wiese et al. [WdSS^{+}16].

For instance, when the IDs were composed of the combination of the initials of the students'

[1] http://metricsgrimoire.github.io/CVSAnalY/

first name with their full last names, or when the IDs were composed of the students' names initials and these initials were used as the students' IDs on GitHub. As a result, we identified the IDs of 367 students (out of 406) within the projects.

Additionally, for all students in our sample, we verified whether they participated in previous GSoC editions. Figure 3.2 illustrates in a Venn diagram students' participation per GSoC edition. It is worth mentioning that a student may have participated in more than one edition, but in our sample, this student may appear in only one edition. Thus, for clarification, Figure 3.2 depicts in the Venn diagram how many students of our sample participated in two or three GSoC editions with letters (a-i) and we caption their meaning below the figure. We summarize students' participation as follows: 32 (8.6%) participated in 2 editions, 15 (4.1%) participated in all 3 editions, 13 (3.6%) participated in one edition but were already project members, and the remaining 307 (83.7%) are newcomers who participated in one of the three editions analyzed. For each edition, we include the total students in parenthesis. In addition, we found 16 students who participated in GSoC editions before 2013.



a = 5 students (from our 2013 sample) also participated in GSoC 2014
b = 5 students (from our 2014 sample) also participated in GSoC 2013
c = 5 students (from our 2013 sample) also participated in GSoC 2014 and 2015
d = 5 students (from our 2014 sample) also participated in GSoC 2013 and 2015
e = 5 students (from our 2015 sample) also participated in GSoC 2014 and 2015
f = 5 students (from our 2013 sample) also participated in GSoC 2015
g = 5 students (from our 2015 sample) also participated in GSoC 2013
h = 6 students (from our 2014 sample) also participated in GSoC 2015
i = 6 students (from our 2015 sample) also participated in GSoC 2014

Figure 3.2: *Number of students by year of participation*

As the last step, for every student in our final working sample, we counted the number of participations as a student and as a mentor, using the list published by GSoC and considering the editions of GSoC 2005 (first edition) to 2015. We used the student's name and the GSoC project name as a matching criterion. That is, when we had a match with a student name as both student and mentor, we analyzed both: whether the GSoC project of the mentor was related to the GSoC project of the student, and; whether the year of participation as a student was earlier than that as a mentor.

Data Analysis

To analyze the data, we split the students' participation in terms of contributions to GSoC into three periods: before, during, and after GSoC. We used the official timelines (i.e., start and end dates) to classify the commits in each period.

Although students can engage in an OSS project in different ways, such as opening issues, fixing bugs, improving documentation, or promoting events, we use the term *students' participation* to refer to their commits (and consequently code churn) to the SCM. Thus, participation before and after GSoC refers to the commits that the students performed before GSoC kickoff and after GSoC ended, respectively. *Students' participation interval* refers to the time in days that a student contributed (i.e., committed). For instance, if a GSoC edition started on the 15th day and a commit was performed on the 10th day of the same month and year, then the participation interval of this contribution is 5 days before kickoff. Additionally, we use the concept of *contribution days* (i.e., distinct commit dates). For instance, if a student performed 3 commits on the 10th day, again 5 days before kickoff, then the contribution days' count before GSoC is 1 (i.e., one distinct commit date before GSoC).

(RQ3.1) To test whether a specific commit was merged to the codebase, we compared each of the students ' commits' Secure Hash Algorithm (SHA)—which uniquely identifies all commits—to the commits' SHAs belonging to the codebase, and grouped them by participation period. The number of students' commits in the codebase was obtained by counting the number of commits in each group.

To determine how much code the students added to the codebase, we used the git-log tool, which creates a log file for the GitHub cloned projects, containing the commits' SHA, authors' name, and how many lines were added and removed for each file in a commit. Next, for every commit, we calculated the code churn per commit and recorded it in the database.

To test whether there are statistical differences in the number of commits among the participation periods, we used the Wilcoxon Signed-Rank test, which can determine whether the corresponding data population distributions are statistically equivalent for non-normal distributions. The null hypothesis is that the commits' distribution in each period tuple—m(before-during), (during-after), (before-after)—are statistically equivalent. If the p-value is less than the 0.05 significance level, we reject the null hypothesis.

To quantify the strength of the difference between two groups of observations beyond p-values interpretation, we used the Cliff's Delta d, a non-parametric effect size. For Cliff's Delta d, the effect size is considered negligible for $d < 0.147$, small for $0.147 \leq d < 0.33$, medium for $0.33 \leq d < 0.47$, and large for $d \geq 0.47$ [GK05].

(RQ 3.2) To accurately determine how long students participated before and after GSoC, we distinguish inexperienced newcomers from the students with experience in the assigned projects, such as former GSoC students and project members. To identify former GSoC students, we counted how many GSoC editions a student participated in.

For distinguishing project members from the GSoC applicants who started contributing early, for each year, we used the GSoC's announcement date of the accepted mentoring organizations as a threshold. Thus, if a developer started contributing after the announcement (threshold), we considered the developer a newcomer. Otherwise, we treated the developer as an experienced student. For GSoC 2013, the announcement of the mentoring organizations was made 70 days before the coding period started. For GSoC 2014 and 2015, the announcement was made 84 days before kickoff.

Therefore, newcomers are students who did not have any commits older than the GSoC announcement date in relation to the start date of their first GSoC edition and are not former GSoC students. We refer to the students who did not meet these criteria as experienced students in their GSoC project.

Lastly, we correlated the collected variables using Spearman's correlation to test their predictive strength across different participation periods. We used the following variables to generate the correlation matrix: participation interval (before & after) GSoC; number of commits (all periods); number of merged commits (all periods); number of contribution days (i.e., distinct commit dates) (all periods); and code churn (all periods).

3.3 Results

We start by characterizing our study sample.

GSoC Sample Characterization

Table 3.1 summarizes the characteristics of our sample in terms of the number of participation in the program as both students and mentors; participation before and after GSoC in the assigned project, and; the total and merged commits to the projects. Note that the rows regarding 3 to 6 participations may also include editions from GSoC 2010 to 2015. It is worth mentioning that there are few students with 3+ participation, which can influence the analysis of participation (before/after) and commits (total/merged). Additionally, the students who participated in only one GSoC edition are not necessarily new to the project, and the ones with 2+ participations are not necessarily project members. We present our results by analyzing these cases.

Table 3.1: *Sample characterization*

# of participations in GSoC	**# of students who participated in GSoC as students**	**# of students who participated in GSoC as mentors**	**avg contrib interval (*days*) after GSoC (std dev)**	**avg contrib interval (*days*) before GSoC (std dev)**	**avg # of commits to the GSoC projects (std dev)**	**avg # of merged commits**
1	307	9	52.0 (135)	36.3 (117)	97.0 (136)	64.3 (92)
2	48	3	77.5 (203)	108.6 (311)	216.9 (489)	115.5 (252)
3	8	0	74.4 (157)	37.2 (99)	115.5 (153)	89.1 (160)
4	3	0	3.5 (305)	0.0 (0)	155.0 (174)	20.0 (17)
5	0	0	0.0 (NA)	0.0 (NA)	0.0 (NA)	0.0 (NA)
6	1	0	476.0 (NA)	1,603.0 (NA)	477.0 (NA)	476.0 (NA)

In Figure 3.3, one can observe that considering our sample, almost half of the students (161/367, ≈44%) had code merged after the official GSoC timeline. In addition, many students (≈19%) had code merged only during GSoC.

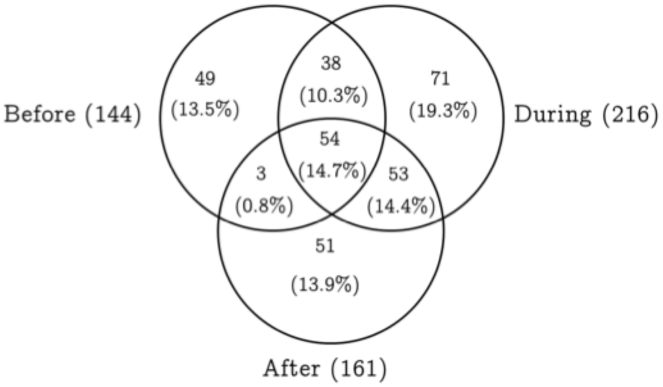
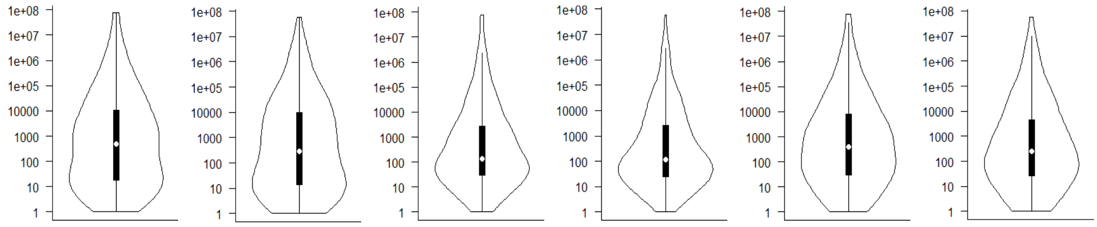


Figure 3.3: *Number of students that had commits merged to codebase before, during and after GSoC.*

RQ 3.1a. How many commits/code churn in the codebase are contributed by the students?

To estimate how many of the students' commits were merged to the GSoC projects in each participation period, we present the violin plots in Figure 3.4. For better data visualization, we removed the students without any commits for that period from the plots. We report how many students were removed and the respective percentages in brackets after the figures' captions.



(a) Commits' distribution **before** GSoC [*189/367 (~51%) did not commit*]

(b) Merged commits' distribution **before** GSoC [*223/367 (~61%) did not have any commit merged*]

(c) Commits' distribution **during** GSoC [*99/367 (~27%) did not commit*]

(d) Merged commits' distribution **during** GSoC [*151/367 (~41%) did not have any commit merged*]

(e) Commits' distribution **after** GSoC [166/367 *(~45%) did not commit*]

(f) Merged commits' distribution **after** GSoC [*206/367 (~56%) did not have any commit merged*]

Figure 3.4: *Commits and merged commits distribution by participation period (before, during, and after).*

Comparing Figure 3.4 (a) and (b), we can see that some of the students' commits were merged even before kickoff. These commits may have come from at least three distinct sources: students who were already project members; former GSoC students; and newcomers. A possible explanation for newcomers' commits is that some candidates contribute to GSoC's projects to increase their odds of being accepted. Indeed, we found students' and mentors' blogs (e.g., [Dan15]) with tips on how to be accepted.

We found support for this explanation in our data. Figure 3.5 depicts the number of distinct students who contributed to their GSoC project in the 180 days before kickoff: the choice of this threshold was based on a study of Lin et al. [LRS17]. While the experienced students' commits (Figure 3.5b) to the project remained relatively constant until the start of the bonding period (i.e., ≈30 days before kickoff), some newcomers (Figure 3.5a) started committing to the GSoC project before knowing they would be accepted, possibly attempting to show their skills to the community before the selection.

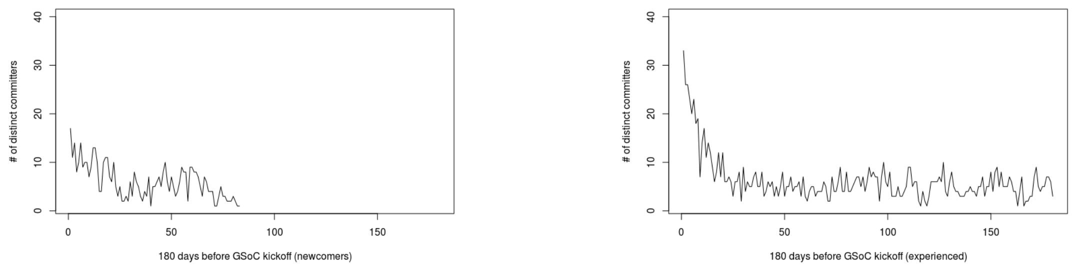


Figure 3.5: *Students' participation 180 days before and after GSoC*

Table 3.2: *Effect size and Wilcoxon Signed Rank Test comparing the # of commits of students by year*

	2013 *d*	**2014 *d***	**2015 *d***
During vs Before	0.73 (*large*) *	0.67 (*large*) *	0.66 (*large*) *
During vs After	0.62 (*large*) *	0.56 (*large*) *	0.58 (*large*) *
After vs Before	0.01 (*neglible*)	0.21 (*small*) *	0.08 (*neglible*)

* p < 0.05: significance level of the Wilcoxon Signed Rank Test. *d*: effect size computed with Cliff's Delta

In Figure 3.4 (c) and (d), we can see that most often the merges into the codebase occurred during GSoC, typically ranging from 1 (Q_1) to 2,676 (Q_3), totaling ≈76.8M merged commits. During this timeframe, the students performed 224M commits. In the worst cases (≈25%), the OSS projects had no merged commits, even though these students performed 143.2M commits. Strictly speaking, most OSS projects of our sample benefited from participation in GSoC, since in ≈87% of the cases, they had at least one merged commit to the codebase. When only newcomers are considered, ≈54% of OSS projects merged at least one commit.

In Figure 3.4 (e), we can see that ≈45% of the students did not commit anything after GSoC. The commits of the students who did, typically ranged between 31 (Q_1) and 8,064 (Q_3). In Figure 3.4 (f), we can observe that ≈44% of the students had commits merged to codebase, typically ranging from 26 (Q_1) to 4,452 (Q_3). Hence, code was merged to codebase in all periods.

Additionally, to test whether the before, during and after commits' distributions for each year were statistically equivalent, we used the Wilcoxon Signed-Rank test. In Table 3.2, we can observe that for all years, there is a significant effect size when we compare the students' commits distribution performed during GSoC to the ones made before and after the program. However, when we compared the commits made after GSoC to the ones made before GSoC, we could only find a statistical difference for GSoC 2014, still with a small effect size. For GSoC 2013 and GSoC 2015, we did not find any statistical difference.

However, when we measured the strength of this difference, we found it to be small, suggesting that, in the long term, the commits performed after tended to return to the levels before GSoC. This happens due to the commits of top contributors, which can be thousands of times higher than regular contributors. In addition, top contributors mostly consist of experienced students.

Analyzing the students' commits provides one perspective on students' contributions; we analyzed code churn (how much code was merged) to offer an additional perspective. Figure 3.6 depicts the distribution of students' code churn per participation period. The churn before boxplot shows the distribution median as 1,482, with its top ≈25% ranging between 8,880 (Q_3) and 21,964 (upper whisker). For the during period, the distribution median was approximately six times higher (≈8,900), with its top ≈25% ranging between 30,132 (Q_3) and 71,000 (upper whisker).

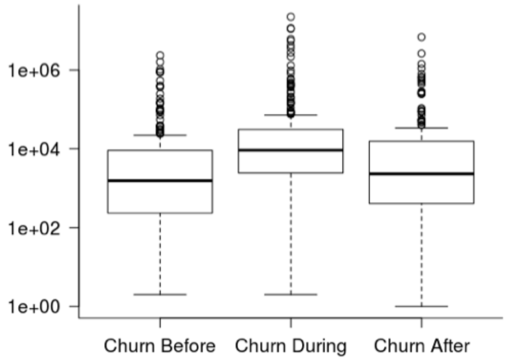
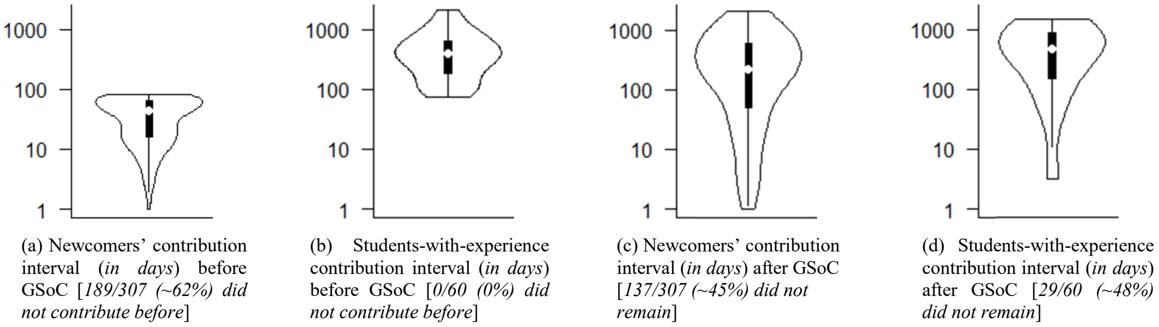


Figure 3.6: *Students' code churn by participation period.*

The churn after boxplot shows that the students' code churn significantly decreased after the program, with the distribution median decreasing to 2,435. The top 25% of the distribution remained high, ranging between 16,000 (Q_3) and 33,700 (upper whisker). We can understand the magnitude of the students' contribution when we add the code churns to the distributions. In this way, we can see that the code churn before GSoC totaled 11.5M, during, 81.9M, and after, 19.1M.

RQ 3.2a. What was the students' participation before and after GSoC?

To understand how long the students participated in their assigned GSoC projects, Figure 3.7 depicts the distribution of the students' participation intervals before and after GSoC.



(a) Newcomers' contribution interval (*in days*) before GSoC [*189/307 (~62%) did not contribute before*]

(b) Students-with-experience contribution interval (*in days*) before GSoC [*0/60 (0%) did not contribute before*]

(c) Newcomers' contribution interval (*in days*) after GSoC [*137/307 (~45%) did not remain*]

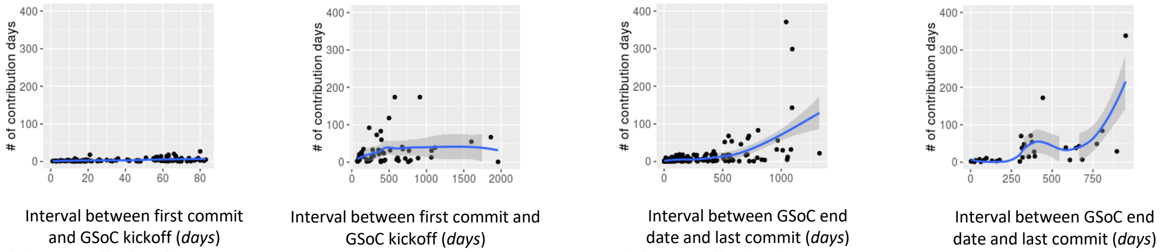
(d) Students-with-experience contribution interval (*in days*) after GSoC [*29/60 (~48%) did not remain*]

Figure 3.7: *Participation before and after (retention) distribution for newcomers and experienced students (in days).*

We split the students into newcomers and experienced. Figure 3.7 (a) and (c) show newcomers' participation (in days) before and after GSoC, while Figure 3.7 (b) and (d) show the same information for the experienced students. As previously, for better data visualization, we only show the students who kept contributing to the assigned projects. Thus, we report how many students were excluded and the respective percentage after the figures' captions, in brackets. Figure 3.8 (a) complements a previous finding, by informing that ≈23% of the newcomers who committed before kickoff did it before knowing whether they would be accepted in. However, typically, the students had not contributed to their GSoC projects before the program, which suggests that GSoC is indeed attracting potential contributors.

In Figure 3.7 (b), we can see that many experienced students have long previous participation intervals in their assigned projects (Q_1=187.2; Q_3=639). By further analyzing these cases, we found that they mostly consisted of GSoC former students (47). In Figure 3.7 (c), we can see that a significant minority of newcomers did not keep committing to their GSoC projects (≈45%). So, some OSS projects of our sample benefited from the newcomers' contributions even after the official program end.

In Figure 3.7 (d), as with newcomers, we can see that experienced students did not keep committing to the repository. The long participation interval of those who did refers mostly to participation in subsequent GSoC editions, which we consider a different, but valid, type of retention. Thus, the experienced students' participation in their assigned projects typically ranged from 114 (Q_1) to 596.5 (Q_3) days after the program. In addition, we found 13 experienced students (21.6%) who continued regularly contributing, which indicates that some participants remained tied to their GSoC project and participated in more than one edition.



(a) Distinct contribution days' count vs range between first commit to the project and GSoC kickoff for newcomers (*307 students*)

(b) Distinct contribution days' count vs range between first commit to the project and GSoC kickoff for students with-experience (*60 students*)

(c) Distinct contribution days' count vs range between GSoC end date and last commit to the project for newcomers (*307 students*)

(d) Distinct contribution days' count vs range between GSoC end date and last commit to project for students-with-experience (*60 students*)

Figure 3.8: *Contribution days' count (# of days) before and after GSoC vs the interval (in days) between: first commit and GSoC start date for newcomers (a) and experienced students (b); and GSoC end date and last commit for newcomers (c) and experienced students (d).*

The analysis of participation intervals before and after GSoC only shows one facet of contributions out of GSoC's timeframe, as it does not inform anything about the frequency of contributions. We used the number of contribution days as a measure of frequency. Figure 3.8 presents a relationship between participation intervals (PI) and the number of contribution days (CD) in scatter plots.

In Figure 3.8 (a), we can observe that although many newcomers started participation after Google's announcement of accepted mentoring organizations, CDs are mostly less than 10. Only 14 (6%) of newcomers contributed more than 10 distinct days. In Figure 3.8 (b), we can observe that the experienced students' PIs are considerably higher than the newcomers' (who are limited to an 84-day limit of previous participation by our definition). However, we can see that most students have less than 50 CDs, with the distribution median being 16 CDs before GSoC. In Figure 3.8 (c), we can see that, after GSoC, newcomers' PI increased considerably, reaching in many cases to PIs higher than 500 days. However, except for a few cases, CDs did not increase proportionally. For instance, we observed a median of 5 CDs for the newcomers who contributed longer than a month, 9 CDs for the ones who contributed longer than six months, and 14 CDs for the ones who contributed longer than a year. In Figure 3.8 (d), we can see that, after GSoC, some of the experienced students

Table 3.3: *Correlations on newcomers and experienced students' participation.*

Newcomers' spearman correlations													
	Before GSoC				**During GSoC**				**After GSoC**				
	B	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**	**N**
A contrib interval before A	.926	.796	.955	.860	.293	.152	.203	.166	.113	.058	.087	.050	.023
.112	**B # of commits before B**	.852	.967	.922	.473	.334	.364	.335	.022	.221	.058	.191	.122
.004	.469	**C merged commits before C**	.817	.806	.360	.277	.301	.414	.009	.165	.022	.161	.159
.315	.679	.325	**D # of contrib days before D**	.911	.389	.246	.291	.243	.057	.135	.021	.117	.040
.120	.923	.499	.677	**E code churn before E**	.459	.318	.389	.319	.025	.216	.067	.204	.119
.076	**.940**	.381	.481	**.883**	**F # of commits during F**	.905	.883	.799	.189	.561	.321	.509	.373
.026	**.825**	.341	.409	**.753**	.897	**G # contrib days during G**	.830	.787	.053	.403	.160	.352	.268
.075	**.911**	.376	.481	**.886**	.974	.873	**H code churn during H**	.753	.121	.465	.247	.476	.295
.137	.598	**.714**	.313	.622	.653	.660	.660	**I merged commits dur I**	.139	.427	.240	.376	.457
.019	**.773**	.342	.320	**.726**	**.818**	.641	**.787**	.535	**J contrib interval after J**	.829	.920	.774	.735
.025	**.877**	.359	.441	**.837**	**.913**	**.750**	**.881**	.538	.920	**K # of commits after K**	.908	.914	.797
.027	**.815**	.299	.371	**.766**	**.853**	.673	**.806**	.497	.957	.943	**L contrib days after L**	.835	.778
.055	**.884**	.384	.473	**.857**	**.912**	**.729**	**.915**	.555	.894	.959	.890	**M code churn after M**	.750
.102	**.702**	.537	.369	**.712**	**.733**	.608	**.743**	.710	.836	.802	.811	.797	**N merged commits # after**
Before GSoC					**During GSoC**				**After GSoC**				
Students-with-experience's spearman correlations													

had CDs comparable to the newcomers who contributed before knowing they would be accepted. In addition, we observed a median of 22 CDs for the experienced students who contributed more than a month, 38.5 for the ones who contributed longer than six months, and 41 for the ones who contributed longer than a year.

RQ 3.2b. Is previous participation associated with students' retention?

Many works in the literature have correlated developers' participation on OSS projects with numerous variables, trying to predict early-on the developers who would continue contributing to the OSS project (e.g., [CF09, DJ15, SLW12]). We correlated the data we collected on the students to study the variables' predictive strength, especially in different participating periods, as shown in Table 3.3. The variables are presented in the main diagonal of the correlation matrix, preceded and followed by a letter A-N. The entries in the upper triangular refer to the newcomers' correlations, while the entries in the lower triangular refer to the experienced students' correlations. Participation periods are highlighted in boxes in the lower and upper triangular of the matrix.

In Table 3.3, for newcomers, we can see that the correlations did not show any predictive strength, as variables are mostly weakly correlated (<0.5). For experienced students, we can see that: the number of commits before (B) and code churn before (E) are strongly correlated (>0.7) with how much code is written during (variables F and H) and after (variables K, M, and N) GSoC. In addition, the variables B and E are strongly correlated with how long students stay after the end of GSoC (variables G, J, and L). Similarly, the amount of code students wrote during the program showed to be good predictors regarding how much code students wrote and how long they stayed after GSoC. Thus, our findings complement the results of Schilling and colleagues [SLW12] that prior development experience in the project is associated with higher levels of retention.

3.4 Discussion

One possible implication of our results is that when OSS projects select newcomers for participating in GSoC, instead of experienced students, they need to be prepared to invest in the newcomers' mentoring, without expectations of long-term commitment, as it can be seen in Figure 3.7. Not surprisingly, the period with higher participation was during GSoC (sponsored period), as depicted in Figure 3.6. Our results showed that ≈64% of the students did not stay later than a month after the program. Based on this finding, we suggest that communities should come up with a strategy for handling the disappearing students, which could be as simple as maintaining contact through email. Future research could investigate alternative ways to prevent students from disappearing.

Our results also suggest that Summers of Code provide OSS projects with applicants' participation before kickoff, possibly due to the competitive nature of the programs. OSS projects should offer a pre-Summer of Code to engage applicants. The project should take advantage of applicants' participation before the program, offering a formal opportunity for applicants to show their skills and interact with the community. As a result, the project should receive more contributions and have the opportunity to showcase the community. This organization scheme could potentially mitigate the mentors' selection and ranking load, as they would have more data on the applicants. This strategy could also work for the BioPython OSS project, which experienced similar problems, as reported by Trainer and colleagues [TCH14].

OSS projects and Summer of Code administrators should also offer opportunities for those who were not selected to receive stipends to participate voluntarily. In this case, the participants would be awarded participation certificates. Thus, even non-sponsored participants would have the chance to acquired knowledge, experience, and branding.

There is another interesting facet related to retention that we would like to highlight. Our results suggest that finding top contributors, though rare, could yield substantial dividends for the community, considering the number of (merged) commits. The findings above—higher visibility, contributing as a strategy to increase acceptance odds, merged code during the program, and finding top contributors—may explain why the number of OSS projects interested in entering GSoC has increased throughout the years.

Summers of Code seem to be a channel of contribution to OSS projects that not only have mitigated barriers for the students who wanted to become volunteer contributors (see Steinmacher et al. [SCGR15] for an overview of the barriers that newcomers usually face), but also have taken advantage of who would never contribute otherwise.

3.5 Limitations and Threats to Validity

This study has several limitations, which we here both acknowledge and report how we aimed to mitigate them. First, our sample may not be representative of the whole GSoC students' population, despite our efforts to collect a representative random sample. This means that it may be possible to reach other conclusions with a different set of students.

A major threat is the misidentification of the students and their projects in the SCMs, and the students' IDs in the local database. For instance, in some cases, the student IDs, both in the SCMs and in the local database, were composed of the students' name initials (or combinations). Although we excluded the cases that we were uncertain about, it is still possible that we incurred some misidentification. In some cases, the same student used multiple IDs to perform the commits. In this case, the threats are that we could have: incorrectly grouped IDs from different students, not identified all the IDs used by a student, and identified the IDs used in a different GSoC edition than the one under consideration. Even though we carefully inspected every student in our sample, it is still possible that these threats weakened our results.

Moreover, we used the students' and projects' names as matching criteria to determine whether the students participated as mentors in other editions. In the case of students who share common names working for the same project, we might have wrongly counted them as the same student. We mitigated this threat by carefully inspecting if the year of participation as a student was before the participation as a mentor for the same project. As we did not personally contact any mentor, it may be the case that students delivered their final code after the official GSoC's end date, which by our method would be wrongly counted as retention.

Finally, our conclusions may be biased toward the number of merged commits. We do not control many potentially important variables, such as programming language, code complexity, or how vital the merged commits were to the OSS projects. It may be the case that the students who had only one merged commit contributed more—in terms of aggregated value than those with numerically more merged commits.

3.6 Final Considerations

There is evidence that OSS projects expect that Summers of Code may be an effective channel not only for the attraction and retention of newcomers but also for the code contributions made during participation [CIA13, TCKH14, TKCH16, TCH14]. In this chapter, we investigated GSoC, providing empirical evidence on different aspects of the students' participation, such as how much the program fostered contributions (i.e., commits, merged commits, and code churn) and how long did students contribute to the assigned projects before and after the program.

For analyzing RQ 3.1, for each period, we counted how many of the students' commits were merged in the codebase. We estimated how much code the students added by calculating the code churn (i.e., lines added + lines removed) for each commit. We found that code merges occurred before, during, and after GSoC, including for newcomers. Most merged commits occurred during GSoC, although many OSS projects merged in other periods. We also could obtain the magnitude of students' code contributions by analyzing code churn' medians: ≈11.5M (before); ≈82M (during); and ≈19M (after).

For analyzing RQ 3.2, we started by differentiating newcomers from experienced students. Then, we investigated participation interval, contributions, and the contribution days' count (i.e., distinct commit dates). We found that ≈23% of newcomers started participating in GSoC projects before knowing they would be accepted. After GSoC, participation decreased from ≈43% newcomers who kept contributing longer than a month to ≈16% of them who kept contributing longer than a year. Experienced students started contributing more than a year earlier than kickoff, while a year later, ≈23% of them were still contributing. Regardless of the experience in the GSoC project, the number of contributions was not proportional to longer participation intervals.

We conclude this study highlighting that OSS projects that need to achieve project task accomplishment should consider prioritizing experienced students, as they are already familiar with the projects' contribution norms, and they possibly have a lower learning curve. For the students who kept contributing after GSoC, contributions tended to diminish slowly, which can signal to OSS projects that they should use their strategy for handling these students. This can be as simple as sending an email explaining the importance of the students' participation. In addition, OSS projects can establish a recommended period before Summers of Code for applicants start contributing and interacting with the community. Thus, applicants who start earlier and contribute more would have more acceptance chances.

Chapter 4

Empirical Identification of Students' Motivations

Quantitative evaluations of students' contributions are not satisfactory for understanding students' motivations. We claim that understanding students' motivation to engage in an OSS project via Summer of Code programs is a complex phenomenon, which requires triangulation through different data sources. As Turner indicated concerning research on students' motivation in general: "*The focus ... should move beyond students alone*" [Tur01, p. 91]. In this Chapter, we present a study that aimed to identify and understand what motivated students to engage in OSS via Google Summer of Code (GSoC), and, in some cases, remain in the projects as contributors.

We highlight that the content of this chapter was accepted for publication at JSS 2018 [SWG+19]. We made minor modifications to the published manuscript for cohesion and readability.

4.1 Research Questions

We established the following research questions (RQ) to guide our answers.

RQ 4.1: According to students, what motivates them to participate in Summer of Code programs?

RQ 4.2: According to mentors, what motivates students to participate in Summer of Code programs?

4.2 Research Method

To answer our RQs, we surveyed students and mentors and interviewed students. Figure 4.1 outlines the research method we followed in this study.

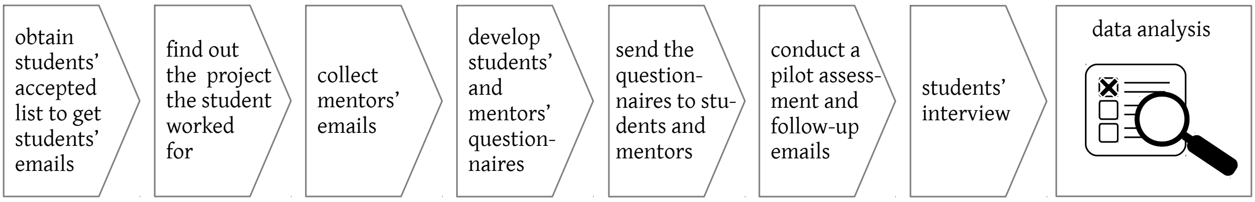

obtain students' accepted list to get students' emails
find out the project the student worked for
collect mentors' emails
develop students' and mentors' question-naires
send the question-naires to stu-dents and mentors
conduct a pilot assess-ment and follow-up emails
students' interview
data analysis

Figure 4.1: *Research method*

4.2.1 Students' and mentors' survey

We investigated students' motivation to join OSS projects through Google's Summer of Code program from the perspectives of both students and mentors.

We conducted a survey not only to assess the motivational factors we found in the current literature but also to uncover potential new ones. This survey enabled us to contact a large number of students and mentors in a structured manner, potentially increasing the odds of reaching saturation concerning the motivational factors.

In the following, we detail: how we contacted students and mentors, how we designed and administered the questionnaires, and how we analyzed the collected data.

Contact information collection

The first step of our study was to search for information (e.g., email address) that would allow us to contact the students. We used the accepted students' list, published by Google, which only contains the students' and the OSS projects' names.

Next, we investigated which specific project a student worked for, considering all the OSS projects under the community the student was selected. For example, although Google informs that the Apache Software Foundation (community) accepted participant John Doe, we still do not know for which Apache project John worked. We considered that we found their emails when we had clear evidence linking the student with their corresponding project name. For instance, when we found students' web blog or their professional resumés describing their experience in the program, or when we found their messages about the program in projects' discussion lists.

As the collection and verification of each student project is a laborious and time-consuming task, we limited our analysis to the GSoC 2010-2015 editions, in which approximately 7,000 students participated.[1] By the end of this step, we could gather 1,000 students' and 730 mentors' emails.

Questionnaire design and administration

We used questionnaires as a data collection method, following Fink's advice on how to design surveys [Fin95]. We asked students[2] about their contributions to OSS before and after GSoC (ques-

[1] http://developers.google.com/open-source/gsoc/resources/stats

[2] The students' questionnaire is available at http://docs.google.com/forms/students

tions 1-5) and general questions about their participation in GSoC (questions 6-13). We also asked them questions that further explored the relationship between stipends and participation in GSoC (questions 14-15). Additionally, we asked students whether they would enter a hypothetical-GSoC that offered all motivational factors but one (question 16), which allowed us to rank and examine how essential these factors were. We concluded by asking them about demographic information at the time of their first participation (questions 17-22).

We designed the mentors' questionnaire[3] using the same structure as the students', with the difference that mentors had to answer about their students in general. It is worth emphasizing that we are aware that the mentors' answers may not refer to the students in our sample. We highlight that the primary focus of this research is on identifying motivation factors for the students' population, which does not require a specific set of mentors. Nevertheless, any disparities found between students' and mentors' answers should be interpreted carefully, as we point out in the limitations Section.

We conducted a pilot assessment of the questionnaire with 2 GSoC 2015 students. After minor adjustments, we sent out emails inviting students to participate in this research. We also employed principles for increasing survey participation [SLMH13], such as sending personalized invitations, allowing participants to remain completely anonymous. Besides, we combined closed and open questions. As an attempt to increase the number of responses, we also sent out two follow up emails as a reminder. The follow-up increased our responses by two students. We point out that all respondents explicitly consented to participate anonymously in this research.

We sent out 1,000 survey invitations (≈14% of the total GSoC students for the investigated period) to students and received answers from 141 students (14.1% response rate). We also sent out 730 survey invitations to mentors, and we received 53 responses (7.3% response rate). The number of survey invitations sent out to mentors is smaller than that of the students because a considerable number of mentors participate in more than one year.

Analysis of survey responses

We used multiple procedures for analyzing the responses of students and mentors. We employed descriptive statistics for analyzing the answers to the closed-ended questions. We analyzed responses to the open-ended questions in two stages, namely open coding and axial coding [SC98]. During open coding, codes are identified, and their properties are discovered in the data. During axial coding, we group semantically similar codes into concepts. Next, connections between the concepts are identified and aggregated according to their properties to represent categories [Cha06, Cre12].

For convenience, only the author of this thesis accomplished the open coding in the first stage, which resulted in 481 different codes. Next, two other researchers collaborated to derive the 17 concepts shown in the white lines in Table 4.2. In the second stage, another researcher reviewed the concepts and collaborated in the generation of the seven categories presented in the gray lines in Table 4.2. To achieve the current form, several adjustments were made to the concepts and categories as new authors revised the work.

[3] The mentors' questionnaire can be accessed at http://docs.google.com/forms/mentors

In our results, we provide a selection of representative quotes from students and mentors, denoted respectively by $S_{\#}$, and $M_{\#}$, with their IDs in subscript. For each concept that emerged in the qualitative analysis, we indicate how many participants mentioned the theme in parentheses. The numbers only show how much evidence the data analysis yielded for each theme; they do not necessarily mean the importance of a theme.

4.2.2 Semi-structured interviews

We interviewed the surveyed students who volunteered for follow-up online interviews to enlighten some motivation factors that were still unclear, such as the role played by the stipends. Besides, we wanted to get their perception of the coding scheme we derived during the survey analysis.

We crafted the interview questions following Merriam's [Mer09] advice to stimulate responses from the interviewees. We asked them questions about their experiences and behaviors concerning GSoC, their opinions, their factual knowledge, and some background/demographic questions.

We sent out 43 invitation emails and received 11 positive responses. We lost contact with one student, and we performed ten semi-structured interviews. The interviews lasted, on average, 23 minutes. At the end of the interviews, we presented and explained our coding scheme derived from the survey analysis, and asked for changes or insights that the students might have. Two interviewees suggested minor changes, such as including buying hardware equipment for participation as one of the roles of the stipends.

4.2.3 Sample characterization

In this Section, we first characterize the students and mentors of the sample we used for the qualitative analysis.

Demographic information about students

Our sample comprises 112 male students, two females, and two self-identified as other. The predominant age for the first participation in GSoC was between 21-25 years old (63), followed by 18-20 years old (45). A minority of students were between 26-30 years old (26) and 31-40 years old (7). No student in our sample was over 40 years old at the time of first participation.

Regarding education, the respondents were mostly university undergraduates (58) and graduates (41), with a significant minority of high school graduates (24). A smaller number of students had some postgraduate work (7) or a postgraduate degree (6). Students who had not completed high school (4) and students with trade/technical/vocational training (1) rarely appeared in our sample.

Besides, our sample is composed of students with previous development experience ranging from 2-4 years (62), and 5-9 years (41). Less numerously, some students indicated their development experience as being less than two years (18), or as being more than ten years (20). In comparison,

Table 4.1: *Students' count per country of residence at the time of first participation*

Country of residence	Count of countries	Count of students per country	% of students per country
India	1	33	23.4
USA	1	21	14.9
Brazil	1	8	5.7
Russia	1	7	5.0
Spain	1	6	4.3
Canada, France, Poland	3	5	3.5
Romania, Sri Lanka	2	4	2.8
Argentina, Germany, Ukraine	3	3	2.1
Austria, Hungary, Portugal, United Kingdom	4	2	1.4
Australia, Belarus, Bosnia, China, Croatia, Czech Republic, Denmank, Egypt, Finland, Greece, Italy, Netherlands, New Zealand, Peru, South Africa, Sweden	17	1	0.7
Did not answer	-	10	7.1
Total	34	141	100.0

GSoC published statistics on students' demographics for GSoC 2014[4] (we could not find other years' detailed statistics). For that year, 10% of the students were females, ≈68% of them were undergraduates, and they were typically between 18-25 years old. Our sample resembles these characteristics.

We also analyzed the students' distribution per country, shown in Table 4.1. We received answers from participants from 34 countries. Approximately 23% of the students resided in India and ≈15% of them in the USA. In comparison with GSoC published statistics from 2013,[5] 2014,[6] and 2015,[7] the sample is likely representative regarding country.

Demographic information about mentors

All respondent mentors were males (53). Half were between 31-40 (27) years old, around one third were more than 40 (15), around one fifth were between 26-30 (10), and only one was between 21-25 (1) years old.

In addition, typically, the respondents participated (as mentors) in more than one edition: 1 edition (10); 2 editions (15); 3 editions (13); 5 editions (11); 6 editions (2); 7 editions (1); and 11 editions (1). On average, the mentors mentored 4.5 students, and the median of this distribution is 3. Most mentors had more than ten years (44) of development experience, with a few that had seven years (5), six years (2), five years (1), and eight years (1).

[4] https://opensource.googleblog.com/2014/06/gsoc-2014-by-numbers.html

[5] https://opensource.googleblog.com/2013/06/gsoc-2013-full-of.html

[6] https://opensource.googleblog.com/2014/05/gsoc-2014-by-numbers.html

[7] https://opensource.googleblog.com/2015/05/gsoc-2015-stats-about.html

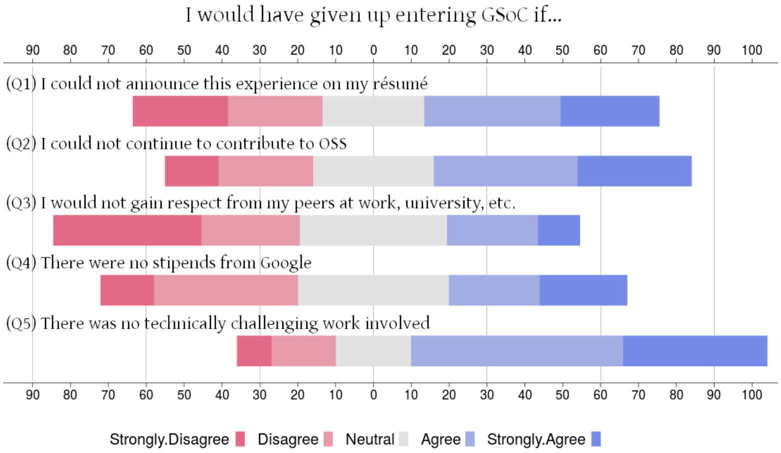


Figure 4.2: *Students' assessment of motivation factors for participating in GSoC*

4.3 Findings

In this section, we present and discuss our findings.

4.3.1 Students' motivations to engage in OSS projects via GSoC

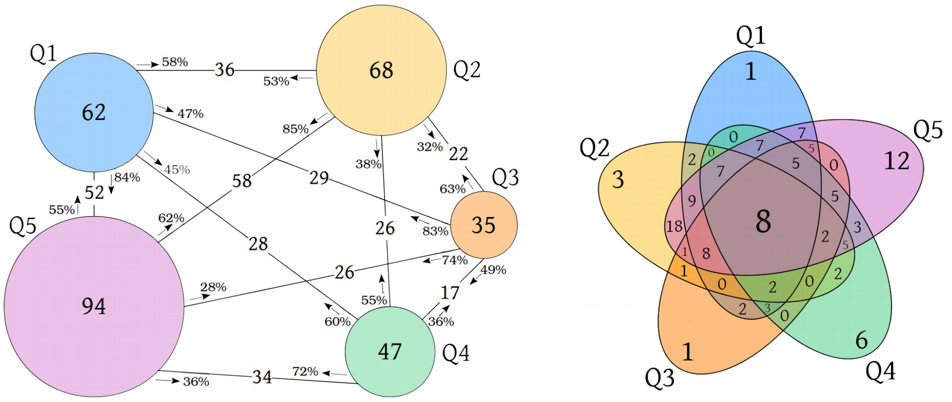
In this Section, we report our findings on what motivated the students in our sample to join GSoC, from their perspective (RQ4.1). Based on the literature (e.g., [BBH^{+}08]), we hypothesized that students would engage in OSS via GSoC for 5 significant motivation factors: career building (Q1); as an entry gateway to OSS projects (Q2); peer recognition (Q3); stipends (Q4); and intellectual stimulation, such as a technical challenge (Q5).

Figure 4.2 depicts in stacked bars the agreement level (5-level Likert items) on how essential the investigated motivation factors were for the students to participate in a hypothetical-GSoC that offered all factors but one. We considered a motivation factor essential when the students reported they would give up entering the hypothetical-GSoC without that factor.

Figure 4.3 (a) offers an alternative perspective, with the students' responses presented in a graph, highlighting counts, proportions, and how the motivations factors relate to each other. Each node in Figure 4.3 (a) indicates the number of students who considered that factor essential. Node sizes are proportional to the students' response counts. The edges depict the counts in the intersection of two motivation factors. Percentages show the proportion that the intersection values represent in relation to a node (i.e., motivation factor).

Figure 4.3 (b) decomposes the students' response counts into sets and subsets, with the results shown in a Venn diagram. We discuss Figure 4.2 and Figure 4.3 in greater detail in the following Sections.

The analysis of students' textual answers yielded motivation factors other than the ones that triggered our investigation, such as *learning* and *academic* concerns. Table 4.2 presents all the concepts and categories derived from the students' answers (survey and interviews). Gray lines refer



(a) Nodes represent the number of students who agreed (or strongly) that a motivation was essential. Edges represent the number of students who agreed (or strongly) for two motivations.

(b) Venn diagram representation of the surveyed students who agreed (or strongly) that a motivation was essential for participation.

Figure 4.3: *Surveyed students' motivation count in a graph (a) and in a Venn diagram (b). Career building (Q1); contribute to OSS (Q2); peer recognition (Q3); stipends (Q4); technical challenge (Q5)*

to categories formed by the grouping of the concepts shown in white lines. Also, Table 4.2 shows the number of students and mentors who mentioned a particular concept. The number in parentheses denotes the percentage of mentions concerning the total of students or mentors. Furthermore, students often mentioned a motivation factor broadly, in which case we present these counts in the same line (gray) as the categories. For each category, we sum all the counts (concepts and categories) and present them in a line named total.

For readability concerns, we adopt the following convention to present the results in Table 4.2. Concepts are presented in small capitals: CONCEPT (1). Categories are presented in italics, such as *category* (1). Totals are presented in bold, such as **total** (1). In all cases, the numbers in parentheses depict the counts. It is worth noting that all students that we talked to during the follow-up interviews validated the concepts and categories presented in Table 4.2. As S_9 representatively said at the end of the interview: "*Yeah, yeah, I mean, I can see myself interested in many of these points [the categories] right, I did it [GSoC] for most of them, I'd say.*"

We start presenting our findings for the motivations that triggered our investigation, followed by the motivations we coded from the students' textual answers and interviews.

Career building

Approximately 44% of the students considered adding the GSoC experience to CVs essential (see Q1 in Figure 4.2 and Figure 4.3), preferring not to participate otherwise. Aside from *technical challenge*, *career building* was the motivation factor students were the least divided about, with ≈20% of them being neutral on whether it was essential. Figure 4.3 (a) depicts that the students motivated by *career building* were also mostly motivated by *technical challenge* (84%) followed by *contribution to OSS* (58%). Figure 4.3 (b) reveals that only one student was purely motivated by *career building*.

We also analyzed students' textual answers to obtain additional information, which resulted in

Table 4.2: *What motivates students to participate in Google Summer of Code?*

Categories (gray) and codes (white)	**# of students (%)**		**# of mentors (%)**		**GSoC related goals**
Stipends (generic mentions)	34	(24)	21	(40)	-
Compensation for a provided service	10	(7)	0	(0)	
Source of funding (living expenses, hardware, dedication)	13	(9)	2	(4)	
Payment of studies or tuition	13	(9)	0	(0)	
Project members	12	(9)	0	(0)	
Currency conversion	2	(1)	1	(2)	
Total	84	(60)	24	(45)	
Contribution to OSS (generic mentions)	27	(19)	2	(4)	(*ii*, *iii*)
Interaction with mentor or other members	21	(15)	5	(9)	
OSS philosophy and culture	16	(6)	0	(0)	
GSoC lowers entry barriers	9	(6)	0	(0)	
OSS/GSoC project itself	8	(15)	1	(2)	
Total	81	(57)	8	(15)	
Learning (generic mentions)	5	(4)	4	(8)	(*v*)
Real-world development experience	51	(36)	13	(25)	
Improvement of skills other than development	2	(1)	0	(0)	
Total	58	(41)	17	(32)	
Career building (generic mentions)	7	(5)	0	(0)	-
GSoC looks good on CV	31	(22)	9	(17)	
Total	38	(27)	9	(17)	
Academic (generic mentions)	7	(5)	1	(2)	(*iv*)
Course credit	2	(1)	1	(2)	
Internships or summer projects	15	(11)	4	(8)	
Research purposes	4	(3)	2	(4)	
Total	24	(17)	6	(11)	
Peer recognition (generic mentions)	0	(0)	1	(2)	-
Prestige or bragging rights	9	(6)	1	(2)	
Total	9	(6)	2	(4)	
Intellectual stimulation (generic mentions)	0	(0)	0	(0)	(*i*)
Technically challenging work	5	(4)	2	(4)	
Total	5	(4)	2	(4)	

the concepts and categories shown in Table 4.2 (see *career building*). The analysis revealed, though not exclusively that the students who mentioned the career as a motive for participation (27%) mostly entered the program because GSoC would look good on their CVs (31) possibly as an attempt to make their CVs more attractive to future employers. Examples include S$_{79}$: "*(...) adding the 'Google' keyword on a resume was a good plus,*" and; S$_{106}$: "*I needed some real experience to my CV (...).*"

While a few other students considered *career building* (7) to be among their primary motivation, their mentions were only vague. The explanation of S$_{39}$ offers an example: "*I participated [in GSoC] because it was a great opportunity for my career.*" As mentioned, we present these counts in the same line as the category.

Moreover, **career building** (38) was a concern for several interviewees who declared they would not have given it up (7), revealing that their careers would still benefit from the: real-world development experience (3); interacting with OSS project members (2); and prestige (2) of the program. No student mentioned using GSoC to get new career opportunities in a different job at the time of first participation.

Contribution to OSS

The students who explicitly stated to have entered GSoC motivated by contributing to OSS were grouped into the **contribution to OSS** (81) category, even though in many cases, they pinpointed different components of the contribution as their primary motivating factor. Table 4.2 shows the concepts extracted for this category (see *contribution to OSS*).

Some students mentioned being driven by the GSoC/OSS project itself (8), such as S$_{136}$: "*I wanted to add a feature to an open source media player, and I felt like GSoC would motivate me to implement this feature in a short amount of time,*" and; S$_{85}$: "*I was interested in contributing to Free/Open source libraries and trying something new.*" It is worth noting that these students did not necessarily mention they were interested in becoming frequent contributors.

We found cases of students who entered GSoC motivated by the OSS culture and philosophy (16), such as S$_{73}$ who said: "*I'm passionate about FOSS and all philosophy around it;*" S$_{58}$: "*I was always attracted to the idea of contributing code for good,*" and; S$_{11}$: "*I love coding and the idea of contributions to open source and your code helping others is too good.*"

Several OSS projects are known to have high entry barriers for newcomers [SWC+14], and in some cases, students considered that GSoC lowers entry barriers (9) into OSS, such as S$_{135}$: "*I wanted to get involved developing OSS but found there to be a high barrier to entry for existing projects (...) The goal for me was primarily to help break into the OSS community, which felt difficult to penetrate at the time.*"

More often, students considered GSoC an opportunity to interact with OSS mentor or other community members (21), such as S$_{48}$, who said: "*It was a chance to interact with an OSS community.*"

Students were not in general contributors to the assigned GSoC projects before kickoff (see Table 4.3), although a significant minority (44%) had already contributed. Besides, most of the students reported having some previous experience in contributing to OSS projects (see Table 4.4).

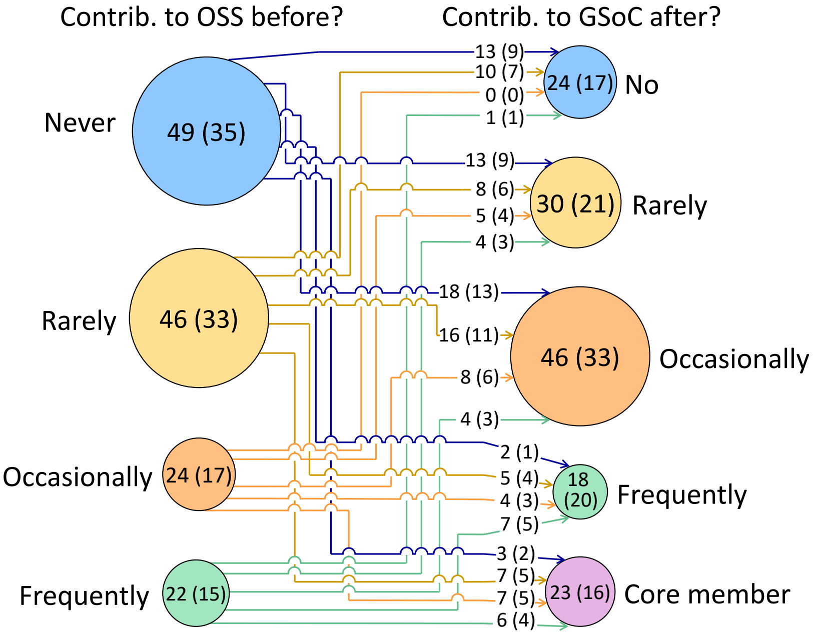


Figure 4.4: *Contribution frequency to OSS (before GSoC) and to GSoC projects (after GSoC). Students' count (%).*

Furthermore, Figure 4.4 illustrates the relationship between the self-reported contribution frequency to OSS projects before kickoff and the assigned GSoC projects after the program. We can observe that 75 students (≈53%) reported an increase in the contribution frequencies after GSoC. The 29 students (≈21%) who before GSoC had occasionally (at most) contributed to OSS projects remained as such after the program concerning contributions to the GSoC projects. Also, the 13 students (≈9%) who self-reported to be frequent contributors to OSS projects before remained as such after the program concerning contributions to GSoC projects. Nevertheless, 24 students (≈17%) lowered their contributions to GSoC projects compared to how frequently they contributed to OSS projects before GSoC.

Virtually half of the students (≈48%) considered it essential for participation to contribute to GSoC projects after the program (see Q2 in Figure 4.2), which is slightly lower than the students who entered GSoC intending to keep contributing (*'Yes'* and *'Definitely'*, which totals ≈57%) (see Table 4.5).

In addition, we can observe in Figure 4.3 (a) that the students motivated by *contribution to OSS* were largely composed of students also motivated by *technical challenge* (≈85%) followed by students driven by *career building* (53%). We found a few cases (3) of students who were exclusively motivated by *contribution to OSS*. Most commonly, students mentioned their interest in the *contribution to OSS* (27) in a broad sense, without providing any further detail.

Surprisingly, we found that there were students (2) that, due to the competitive selection process, engaged in OSS projects to increase their odds of participating in GSoC. As evidenced by S_3: "*I knew I had to do GSoC for which I started contributing to FOSS.*" This confirms what we found in

Table 4.3: *Before GSoC, did you contribute to the project you've chosen for the program?*

Responses	**Count (%)**
Never	79 (56.0)
Rarely	19 (13.5)
Occasionally	10 (7.1)
Frequently	14 (9.9)
My project started in GSoC	13 (9.2)
Core member	6 (4.3)

Table 4.4: *Before GSoC, did you contribute to OSS projects other than your own?*

Responses	**Count (%)**
Never	49 (34.7)
Rarely	46 (32.6)
Occasionally	24 (17.0)
Frequently	22 (15.6)

students' and mentors' web blogs[8] with tips on how to be accepted, suggesting that the candidates get involved with the community to increase their chances. We also found evidence of the previous *contribution to OSS* projects as an acceptance condition into GSoC, as exemplified by the Octave community: "*Previous contributions to Octave are a condition for acceptance. In this way, we hope to select students who are familiar with the codebase and able to start their project quickly.*"[9]

Some students (2) tactically chose the OSS project they would work. During the interview, S_2 explained: "*The organization [community, in this research] [that I was interested in] had some [OSS] projects, several projects, and then I think I chose the one that no other student had [yet applied for]. So I thought that my chances [could] be fair in some project nobody else was interested in.*"

Peer recognition

A quarter of the students (≈25%) considered *peer recognition* essential for participation (see Q3 in Figure 4.2 and Figure 4.3), being the least essential, when compared to the other motivation factors. Similarly, a minority of students mentioned **peer recognition** (9) motives (see Table 4.2, *peer recognition*). Often, students referred to *peer recognition* concerning PRESTIGE (5) of the program among their peers of yet BRAGGING RIGHTS (4).

Stipends

Around 30% of the students considered *stipends* essential for participating in GSoC, even though this motivation factor had the largest number of neutral students (see Figure 4.2 and Figure 4.3). In most cases, students did not include any information that allowed us to understand how the *stipends* (34) were used. S_{100} offers an example, who succinctly described his primary interest: "*The money.*"

[8] https://danielpocock.com/getting-selected-for-google-summer-of-code-2016

[9] https://wiki.octave.org/GSoC_2018_application

Table 4.5: *Before GSoC, did you intend to continue contributing to the project (excluding for other GSoC editions)?*

Responses	**Count (%)**
Not at all	8 (5.7)
No	11 (7.8)
Maybe	42 (29.8)
Yes	40 (28.4)
Definitely yes	40 (28.4)

Table 4.6: *Have you actually continued contributing (excluding for other GSoC editions)?*

Responses	**Count (%)**
No	24 (17.0)
Rarely	30 (21.3)
Occasionally	46 (32.6)
Frequently	18 (12.8)
Core member	23 (16.3)

On the other hand, in several cases, students revealed the roles the *stipends* played. In several cases, students used the *stipends* for the PAYMENT OF THEIR TUITION (13). Some PROJECT MEMBERS (12) decided to contribute to OSS projects through GSoC to benefit from the stipends, such as the case of S$_2$: "*I already got that project[,] but then I got to know it could have been sponsored by Google. So, I got [interested] in the program mostly because it was providing a lot of money.*"

Often, the *stipends* were used as a SOURCE OF FUNDING (13). We used this concept when the *stipends* were used for LIVING EXPENSES (10), as a means to make students' participation feasible, such as explained by S$_{115}$: "*As a student[,] I need to earn money for existence,*" and S$_{125}$: "*I needed the stipend for living expenses.*"

During the interviews, we found that students used the *stipends* to BUY HARDWARE EQUIPMENT (1), which we coded as SOURCE OF FUNDING (13). As S$_{47}$ said: "*I used that [the stipends] to purchase hardware equipment so [that] I could improve my development environment.*"

Furthermore, we considered SOURCE OF FUNDING (13) when existing project members could DEDICATE TIME AND EFFORTS TO THEIR PROJECTS (2), such as S$_6$: "*I was already contributing to the OSS project before the GSoC although that was in my free time. GSoC was a chance to really spend time for the project,*" and; S$_{111}$: "*GSoC was a chance for us to have a core member work on the project full time instead of just in the spare time and this helped to get lots of development and some crucial refactoring done.*"

Alternatively, some other students viewed stipends as compensation for either the service provided or the time spent, which we labeled as STIPENDS AS COMPENSATION (10), such as explained by S$_{40}$: "*I would prefer to get paid for my time. Otherwise[, I would have] contributed to open source without GSoC,*" and; S$_{86}$: "*I like to be paid for my work.*"

No student in our sample explicitly declared to enter GSoC for profit. There are many responses which mentioned the stipends to be huge, such as S$_{84}$, who commented: "*I was a really cool opportunity to (...) get a (huge) amount of money (...).*" Since the stipends' role was not explicitly stated, we present these counts in the same line as the category. This rationale also was applied to students who were motivated by CURRENCY CONVERSION (2) rates, such as S$_{137}$, who said: "*For the financial incentive (which is quite a big amount in my country) and for the opportunity to contribute to OSS projects.*" These students resided at Sri Lanka and Belarus when they participated in GSoC.

A stipends-for-profit concept would be used to group the students who wanted to make more money than what they needed for their funding. However, we could find a Q&A website[10] where allegedly some students declared they used the stipends to "*buy gifts for their families;*" "*pay for tourism trips;*" "*save money for the future;*" "*give loans to friends,*" and other uses. Thus, it is possible that some of the students who mentioned the motivation factor broadly could be labeled with a more specific stipends-for-profit concept in case we further investigated their motivation.

Stipend-motivated participation incited different sentiments in the students. Although most students' responses were neutral (120) towards the stipends, we could identify some responses with a positive tone (8), typically linking the payments to the heart of the program. As S$_{95}$ answered when asked if he would enter a no-stipend hypothetical-GSoC: "*That's a weird question, the point of GSoC is the stipend, [otherwise] there wouldn't be any GSoC.*" On the other hand, we also identified a minority of students (3) with negative sentiments towards participation motivated by payments. As S$_{52}$ mentioned: "*There are many people who try for GSoC merely for the money! That's something of [an] utter shame. People should contribute only if they're genuinely interested and not for the money.*"

Learning

Several students reported that the potential **learning** (58) experience provided by GSoC was among their motivation for participation, mostly for the REAL-WORLD DEVELOPMENT EXPERIENCE (51), which means that the students wanted to improve their programming skills or be introduced to software engineering practices. As S$_{67}$ detailed: "*Was looking for an internship/summer experience and GSoC caught my eye because it seems like a good way to improve one's programming skills (...).*"

We also found evidence of some students motivated to enter GSoC because they wanted to GAIN OTHER SKILLS (2)(other than programming), such as S$_{99}$, who described his interest: "*To improve English.*" In addition, a few students vaguely mentioned *learning* (5), without specifying what they wanted to learn.

Academic

Not surprisingly, several students entered GSoC for **academic** (24) motives, mostly as a way to make money during summer breaks. This motivation is in accordance with GSoC's goals (goal *iv*: "*flip bits, not burgers.*")

While a few students vaguely reported participating in GSoC for *academic* (7) concerns, others wanted an alternative to traditional INTERNSHIPS (6). These students often indicated as a primary

[10] https://www.quora.com/How-did-you-spend-your-GSoC-stipend

motivating factor, the flexibility that GSoC offered, such as working remotely. The quote of S_{109} exemplifies these cases: "*It was a good summer internship, getting good internship locally was difficult for me.*" The work conditions offered by GSoC motivated another student. As S_{118} explained his interest: "*[I] needed a [low-pressure] internship like thing.*"

Similarly, other students driven by *academic* motives mentioned the need for the accomplishment of SUMMER PROJECTS (9). As S_{58} said: "*I was looking for a summer project.*" Due to the similarity, we grouped the concepts INTERNSHIPS (6) and SUMMER PROJECTS (9) into a single INTERNSHIPS/SUMMER PROJECTS (15) concept. Also, graduate students mentioned participating in the program for RESEARCH PURPOSES (4), such as S_{130}, who commented: "*I was a graduate student looking for summer funding and I wanted to improve my coding for my research.*"

An interesting observation is that while most students did not report a conflict between GSoC's and universities' schedules, a student said that his university (from Germany) did not encourage participation due to schedule and funding conflicts. As S_{111} replied: "*GSoC schedule collides with the lecture period, and will impact the ability of students to study full time. My university also disliked the option of abusing GSoC to fund research assistants (happens quite often elsewhere).*"

During the interview, two students added that the participation in GSoC could be used for obtaining COURSE CREDITS (2) in their college. As S_5 said: "*There are some students I know [from his college] that specifically did GSoC just for the college course credit.*"

Technical challenge

Approximately 67% of the students considered *technical challenge* essential for participation (see Q5 in Figure 4.2 and Figure 4.3). It was the motivation factor for which the largest number of students declared they would not enter GSoC without and that the students were least divided. This finding resonates with Lakhani's [LW05], who also found that intrinsic motivation—such as being motivated by technical challenge—is the main driver of volunteer contribution to OSS projects.

Surprisingly, analyzing our coding, we found that TECHNICAL CHALLENGE (5) was the least mentioned motivation factor (see Table 4.2), with only a few mentions. Still, these mentions were subtle. For instance, S_{72} said: "*Because it [is] really good chance for a student. It's challenging, it's interesting, and it's [paid].*" Nevertheless, several students (≈67%) considered this motivation factor as essential for participation (compare to Figure 4.2 and Figure 4.3).

One possible explanation is that the students' classification on how essential *technical challenge* may be moderated by social desirability [L. 59], which occurs when respondents answer questions in a manner that will be viewed favorably by others or to feel good about themselves. More research is necessary to verify if this is the case.

Answer for RQ4.1: Based on our data, the students typically entered GSoC for experiencing the dynamics of the contribution to OSS, which includes several participation rewards such as technical learning, branding, career building, intellectual stimulation, and the stipends. These participation rewards serve both as incentives for entering the program as for them to go through to the end, achieving established goals and delivering artifacts on time.

4.3.2 Students' motivations to engage in OSS projects via GSoC from mentors' perspective

In this Section, we present our findings on what motivated the students to enter GSoC according to the mentors in our sample (RQ4.2). As with the students, we show the motivations in 3 different figures.

Figure 4.5 depicts in stacked bars the mentors' assessment on how essential the investigated motivation factors were for students to join GSoC. Figure 4.6 offers 2 additional perspectives. Figure 4.6 (a) highlights in a graph the counts, proportions, and relationships between motivation factors, with each node indicating the number of mentors who considered that motivation essential for participation. Node sizes are proportional to mentors' assessments counts. The edges depict the mentors' count in the intersection of a motivations' pair. Percentages show the proportion that edge values represent in relation to a node (i.e., motivation). Figure 4.6 (b) decomposes the counts into sets and subsets in a Venn diagram. We discuss the Figure 4.5 and Figure 4.6 in greater detail in the following Sections.

As with the students, for better readability, we adopt the following convention for presenting our categories and concepts, as shown in Table 4.2. Concepts are presented as the following example: CONCEPT (1). Categories are presented in italic, such as *category* (1). Totals are presented in bold, such as **total** (1). In all cases, the numbers in parentheses depict the counts.

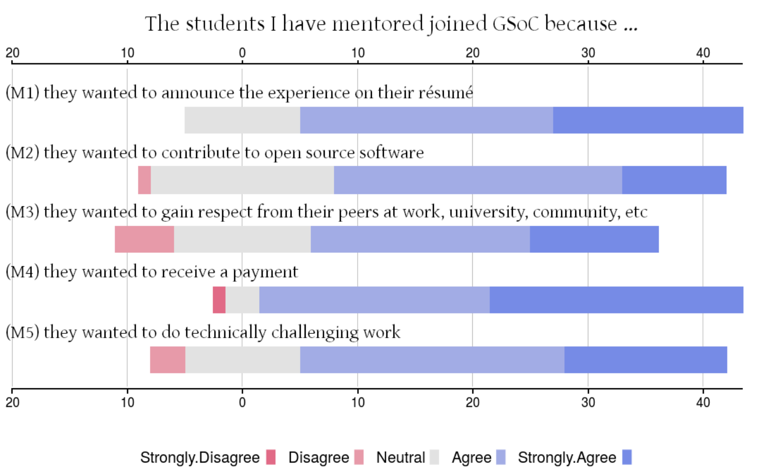
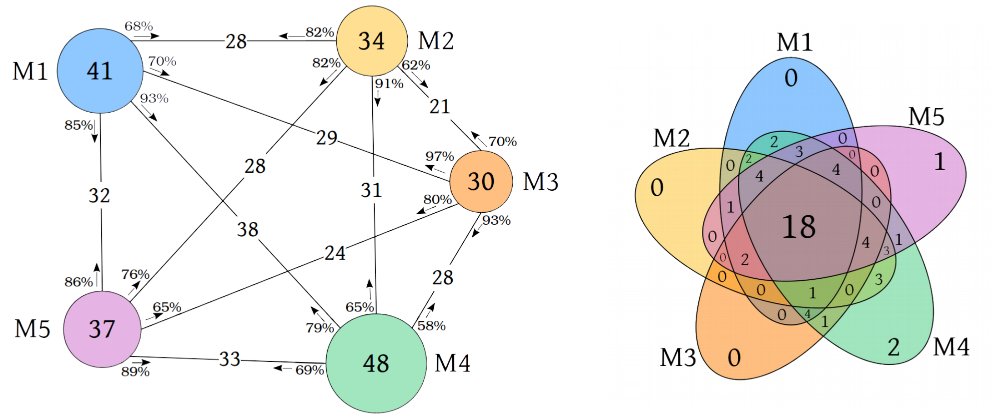


Figure 4.5: *Mentors' perception on the students' motivation for entering GSoC*

Career building

Approximately 77% of mentors agreed (or strongly agreed) that students entered GSoC for announcing the experience in CVs (see M1 in Figure 4.5 and Figure 4.6). It is worth noting that *career building* was the only motivating factor for which no mentor disagreed that it was essential for students. Mentors who were neutral (10) and the ones who did not know the answer (2) totaled together ≈23% of the sample.

In Figure 4.6 (a), we can observe that virtually all the mentors who agreed that *career building*



(a) Nodes represent the number of mentors who agreed (or strongly) that a motivation was essential. Edges represent the number of mentors who agreed (or strongly) for two motivations

(b) Venn diagram representation of the surveyed mentors who agreed (or strongly) that a motivation was essential for participation

Figure 4.6: *Count of students' motivation according to mentors in a graph (a) and in a Venn diagram (b). Career building (M1); contribution to OSS (M2); peer recognition (M3); stipends (M4); technical challenge (M5)*

was essential (M1, edge: 93%) also agreed that *stipends* were essential (M4). The remaining edges equally show that more than 2/3 of the mentors in M1 also considered the remaining motivation factors essential for participation.

Figure 4.6 (b) shows that no mentor considered that students were only trying to improve their CVs by participating in GSoC. Instead, mentors tended to assess students' motivations as being multifaceted, to the point that approximately 1/3 of the mentors (i.e., 18 mentors) considered all motivation factors essential for participation.

In their textual answers, some mentors mentioned CV IMPROVEMENT (9) as a motive for students to enter GSoC. As M_{36} representatively said: "*They [the students] are [mostly] interested in building their CV, being recognized as part of a Google's program.*"

Contribution to OSS

Around 64% of mentors agreed that students joined GSoC motivated by the *contribution to OSS* (see M2 in Figure 4.5 and Figure 4.6). It is interesting to notice that, while *contribution to OSS* was the second most essential motivation factor in the students' perception, mentors' assessment was that *contribution to OSS* is the second least essential factor (compare Q2 in Figure 4.2 to M2 in Figure 4.5). Besides, it was the motivation factor for which mentors had the highest number of neutral responses.

Mostly, mentors perceived students as contributors to either OSS projects (see Table 4.7) or GSoC projects (see Table 4.8), though in several cases mentors classified contribution frequency as rare. This perception may explain why mentors possibly underestimated (compared to the other motivation factors) how essential *contribution to OSS* was for the students since, in the mentors' view, most students already had that experience.

We also found potential disparities among mentors' and students' perception regarding contributing to OSS before GSoC. In Table 4.7, we can observe that ≈13% of the mentors in our sample considered that students had never contributed to OSS, while ≈35% of the students self-reported to have never contributed to OSS before GSoC (compare to Table 4.4). On the other hand, while ≈3% of the mentors reported that students were frequent contributors before GSoC (see Table 4.7), 16% of the students self-reported to be frequent contributors (compare to Table 4.4). A similar disparity can be evidenced, when we compare the students' (Table 4.3) and mentors' (Table 4.8) perception of the frequency of previous contributions to GSoC projects.

These disparities can be, in part, explained considering that the students that mentors referred to were not necessarily GSoC first-timers; were active project contributors before GSoC, and started contributing to OSS/GSoC projects to increase the odds of being accepted in GSoC. Complementarily, another plausible explanation is that students' and mentors' views differed towards what they considered to be a frequent contributor.

Table 4.7: *In your experience, how often were your GSoC students contributors to OSS software projects (other than their own) before the program?*

Responses	Count (%)
I don't know	4 (6.4)
Never	8 (12.7)
Rarely	20 (31.7)
Occasionally	19 (30.2)
Frequently	2 (3.2)

Table 4.8: *Were they already contributors to the project you mentored before GSoC?*

Responses	Count (%)
I don't know	0 (0.0)
Never	23 (43.4)
Rarely	14 (26.4)
Occasionally	14 (26.4)
Frequently	2 (3.8)

Table 4.9: *How often do students keep contributing to the projects you mentored after the program?*

Responses	Count (%)
I don't know	1 (1.9)
Never	8 (15.1)
Rarely	16 (30.2)
Occasionally	18 (34.0)
Frequently	10 (18.9)

Figure 4.6 (a) and Figure 4.6 (b) show that mentors perceived a strong link between the *contribution to OSS* and *stipends* motivation factors. We can observe that 91% of the mentors who considered *contribution to OSS* an essential motivation factor did the same for *stipends* (see M2 in Figure 4.6a). The remaining motivation factors also had more than 2/3 of the mentors who considered them essential, except for *peer recognition* (M3).

The coding of mentors' answers revealed that mentors rarely mentioned the **contribution to OSS** (8) projects as a motive for participation. We found some generic mentions to *contribution* *to OSS* (2), without any further detail. Most mentions pinpointed the INTERACTION WITH OSS COMMUNITY MEMBERS (5) as students' primary interest, even though there was a subtle mention to the OSS PROJECT ITSELF (1) as a motive.

We also found evidence from mentors (1) that the GSoC selection process can potentially make candidates contribute to OSS projects as a means to get accepted in the program (1). As M_9 commented: "*So the strategy I want to [implement] for later years is to make a call for students way before GSoC and try having them interacting with the project for a few months, even if for a few hours a week, before the submission period. They would [then] be able to work together with me to work on a project idea that is tailored to what they like best, and that fits with my interests.*"

Peer recognition

Around 57% of mentors considered *peer recognition* an essential motivation factor for students, being the least essential when compared to the other studied factors (see M3 in Figure 4.5 and Figure 4.6). This finding is coherent with students' assessment, which also considered *peer recognition* the least essential motivation factor.

In Figure 4.6 (a), we can observe that virtually every mentor who considered *peer recognition* essential also did the same for *career building* (see M3, edge: 97%) and *stipends* (see M3, edge: 93%), although more than 2/3 of mentors considered the other motivation factors essential. Figure 4.6 (b) shows that no mentor considered *peer recognition* alone essential for participation.

In their textual answers, mentors rarely mentioned **peer recognition** (2) as a motive for participating in GSoC, and we only found two subtle mentions. M_{15}, who broadly mentioned the theme: "*Kudos and getting paid*" and M_{27}, who was more specific: "*...for bragging rights.*"

Stipends

According to mentors, the *stipends* were an essential motivation factor for students to participate in GSoC (see M4 in Figure 4.5 and Figure 4.6), with a consensus of ≈91%. It was also the motivation factor for which mentors were the least divided. We can see in Figure 4.6 (a) that most mentors classified students' motivation as a combination of *stipends* and other factors, typically *career building* (≈79% of cases). In Figure 4.6 (b), we can observe that two mentors judged that the *stipends* alone sufficed for students to enter GSoC.

The coding of mentors' answers was consistent with the previous finding, showing that the **stipends** (24) were the most cited motivation factor for participation (see Table 4.2), even though often the mentors mentioned the *stipends* (21) broadly, without offering any context. Nevertheless, in a few cases, it was possible to discern how mentors interpreted the role *stipends* played for students. A few mentors mentioned stipends as a SOURCE OF FUNDING (2). For instance, when M_{40} commented on what his students were most interested in by entering GSoC: "*Money. Honestly, they're students, which I'm pretty sure is a synonym for starving and broke.*" We also could find evidence for CURRENCY CONVERSION (1) as a motive for participation. For example, M_{10}, who said: "*The money seems to be a strong incentive. Especially in countries where approx $5,500 USD carries a lot of purchasing power.*" No mentor mentioned STIPENDS AS COMPENSATION (0) as a motive.

Additionally, while several mentors who commented on *stipends* as a motive implied a neutral (30) or positive (1) tone in their answers, some mentors (3) indicated a negative tone. As M_2 said: "*Sadly, the money,*" and; M_{46}: "*I guess good students are more interested in learning and contributing, and not so good students by improving their CV and money,*" and; M_{33}, who commented: "*Many of the students I have mentored (15 or so at this point?) seemed to want to do the bare minimum to pass their deadlines and get paid.*"

Encouragingly, we found evidence of mentors with a different experience. As M_{11} said: "*Money is a strong motivator to join the program obviously, but most of them continue contributing after that factor disappears.*"

Technical challenge

Approximately 70% of mentors agreed that the TECHNICAL CHALLENGE (2) that the GSoC projects placed on their students is something the students had aimed for (see M4 in Figure 4.5). However, as with the students' answers, the TECHNICAL CHALLENGE (2) motivation factor had few mentions in mentors' coding.

Academic

Several mentors mentioned that **academic** (6) concerns motivated students to enter GSoC. Except for a single generic mention to *academic* (1) as a motivation factor, mentors identified that their students entered GSoC for COURSE CREDITS (1), for RESEARCH PURPOSES (4), and INTERNSHIP/SUMMER PROJECTS (4).

Interestingly, one mentor linked participating in GSoC to students' academic schedules. As M_{48} said: "*Students did stay involved for as long as they [were] in school. Once they find a job, they do [disappear].*"

Learning

Several mentors commented that **learning** (17) plays a central role in motivating students to enter GSoC. Only a few mentors mentioned *learning* (4) broadly. More commonly, mentors linked *learning* to the acquiring of REAL-WORLD DEVELOPMENT EXPERIENCE (13). As M_{13}, who explained his perception of why their students entered GSoC: "*(...) [for the] learning new software development practices and [the learning of] tools they [the students] wouldn't have learned on their own.*"

Answer for RQ4.2: Based on our data, we can interpret that the mentors in our sample perceived their students as entering GSoC for the technical learning, in a favorable environment, which the mentors portrayed as including stipends and mentoring, mainly for building the students' career portfolio.

4.4 Discussion

In this Section, we review and discuss our findings. In this research, we investigate whether the introduction of incentives offered by Summer of Code programs add new elements to the students' motivation.

(RQ4.1) Our research is the first to document what motivates students to participate in Summer of Code programs (Table 4.2). The motivation factors we found have never been studied in a context where everyone gets paid, so even if some of the factors are similar to the context in which OSS developers voluntarily contribute to OSS projects (see [vKHSW12] for a review) the contribution the projects through Summer of Code context is quite different, leading to a different prioritization of factors. Additionally, three motivating factors seem to be new: participate in GSoC for TAKING ADVANTAGE OF CURRENCY CONVERSION; OBTAINING COURSE CREDITS; and LOWERING OSS PROJECTS' ENTRY BARRIERS.

(RQ4.2) We also document the mentors' perception of the students' motivation to participate in a Summer of Code program (see Table 4.2), which is also not targeted by previous research. Mentors provide a perspective that considers the project's point of view, the comparison to non-GSoC newcomers, and an external view of the students' motivation to enter Summer of Code programs. In essence, mentors perceived students' motivation as a pursuit of tangible rewards such as *stipends*, and the *learning* of *technical skills* to be used mainly for *career building.*

Regarding students' retention, our findings suggest that most students do not remain contributing to GSoC projects after the program, regardless of their initial intentions (see Table 4.5). This finding is supported by our previous work [SWG$^+$17], in which we found that most students stopped contributing after GSoC, while the students who remained had only a few commits to the GSoC projects. Encouragingly, as with the findings of this research (see Figure 4.2 and Figure 4.3), our previous work [SWG$^+$17] indicated that some students became frequent contributors after GSoC. Thus, it seems that most students enter the program for an enriching (work) experience that cannot be detached from the name of a high profile software company (such as Google). In this sense, our findings suggest that most OSS projects should expect feature development from participating in GSoC.

Furthermore, our findings suggest that students are reluctant to admit financial motivation according to mentors' answers. Our data (see Table 4.1) did not allow us to investigate in-depth a mentors' belief that residing in a developing country increases the interest in currency conversion rates (i.e., financial motivation). Nevertheless, we could notice that for students with 2 to 5 years (61 students) of previous software development experience would still enter a hypothetical-GSoC that did not offer any stipends, as opposed to the ones with the same time experience who would not (20). In contrast, the students with 10 or more years (15 students) of prior development experience would not enter a hypothetical-GSoC with no payments, as opposed to the ones within the same experience range (5) who would still enter. Therefore, although the stipends are an essential motivator, it seems to be essential for participation for students with high software development experience, while the students who lack development experience value participation in GSoC for boosting their careers.

Indeed, low retention levels (or high levels of absenteeism in some contexts) is the most expected outcome in volunteer engagement programs (see [Smi14] for the firefighting community in the USA; [LMS13] for blood donation, and [RK09, ZZH^{+}13] for online communities). Encouragingly, regardless of their motivation for entering GSoC, students self-reported an increase of their previous contribution level to the assigned GSoC projects in ≈53% of cases (see Figure 4.4).

Our findings revealed that there are students whose primary goal was to participate in GSoC, not necessarily to contribute to OSS projects. We speculate that these students would not have contributed to OSS projects otherwise. In addition, we conjecture that Summer of Code programs can potentially assist students in overcoming several of the onboarding barriers reported by Steinmacher et al. [Ste15], which can be investigated in future research.

Previous research reports positive associations between receiving stipends and participating in OSS projects (e.g., [RHS06]). Nevertheless, we found that the goals among stipend-driven students can be different. As shown, while some students understand the stipend as compensation for service, others need it for living expenses or buying hardware equipment. Our findings trigger some questions to future research to understand these associations at a finer-grained level.

4.5 Limitations and Threats to Validity

This research has several limitations, which we here both acknowledge and report how we aimed to mitigate them.

Internal validity: Surveys are typically subject to *sampling* bias, namely *self-selection* bias, which could distort our sample towards the students and mentors who chose to participate in this research. Also, our sample of students and mentors is not sufficiently large for statistically grounded inferences. These threats could result in a biased sample, in which case it would not be representative of the actual population of students and mentors. Nevertheless, our focus is not on understanding how generalizable the motivation factors we found are but on identifying them.

Also, we cannot rule out that some responses were not affected by *social desirability* bias. Our findings show that some students and mentors classify (at least some) motivation factors as desirable or undesirable, which can potentially distort our results towards these preconceptions. In particular, we offer the stipends' classification as an example. Our data include negative viewpoints of students towards stipend-driven participations, which could indicate that a more significant number of students can perceive this motivation factor as undesirable, underreporting (conscious or not) how essential the stipends indeed were for their engagement in OSS via GSoC. We expect to motivate future research on this aspect.

External validity: The main limitation affecting external validity is our focus on GSoC. Also, we only investigated the GSoC editions from 2010 to 2015. Furthermore, as few respondents identified themselves as female or other, our results may be biased towards male students. Although we are confident that most of our results are also valid in other settings, we leave this investigation to future research.

Construct validity: Another threat is the data classifications' subjectivity. We used coding procedures to mitigate this threat, given that our findings are grounded in the data collected. Additionally, we discussed the analysis process, codes, concepts, categories, and the findings among the authors to encourage a better validation of the interpretations through agreement.

As mentioned, while we report the amount of evidence for each theme yielded by the data analysis, we cannot infer the strength or pervasiveness of a theme from these numbers since we did not explicitly ask all participants about each theme specifically. However, the data collected through Likert-scale questions in the survey and the follow-up interviews confirmed our coding scheme.

4.6 Final Considerations

In this Chapter, we investigated what motivated students to participate in Google Summer of Code (GSoC). More specifically, we collected and analyzed answers to a questionnaire administered with 141 students and 53 mentors that participated in different GSoC editions (2010-15), comprising several OSS projects, followed by ten confirmatory interviews.

We established two research questions. In RQ4.1, we surveyed students to understand their motivations to enter and contribute to OSS projects. We evidenced that multiple motivations drive the students, and our findings suggest that the typical students participate in GSoC for work experience, rather than becoming a frequent OSS contributor. Our findings revealed that the students considered essential for participation: a *technical challenge*, *contributing to OSS*, *build their careers*, *stipends*, and *peer recognition*. *Learning* and *academic* concerns were cited as important motivation factors.

In RQ4.2, we surveyed mentors to understand how they perceived their students' motivation since their impressions represent an external view on an aggregate of students. We found that mentors mostly considered students' motivation as a pursuit of tangible rewards, such as stipends and technical learning, to be used for career building.

We concluded this work by highlighting that the participation in Summers of Code provided some OSS projects with new collaborators, even though it is not the typical scenario. For the OSS projects which need to attract and retain newcomers, we stress that participating in GSoC should be viewed as an investment (regarding effort). For the students who kept contributing after GSoC, contributions tended to diminish, which can signal to OSS projects that they should use strategies for handling these students. A strategy can be as simple as sending feedback through email, explaining the importance of the students' contribution, which prior research showed to increase intrinsic motivation [ZZH+13]. Also, OSS projects can establish a recommended period before Summers of Code for applicants to start contributing and interacting with project members. Thus, applicants who start earlier and contribute more would have more acceptance chances.

Chapter 5

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

In this Chapter, we performed the study S4, as depicted in 1.1. Here, we identified the tasks that the OSS projects in our sample typically proposed to students and offered a theory of how the accomplishment of such tasks can motivate students to engage in OSS projects via Summer of Code programs.

We employed the term *engagement process*, which refers to all the tasks students need to accomplish in Summer of Code programs. We stress that we use the term *process* following Ralph's definition of *process theory* [Ral15]: *"a system of ideas intended to explain (...) how an entity changes and develops."* Here, the entity is the students who participate in Summer of Code programs as they are the ones that can become contributors after the program. It is worth mentioning that our use of *process* should not be confused with *methods* such as Scrum [SS18], Lean [PP03], and the V-model [FM92]. Additionally, we use the term *process* not as a set of steps in a specific order (i.e., an algorithm) as the tasks that comprise the process do not necessarily posit sequential orders. Thus, *engagement process* should be understood here as the set of ideas that can explain how students engage in OSS via Summer of Code programs by accomplishing the tasks proposed by the OSS projects.

The content of this Chapter was submitted to the Computer-Supported Cooperative Work Conference (CSCW) 2020. We modified the submitted manuscript for cohesion and readability.

5.1 Research Questions

We present the thesis' research questions (RQ) for better readability.

RQ1. How do OSS projects mentor students in Summer of Code programs?

RQ2. How do Summer of Code programs motivate students to engage in OSS projects?

5.2 Research Method

In this section, we present how we designed this research. Following Ralph's guidelines on how to develop theories [Ral18], we built two separate but interrelated process theories (one for each RQ), as depicted in Figure 5.1. We present them separately for better visualization. Although we present them as two separate theories, they form one process theory, which we refer to as the *engagement theory* in OSS projects via Summer of Code programs.

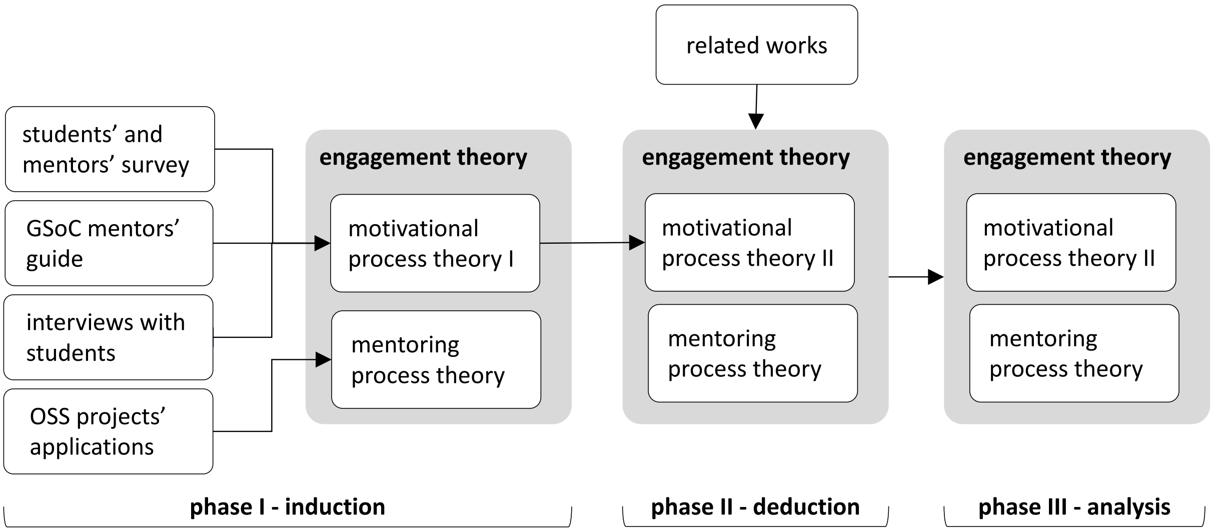


Figure 5.1: *Research design*

We split the building of each process theory into three phases. In Phase I, we grounded the process theories in empirical data (i.e., *induction* [Gre06]). In Phase II, we reviewed the literature (i.e., *deduction* [Gre06]). In Phase III, we analyzed the process theories in light of the perceptions of college students who did not participate in Summer of Code programs. We detail each phase separately in the following.

5.2.1 Phase I: Building the engagement process theory

To find empirical support for RQ1, we searched for data that could show us the tasks that OSS projects accomplish to mentor students in Summer of Code programs. For joining GSoC, OSS projects need to apply for the program. We used Google's search engine to find application forms that OSS projects made publicly available. By using the questions of 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 several applications (88) of distinct OSS projects applying for GSoC. We randomly analyzed 25 applications before reaching saturation. Table 5.1 lists the OSS projects' applications we analyzed, and it contains the OSS project name, the project's website, and the year that the OSS project applied for GSoC. We marked with the value N/A the cases that the project's application year was not available. For analyzing the OSS projects' answers, we employed the process of coding, which consists of assigning words or phrases to portions of unstructured data [Sal15]. We followed Charmaz's constructivist approach [Cha06], which divides the process into three steps: (*i*) *initial coding*; (*ii*) *focused coding* and *categorizing*, and (*iii*) *theory building.* As a

Table 5.1: *List of OSS projects*

OSS project name	Website	Application Year
Apertium	https://www.apertium.org	2017
Asciidoctor	https://asciidoctor.org	2016
C4	http://www.c4ios.com	N/A
checkstyle	https://github.com/checkstyle/checkstyle	N/A
Clojars	https://clojars.org2017	N/A
Coala	https://coala.io	N/A
Code4Nepal	https://github.com/Code4Nepal	2018
Drupal	https://www.drupal.org	2016
Gentoo	https://www.gentoo.org	2018
GNU Octave	https://www.gnu.org/software/octave/	2017
Janitor	https://pypi.org/project/Janitor/	N/A
OpenAstronomy	https://openastronomy.org/	N/A
OpenCV	https://github.com/opencv/opencv	N/A
OpenStack	https://www.openstack.org	2016
openSUSE	https://www.opensuse.org	2018
phpMyAdmin	https://www.phpmyadmin.net	2018
PMD	https://github.com/pmd/pmd/	N/A
Powerplant	https://github.com/Ecohackerfarm	N/A
R project	https://www.r-project.org	2018
Rapid7 Metasploit Framework	https://www.metasploit.com/	N/A
SciRuby	http://sciruby.com/	N/A
sympy	https://www.sympy.org/	N/A
The Document Foundation Wiki	https://wiki.documentfoundation.org	2016
The STE||AR Group	https://github.com/STEllAR-GROUP	N/A
XWiki	https://www.xwiki.org	N/A

result, we obtained a total of 34 concepts, 11 categories, and four major categories. We refer to this theory as the Mentoring Process Theory (Figure 5.2).

To find empirical data for answering RQ2, we investigated multiple empirical data sources that could show us how students are motivated to contribute to OSS projects through Summer of Code programs. First, we revisited the students' and the mentors' answers to draw connections between the previously identified participation rewards. In Chapter 4, we performed the steps (*i*) and (*ii*). In this Chapter, we carried out the theory-building process by accomplishing the step (*iii*).

Additionally, we investigated the perception of the GSoC program administrators on how students are motivated. In the case of GSoC, program administrators are Google's employees who run the program [The18]. We could not collect the program administrators' perceptions directly, so we used the GSoC mentors' guide [The18] as their proxy. The GSoC mentors' guide contains several guidelines for OSS projects on how to participate in GSoC, including some suggestions on how to motivate students to become lifetime contributors. We refer to this theory as the Motivational Process Theory (Figure 5.3).

Furthermore, for all grounded theory procedures in this study, there was a collaboration with other researchers for more solid interpretations of our coding through mutual agreement.

5.2.2 Phase II: Reviewing the literature

In Phase II, we aimed at grounding our previous empirical results in the relevant literature. We reviewed works that targeted **Summers of Code**, **motivation**, and **engagement** in OSS projects.

We presented the works we reviewed in Chapter 2.

5.2.3 Phase III: Analyzing the motivational process theory's usefulness

In this Section, we show the usefulness of the Motivational Process Theory to *students with no previous participation in Summer of Code programs* here referred to as non-participants. Although Grounded Theories do not need to be *'validated'* in the same sense as other theories (because they are grounded in empirical data) [Ral18], we wanted to understand how non-participants' perceptions of Summer of Code programs change when presented to our theory. We stress that our intention here is not to confirm the theory but merely to assess its usefulness. It is worth noting that, since we targeted non-participants, we only showed them our Motivational Process Theory, instead of the Engagement Theory. Moreover, to our best knowledge, no research has previously targeted these students, which may offer us insights about how to conduct/engage in OSS via Summer of Code programs. Furthermore, there is some anecdotal evidence that these students often carry misconceptions, such as "*only computer scientists are accepted*," and "*participation is too tough*" [Dug12].

Data collection. We choose to survey non-participants because it enables us to contact a larger number of students in a structured manner. We used questionnaires as a data collection method, following Fink's advice on how to design surveys [Fin95]. We split the survey into three steps. In the first step, the non-participants answered a questionnaire[1] containing four questions. We asked non-participants whether they had heard of GSoC or similar (question 1). For the non-participants that had heard of the program, we asked them to describe what is GSoC about (question 2) and whether they had considered joining it (question 3). Next, we asked non-participants to read an explanation of GSoC (written by program administrators).[2] After that, we asked them to list (according to their opinion) what Summer of Code students get by participating in GSoC in order of importance (question 4).

In the second step, we instructed the non-participants to watch a 7.5-minute video.[3] It is worth saying that we prepared the video for this study to summarize the motivational theory. The non-participants were allowed to watch the video as many times as they wished. The video explains what GSoC is, how it works, and the Summer of Code students' motives to participate in GSoC and contribute to OSS projects.

In the third step, the non-participants had to answer a final questionnaire[4] containing six questions plus demographics. We asked them: what changed in their perception of GSoC (question 1); what have they learned (question 2); whether our results would influence their decision to participate (question 3); how does GSoC contribute (or not) to attracting new contributors to OSS projects (question 4); what could motivate other people to contribute to OSS projects through Summers of Code (question 5); what do Summer of Code students get by participating in GSoC in order of importance (question 6), and; demographics.

[1] https://pt.surveymonkey.com/r/HX9H7FX

[2] See the **What is Google Summer of Code?** section: https://google.github.io/gsocguides/student/

[3] https://figshare.com/s/88704bc89fac722ac073

[4] https://pt.surveymonkey.com/r/HN82XYJ

Data analysis. For analyzing non-participants' answers, we applied descriptive statistics and grounded theory procedures [Cha06]. In the case of the answers for what students get by participating in GSoC in order of importance (Q4 in Phase I, and Q6 in Phase III), non-participants gave us a reward list. We classified each reward, according to the motivation scheme presented in Table 4.2. We discarded all unclear rewards. For example, when a non-participant 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 were not listed in our motivation scheme, such as "*maturity,*" and "*organization.*" It is worth saying that although we discarded rewards, we maintained 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 non-participants 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 our motivational scheme (i.e., $b = 7$), r is the rank of a reward in an answer, and s is the final score of a reward.

Sampling. We used a convenience sample comprised of Brazilian and Chinese allegedly non-participants. We emailed ≈130 survey invitations. A total of 41 respondents completed all three steps (18 Brazilian and 23 Chinese). Some survey respondents did not complete all steps (4 Brazilian and 17 Chinese). After a preliminary analysis, we observed that some Chinese respondents had already participated in GSoC (2) or a similar program (1). Since our goal was to assess how our theory impacted non-participants, we excluded these instances from our analysis. Thus, our working sample is comprised of 38 non-participants (18 Brazilian and 20 Chinese).

5.3 Results

In this Section, we present our findings. Our findings are divided into concepts, categories, and major categories. For better understanding, we present concepts in SMALL CAPITALS, categories in *italics*, and major categories in **boldface** font.

5.3.1 Phase I: The engagement process theory

(RQ1) For obtaining empirical data on which tasks OSS projects accomplish to motivate students in Summer of Code programs, we analyzed the OSS projects' applications for GSoC. We found a significant number of strategic tasks that OSS projects accomplished to create engaging and meaningful experiences for students (Figure 5.2). Broadly speaking, OSS projects tasks can be grouped into four major categories: **planning**, **integration**, **rewarding**, and **execution** tasks.

Planning tasks. We grouped into this major category the tasks that OSS projects did before GSoC kickoff (Figure 5.2). OSS projects worked to establish a well-designed contribution context, which included establishing the tasks not only to manage and monitor the students but also to integrate and reward them for their contributions. For applying for GSoC, OSS projects typically started by FORMULATING COLLECTIVELY AN IDEA LIST that aimed at creating the best experience

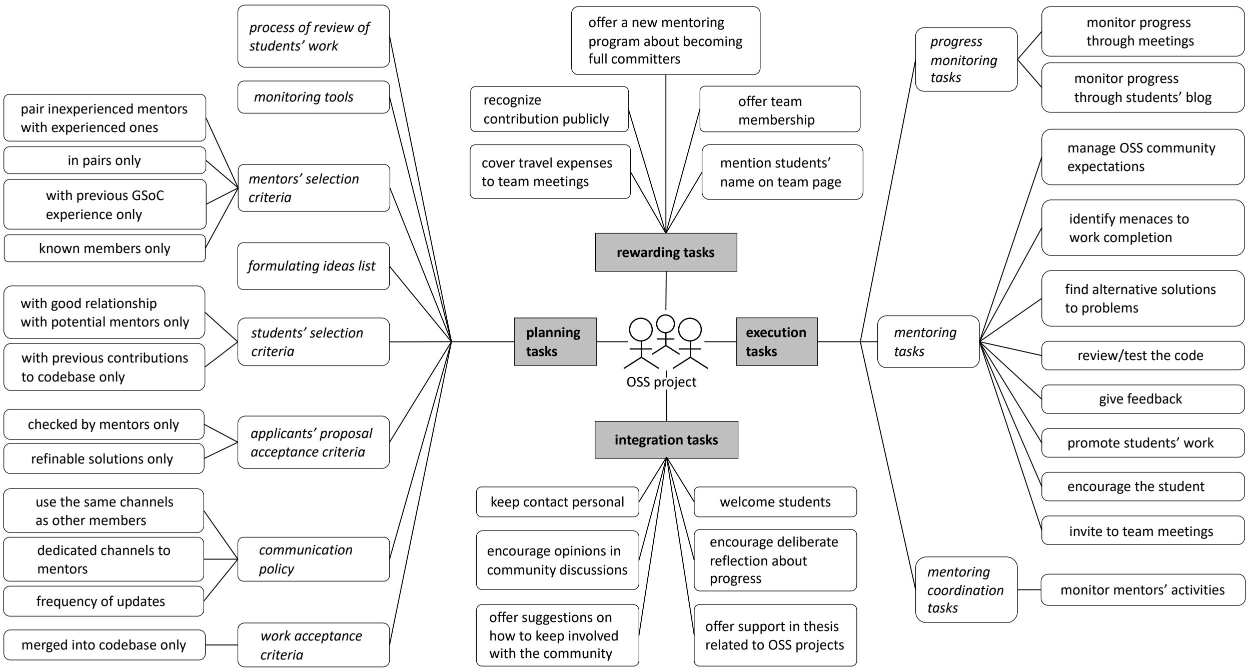

process of review of students' work
monitoring tools
pair inexperienced mentors with experienced ones
in pairs only
with previous GSoC experience only
known members only
mentors' selection criteria
formulating ideas list
with good relationship with potential mentors only
with previous contributions to codebase only
students' selection criteria
checked by mentors only
refinable solutions only
applicants' proposal acceptance criteria
use the same channels as other members
dedicated channels to mentors
frequency of updates
communication policy
merged into codebase only
work acceptance criteria
offer a new mentoring program about becoming full committers
recognize contribution publicly
offer team membership
cover travel expenses to team meetings
mention students' name on team page
rewarding tasks
planning tasks
OSS project
execution tasks
integration tasks
keep contact personal
welcome students
encourage opinions in community discussions
encourage deliberate reflection about progress
offer suggestions on how to keep involved with the community
offer support in thesis related to OSS projects
progress monitoring tasks
monitor progress through meetings
monitor progress through students' blog
manage OSS community expectations
identify menaces to work completion
find alternative solutions to problems
mentoring tasks
review/test the code
give feedback
promote students' work
encourage the student
invite to team meetings
mentoring coordination tasks
monitor mentors' activities

Figure 5.2: *The Mentoring Process Theory*

for students and addressing OSS projects' needs. It is worth mentioning that program administrators advise OSS projects that the program "*is about building the student's experience*" and that "*getting code in [the] project is a nice side effect*" [The18].

After acceptance into GSoC, some OSS projects reported worrying about being fair in ranking applicants' proposals, employing *applicants' proposal acceptance criteria* previously defined, such as only accepting proposals that were CHECKED BY MENTORS or that contained SOLUTIONS THAT COULD BE REFINED later by other members. Complementary, some OSS projects employed *students' selection criteria*, deciding to only accept applicants WITH GOOD RELATIONSHIP WITH POTENTIAL MENTORS and WITH PREVIOUS CONTRIBUTIONS TO CODEBASE.

In several applications, good communication was described as a key to successful participation. Thus, some OSS projects instituted a *communication policy*, which defined which channels should be used for communication with the students. Frequently, OSS projects used the SAME CHANNEL USED BY OTHER MEMBERS. In some cases, mentors used dedicated MENTORS' COMMUNICATION CHANNELS, allowing them to ask more experienced members freely. Also, a *communication policy* defined the FREQUENCY OF UPDATES students should provide, which was used to not only manage the OSS projects' expectations towards the project completion but also to identify students' drop out.

OSS projects employed *mentors' selection criteria* for finding mentors with a good fit for the students. For example, OSS projects defined 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.

Some OSS projects faced difficulties in deciding when to accept students' work. In these cases, OSS projects defined students' *work acceptance criteria*, such as accepting code that was MERGED

INTO CODEBASE ONLY. In addition, to keep track of students' work, some OSS projects established MONITORING TOOLS, and a *review process of students' work* such as CODE INSPECTION.

Integration tasks. We grouped into this major category the tasks intended to integrate students into the social structure of OSS projects. Several OSS projects (13) mentioned to employ WELCOMING TASKS as a strategy to keep students involved, sometimes even before GSoC kickoff. As an OSS project explained: "*(...) we embrace you [the student] warmly and without condition.*" Other tasks aimed at making students experience what project members regularly do, such as ENCOURAGING EMITTING OPINIONS IN DISCUSSIONS. An OSS project mentioned ENCOURAGING DELIBERATE REFLECTION about progress as a strategy to keep students involved in the OSS projects during GSoC. Tasks to integrate students after GSoC included KEEPING PERSONAL CONTACT, OFFERING SUGGESTIONS ON HOW TO KEEP STUDENTS INVOLVED WITH THE OSS PROJECT, and OFFERING SUPPORT IN THESES RELATED TO THE OSS PROJECT.

Rewarding tasks. We grouped into this major category the tasks intended to acknowledge students' contributions. OSS projects rewarded students' contributions in different ways. For example, an OSS project stressed the importance of RECOGNIZING STUDENTS' CONTRIBUTIONS PUBLICLY, especially to other members. Another rewarding strategy consisted of MENTIONING STUDENTS' NAMES ON THE TEAMS' PAGE to increase their exposure, both internally and externally, especially for career concerns. Some OSS projects could COVER TRAVEL EXPENSES TO TEAM MEETINGS, which aimed not only at deepening students' ties with other members but also at giving students a good chance for networking. For students with good performance, OSS projects offered TEAM MEMBERSHIP. Moreover, an OSS project offered a specific MENTORING PROGRAM ABOUT BECOMING FULL COMMITTER to high achieving students. In one instance, an OSS project mentioned that the project would not provide any external reward beyond those obtained from accomplishing the project's tasks, explaining that it is the students' role to motivate themselves.

Execution tasks. We grouped into this category the OSS projects' tasks intended at coordinating and mentoring students during GSoC. In this work, we refer to the mentoring and the coordinating tasks as *management*. The *mentoring tasks* consisted of REVIEWING/TESTING CODE; GIVING FEEDBACK frequently; PROMOTING STUDENTS' WORK such as tweeting great achievements or simply keeping other members posted about work progress; ENCOURAGING STUDENTS when they were demotivated; IDENTIFYING MENACES TO WORK COMPLETION such as checking for 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 could not be reached, and; INVITING STUDENTS TO TEAM MEETINGS. Moreover, several OSS projects instituted *progress monitoring tasks* 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 adopted *mentoring coordination tasks* such as MONITORING MENTORS' ACTIVITIES as a strategy to reduce the odds of failures.

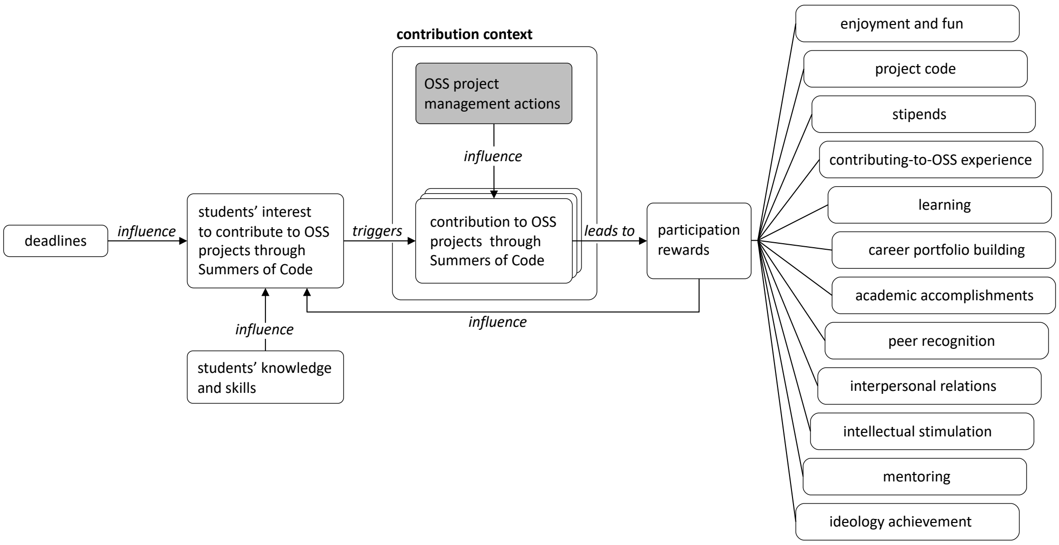


Figure 5.3: *The Motivational Process Theory I (i.e., grounded in empirical data)*

Answer for RQ1: To mentor students, OSS projects accomplish several tasks that go well beyond providing the students with the practical knowledge necessary for contributing to OSS projects. Although OSS projects' tasks differ for each project, they converged towards planning their participation, socially integrating the students, rewarding contributions and achievements, and management.

(RQ2) Figure 5.3 depicts our Motivational Process Theory, which is grounded in multiple empirical data sources. We adopted the construct *reward* because it is frequently used in the psychology literature to refer to what individuals expect to get for carrying out a certain behavior [DKR99]. Here *participation rewards* refer to what students expected to get when they participated in GSoC for the first time. Some participation rewards refer to motives related to the feelings that the contribution to OSS projects evoked in students such as ENJOYMENT and FUN. Similarly, 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 their LEARNING, which was often linked to the increase of job prospects. Several students were interested in rewards typically linked to traditional OSS developers' motives such as developing useful PROJECT CODE, having a CONTRIBUTING-TO-OSS EXPERIENCE, PEER RECOGNITION, IDEOLOGY ACHIEVEMENT, and developing INTERPERSONAL RELATIONS. Moreover, students also participated in GSoC for ACADEMIC ACCOMPLISHMENTS. Furthermore, students indicated different reasons for their interest in the STIPENDS, such as paying tuition, living expenses, or simply financial gain.

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

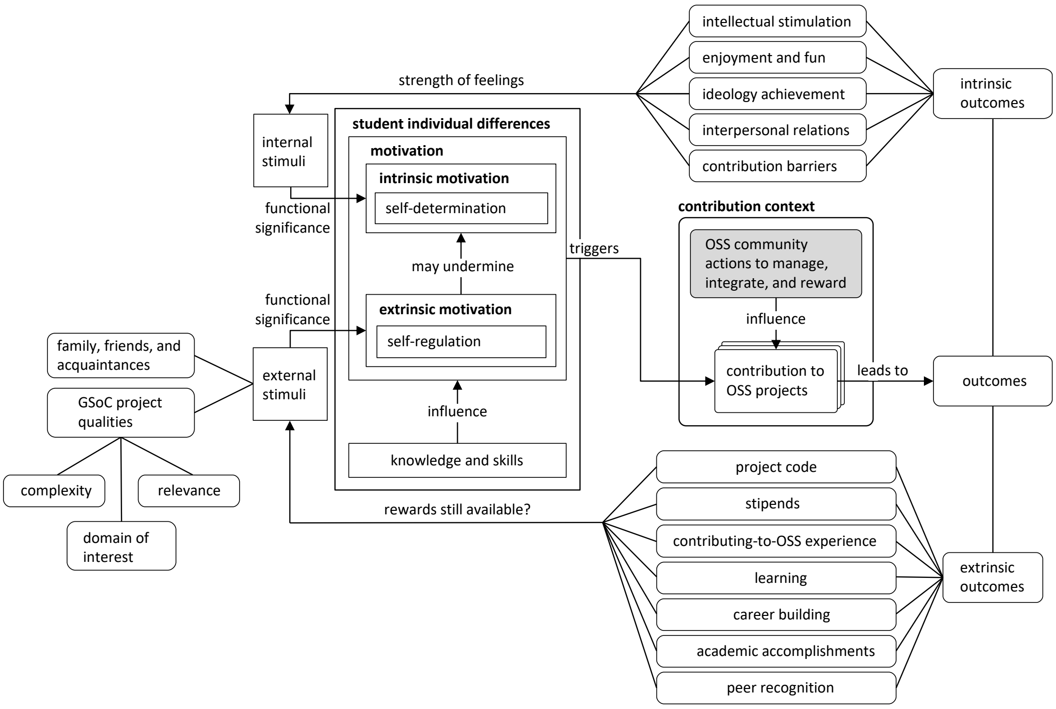


Figure 5.4: *The Motivational Process Theory II*

students. It is noteworthy that understanding the precise nature of the influence of external factors on students' interests and contributions comprises a gap that should be explored by future research, perhaps in the development of a variance theory.

While participation rewards seem to increase students' interest, their level of KNOWLEDGE and SKILLS seem to moderate their interest to contribute to OSS projects, at least in the case of students with more development experience. For example, students with 2-3 years of experience in software development reported being more interested in participating in Summers of Code and becoming frequent contributors than students with ten years or more. Similarly, our findings suggest that DEADLINES also had 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 were what made them meet agreed timelines.

Several students reported MENTORING as a key factor to achieving rich OSS-contribution experiences. In Figure 5.3, we can observe that students' CONTRIBUTIONS TO OSS PROJECTS THROUGH SUMMERS OF CODE occurred within a CONTRIBUTION CONTEXT and they are directly influenced by what OSS projects did for students (OSS PROJECTS' MANAGEMENT TASKS).

Phase I: In essence, the Motivational Process Theory I reveals that students engaged in OSS via Summer of Code programs because they were interested in obtaining a blend of rewards associated with participating in such programs. In addition, the students' interest may be moderated by the students' knowledge, skills, and external factors such as deadlines.

5.3.2 Phase II: Merging the engagement process theory with literature

In this phase, we show the changes to the Engagement Theory in light of the literature. Our literature review did not add new tasks to OSS projects. Thus, we did not make any changes to the Mentoring Process Theory (Figure 5.2). In contrast, our literature review allowed us to build our Motivational Process Theory II, which is grounded in all our data sources (Figure 5.4). We describe the changes in the following.

Student individual differences. In the Motivational Process Theory I, the contributions to OSS projects through Summer of Code programs are triggered by the students' *interest* in participation rewards. Besides, students' interest can be moderated by their *knowledge* and *skills*. In the Motivational Process Theory II, we modeled the constructs interest, knowledge, skills, and demographics as different dimensions of individuals, gathering them into STUDENTS' INDIVIDUAL DIFFERENCES. We adopted the construct *individual differences* because it is frequently employed in the psychology literature to encompass knowledge, personality, and abilities [Mar14].

Motivation. We changed the term *interest*, grounded in the students' and mentors' questionnaires, to *motivation*, which is the construct typically used in the literature as the psychological state that antecedes a certain behavior [RHS06, RD00, SDPR92]. In addition, we split motivation into *intrinsic motivation* and *extrinsic motivation*. Intrinsically motivated behaviors are performed out of interest, requiring no reward other than the enjoyment of performing them, while extrinsically motivated behaviors are instrumental in obtaining external rewards [SDPR92]. The literature documents both types of motivations in OSS developers (e.g., [LW05, TL02]).

Outcomes and Stimuli. We used the term *outcomes* in the Motivational Process Theory II instead of *participation rewards*. While the term *participation rewards* refers to positive things students expect to get, the term *outcomes* allows for positive, neutral, or negative things 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 [SDPR92]. For example, a contribution to OSS projects that does not lower *contribution barriers* may negatively affect students' feelings of autonomy and competence, diminishing their intrinsic motivation. Extrinsic outcomes, on the other hand, refer to outcomes that can become *external stimuli* to students' extrinsic motivation [SDPR92]. 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 Summers of Code. Students can interpret an outcome in different ways. For example, while some students interpreted *meeting deadlines* negatively as shown in Chapter 4, others framed the experience more constructively [Ndu16]. We employ the term *functional significance* [DKR99] to refer to the interpretation that students give to outcomes.

Knowledge and skills. The literature on contribution to OSS projects considers *knowledge* and *skills* one of the main drivers of participation [LW05]. It is one's set of motivations, combined with knowledge and skills that trigger one's behavior [MD03]. Baltes and Diehl [BD18] considered "*a developer's knowledge base as the most important aspect affecting their performance.*" The authors [BD18] continue explaining that general knowledge does not only refer to technical aspects or general concepts but also knowledge about and experience with successful strategies. In the context of Summers of Code, several students and mentors equated engaging in OSS via GSoC with a pursuit of knowledge and skills.

One question that may arise at this point is how Motivational Process Theory I and II differ. In essence, both theories have different explanatory power. Explanatory power is a measure of how good a process theory is [Ral18]. In the Motivational Process Theory I, students' engagement occurs due to the strength of their interest in *participation rewards*. It follows that if students engage in OSS via a Summer of Code for the same participation rewards, they should end with similar engagement levels, which is not what occurs.

In the Motivational Process Theory II, on the other hand, students' motivation is influenced by the *outcomes* of engaging in OSS via Summers of Code, which include participation rewards. In this way, each student (due to individual differences) can interpret outcomes as positive, neutral, or negative. Although students enter a Summer of Code 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 influenced some students negatively, as shown in Chapter 4, others framed it more constructively [Ndu16]. Also, the Motivational Process Theory II allows explanation for the cases when students' motivation was influenced by things they could not anticipate. For example, ineffective mentoring might lead to the maintenance of *contribution barriers* [Ste15], which in turn might lead to the negative stimulus *feeling of incompetence* that might diminish a student's motivation to repeat the experience. Thus, we believe that, by considering outcomes, we add explanatory power to the Motivational Process Theory II.

Answer for RQ2: The interplay of OSS projects' engagement via Summers of Code and students' contributions produces several outcomes, which can stimulate students' motivation in three ways. First, some outcomes enhance students' sense of competence, autonomy, and relatedness (i.e., intrinsic motivation), which motivates their engagement via Summers of Code. Second, some outcomes motivate their engagement via such programs because they are instrumental in achieving students' goals. Finally, some outcomes motivate their engagement via such programs when students internalize OSS projects' culture and values, which can prompt voluntary contributions.

5.3.3 Phase III: Analyzing the motivational process theory's usefulness

In this Section, we analyze the Motivational Process Theory II in light of the perceptions of non-participants. Most of them were between 18-25 years old (Figure 5.5a). Although most of them were male, gender distribution was different for Brazilian and Chinese non-participants (Figure 5.5b). While in Brazil, we obtained 50% of females and males in China, 5% of the responses were by females and 95% by males. No one self-declared as other.

The majority of the Brazilian non-participants had never contributed to OSS projects, while the Chinese non-participants had mostly contributed as part of a school project (Figure 5.5c). Most non-participants had not heard of GSoC before the survey (Chinese: ≈95%; Brazilian: ≈70%). The one Chinese non-participant that had heard of the program described GSoC accurately, claiming

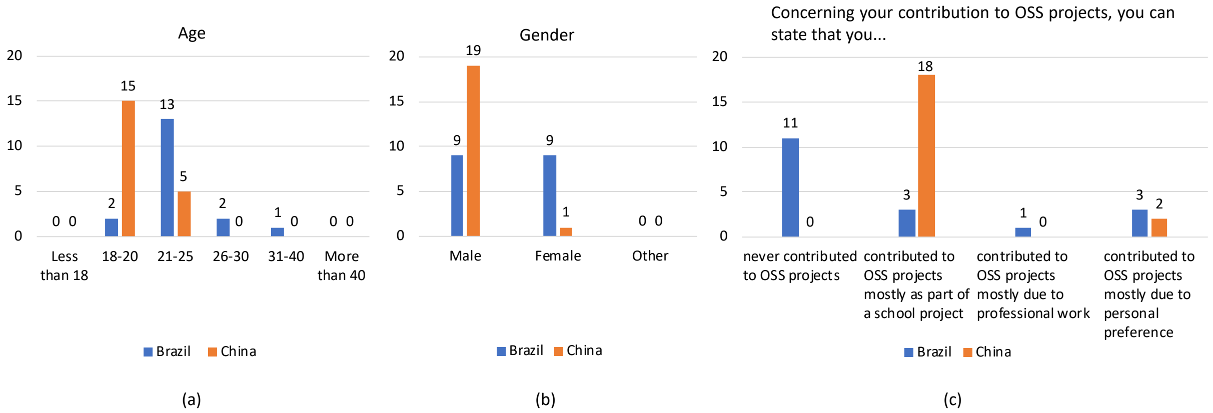


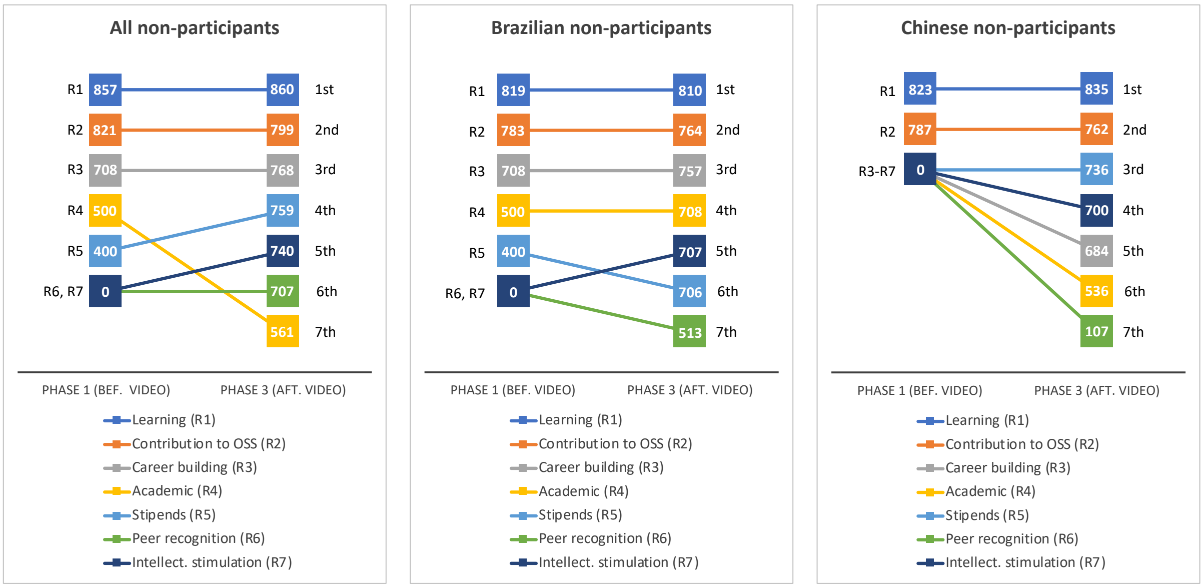
Figure 5.5: *Non-participants' demographics*

that his knowledge came from his efforts to join the program. On the other hand, the Brazilian non-participants' descriptions (2) of GSoC were at best simplistic. For example, P2 understood GSoC as a program "*that gives scholarships to developers.*" Additionally, other descriptions were inaccurate (2), such as P9's that described GSoC as "*an event that gives scholarships to developers,*" and P3's that described the program as "*a platform for students to code.*" Interestingly, with one exception, even contributing to OSS projects as part of a school project, the Chinese non-participants had never heard of GSoC (or similar) before this survey (Figure 5.5c).

Figure 5.6 presents the ranking and scores of each participation reward listed in our motivational scheme. After reading GSoC's description, the majority of non-participants (20) ranked LEARNING as the most important reward, and several others (8) ranked it second. Similarly, several non-participants (10) ranked CONTRIBUTION TO OSS first, some (5) ranked it second, and a few (2) ranked it third. Other rewards were mentioned rarely or did not mention. Interestingly, Figure 5.6 also shows that several non-participants knew nothing to little about some rewards. For example, while Brazilian non-participants did not mention PEER RECOGNITION (R6) and INTELLECTUAL STIMULATION (R7), Chinese non-participants did not mention several others (R3-R7).

After watching the video, most non-participants reported changes in their perception of GSoC. Most commonly, they reported having acquired a more comprehensive understanding of the program. As P10 representatively said: "*The video exposes several points that I had not thought of (...) [,such as] being able to develop projects [with] values that I believe.*" Encouragingly, several students could learn 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 5.6 shows different changes in non-participants' perception about the importance of the rewards. Except for LEARNING and CONTRIBUTION TO OSS, which remained stable, the other rewards varied greatly. For example, we can observe that the Chinese non-participants did not know that GSoC could be used for CAREER BUILDING (R3), ACADEMIC (R4) concerns, earning STIPENDS (R5), getting PEER RECOGNITION (R6), and for INTELLECTUAL STIMULATION (R7) (Figure 5.6c). Similarly, Brazilian non-participants did not know of the existence of some rewards (R6 and R7) (Figure 5.6b). In this way, our theory seemed to be effective at presenting new aspects of GSoC to non-participants.



(a) (b) (c)

Figure 5.6: *Change of non-participants' perception of the rewards importance*

Even when non-participants knew about some rewards, Figure 5.6 shows that after being presented to our findings, their perception of the importance of most rewards increased. We can observe this finding by the increase of the rewards' score after non-participants watched the video. Higher scores for a reward indicate that either a non-participant ranked that reward higher or included it in the ranking. Thus, our theory seemed to deepen the non-participants' understanding of how most participation rewards could be useful to GSoC students.

Finally, Figure 5.6 also shows that the non-participants reprioritized the importance of several rewards. When we observe all participants rankings, we can see that ACADEMIC concerns ranked last despite its increase in the score (Figure 5.6a). Nevertheless, Figure 5.6b and Figure 5.6c suggests that there may be differences in how non-participants from different countries perceive the importance of the rewards that should be further explored. For example, while Brazilian non-participants seem to consider CAREER BUILDING (R3) to be more critical than the STIPENDS (R5), the Chinese non-participants seems to think otherwise. Thus, our theory seems to lead not only to a reprioritization of the importance of the rewards but also in different ways for different populations.

In addition, several non-participants (22) answered that our results influenced their decision to engage via GSoC. However, we noticed that in several cases, non-participants did not feel confident enough in their programming skills to engage via a Summer of Code despite their will to do it. We believe that this finding may indicate 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 alternative ways of matching OSS projects with students with low confidence in their programming skills.

Phase III: By analyzing the non-participants' perceptions, we could observe that the Process Theory II broadened their understanding of Summers of Code and of how such programs could assist them in achieving their goals, especially the ones related to their career and the stipends. In a sense, Process Theory II seemed effective in communicating the experiences of former students to non-participants, which could inspire them to engage via such programs in the future.

5.4 Implications

We consider some implications for researchers, OSS projects, and students.

OSS projects: The engagement theory encapsulates the empirical knowledge of several OSS projects. In this way, OSS projects can use our results to collectively make a well-informed decision about the engagement via Summers of Code. While OSS projects that already engaged via such programs can revise their action plans in light of our results, the projects that never applied can use them as a guide. Additionally, our theory explains the process of motivating students that goes beyond the offering of rewards, especially extrinsic rewards.

Students: The engagement theory condenses the participation rewards that motivated several former students. In this way, our theory can broaden new students' perspectives, not only giving them a better understanding of Summers of Code in general but also communicating the experiences of former students.

5.5 Limitations and Threats to Validity

In this Section, we acknowledge some limitations of our approach and show how we directed our best efforts to handle them.

Internal validity: 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 tasks that increase their odds of acceptance in the program. In this way, underreporting might occlude tasks that are relevant for the OSS projects' decision process of engaging via Summers of Code but usually are not written in application forms. Nevertheless, we believe that even in this context, our theory would still be a useful description of the most significant tasks, which should be enough in several cases.

External validity: Although we collected data from multiple sources for different OSS projects, we likely did not find all tasks OSS projects accomplish or consider all factors that motivate students to contribute. Each OSS project has its singularities, and the tasks for engaging via Summers of Code can differ. With more data, perhaps we could find different ways of categorizing concepts, which in turn could increase explanatory power. We believe that our strategy to consider multiple sources, which includes students, mentors, OSS project applications for GSoC, and Summers of Code administrators (guides), can mitigate this issue, identifying commonalities that usefully describe the tasks necessary for participation.

Moreover, our results are grounded mainly on GSoC. Hence, our theory may not be generalizable to other Summers of Code. Nevertheless, as we mentioned previously, GSoC is the best-known and well-established Summer of Code. In this way, we believe that our theory is representative of most cases. We leave for future research to identify and adapt our theory to new situations.

Other threats: Another threat to the validity of our results is the data classification subjectivity. To alleviate this threat, we employed grounded theory procedures [Cha06], 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' imagination [Lan99, BD18]. Furthermore, we could not contact applicants who were not accepted, which makes our theory biased towards the applicants who were accepted. Future research can devise methods to contact these applicants. Nevertheless, we acknowledge that having both types of applicants would help the theory to explain successful and unsuccessful engagement cases, increasing its explanatory power.

5.6 Final Considerations

Attracting and retaining new contributors are vital to the sustainability of several open source software (OSS) projects that depend mostly on a volunteer workforce. Some OSS projects engage via Summer of Code programs expecting to onboard new contributors, after mentoring them in the projects' contribution norms. Summers of Code are programs that promote OSS development by students (typically) for a few months. Here, by studying the well-established Google Summer of Code (GSoC), we posit the Engagement Process Theory that explains the engagement process in OSS projects via Summer of Code programs. We modeled the engagement process for OSS projects (Mentoring Process Theory) and students (Motivational Process Theory).

Our Engagement Process Theory is grounded in multiple data sources such as participation guides provided by GSoC program administrators [The18], OSS projects' applications for GSoC, surveys involving students and mentors, interviews with students, quantitative studies [SWG+17], and the literature (e.g., [TCH14, TCKH14, TKCH16, SLW12]). Additionally, we present our Motivational Process Theory to students who never engaged via Summer of Code programs to collect insights on how to adapt/conduct/engage via such programs. This study differs from existing research in several ways. We offer an in-depth investigation of the engagement process in OSS projects via Summers of Code, while previous studies target different aspects of such programs and on a smaller scale.

In addition, we believe that the development of our Engagement Process Theory is a first step towards building a *variance theory*, which can explain in greater detail why and when students contribute meaningfully to OSS projects. A variance theory could ultimately predict the students who are more likely to keep contributing to OSS projects after participation.

Chapter 6

Discussion

In this Chapter, we discuss some findings and provide a higher-level discussion. There is evidence that high-quality mentoring is labor-intensive and time-consuming. While offering dedicated mentorship plus designing a high-level Summer of Code project could potentially enrich students' learning experience, it may have the adverse effect of burdening mentors. Thus, an issue that may arise for some OSS projects is how much return on their mentoring investment they can expect in terms of gaining code contributions and new volunteers. Ideally, Summer of Code programs should benefit both OSS projects and students. Students should acquire experience, branding, and learning by joining the projects' workforce, while the projects by providing mentoring should achieve tasks accomplishment during, and possibly after the programs.

Encouragingly, there is evidence that some OSS projects accomplish tasks and goals by participating in such programs: "*GSoC is an important program because it provides a possibility to mentor students intensively over a [relatively] long period of time. The student gets more experience, while the project [gets] tasks done, that [otherwise] would be harder to do [by] pure volunteers.*"[1] (We leave to future research the investigation of students' actual learning and the nature of project tasks.) Additionally, some researchers presented some empirical evidence on task accomplishment. For example, in two different studies, Trainer and colleagues [TCKH14, TCH14] suggested that features developed during GSoC were incorporated into scientific projects' codebase. Moreover, Schilling et al.'s work [SLW12] identified and studied former GSoC students who kept contributing to the KDE project after the program.

Nevertheless, the OSS projects' engagement via Summer of Code programs involves a trade-off between the effort invested in mentoring students and the mentors' ability to simultaneously address the project tasks, personal, and professional affairs. As explained by a mentor: "*I participated in GSoC as a mentor (...) While it didn't 'cost' me anything in dollars, it cost me probably 200 hours of my time (...) This is an enormous cost to me, as an employee (...) A student received payment to write code that was discarded. And I (and several of my colleagues) spent a huge amount of time, which I could have spent on other things (...) Benefit to project, pretty heavily negative.*"[2] Over time, this situation can affect several OSS project members such as the case of the Debian community,

[1] https://wiki.documentfoundation.org/Development/GSoC

[2] https://mail-archives.apache.org/community-dev-mailing-list-archives

who decided not to participate in GSoC 2017: "*Due to the lower amount of general motivation, and most notably the weakness of our projects page during the Google review (...) Debian will not be part of [GSoC] this year. Some of our recurring mentors have shown some signs of 'GSoC fatigue' (...) let's have a summer to ourselves to recover (...) and come back next year.*"[3]

We believe that our theory contributes by offering a theoretical foundation for the OSS projects that need to make well-informed decisions about engaging via Summer of Code programs in at least two ways. First, we structured the OSS projects' engagement process by describing the tasks accomplished before, during, and after such programs (see the Mentoring Process Theory in Figure 5.2). We posit that the process of engagement for OSS projects leads to tasks towards planning their participation, socially integrating the students, rewarding contributions and achievements, and management. That is, although concrete tasks certainly differ for each OSS project, our data suggest that they can be classified into these major categories.

Complementary, for better clarity, it is equally important to discuss what we *do not* claim. For example, we do not claim that *all* OSS projects accomplish *all* these tasks. Nor that any OSS project accomplishes these tasks *all the time*. Moreover, we do not claim that the execution of these tasks leads to students' retention. Furthermore, it is not the case that these tasks need to be accomplished in any particular sequence. In contrast, the Mentoring Process Theory only condenses all the ways the OSS projects in our sample acted when they participated in GSoC, which is expected from a process theory.

Second, we modeled the students' engagement process in terms of how their motivation is influenced when they contribute to their assigned projects (see the Motivational Process Theory in Figure 5.4). According to our theory, students engage in OSS projects not only due to their aspirations but also by a contribution experience that goes beyond acquiring knowledge, developing skills, or socialization, which seems to be the consensus in the literature [DJ15]. This notion is illustrated by Butler et al. [BSKK02], who stressed that volunteers come for the knowledge but stay for the people. It is worth mentioning that we can make observations to the Motivational Process Theory (see Figure 5.4) similar to ones made about the Mentoring Process Theory (see Figure 5.2).

Following Ralph's advice [Ral18], we also consider whether our Engagement Theory has more explanatory power than rival theories. In general, Legitimate Peripheral Participation (LPP) theory is used to explain how newcomers (i.e., apprentices) engage in OSS projects (i.e., communities of practice) [LW91]. When applied to OSS projects, LPP [LW91] explains that newcomers begin their involvement by first observing experienced project members (i.e., masters). After a while, newcomers become in charge of straightforward but valuable tasks (e.g., *opening a PR*). In time, newcomers get familiar with contribution norms and become in charge of more important tasks. This process (i.e., legitimization) should culminate in the emergence of frequent contributors [LW91].

Nevertheless, LPP does not describe precisely the engagement in OSS projects via Summers of Code. Contributing to OSS through Summer of Code programs may significantly alter the engagement process described by LPP. In several cases, students did not start at the margin by first observing experienced members. Instead, they were individually guided—and sponsored—to

[3] https://lists.debian.org/debian-outreach/2017/02/msg00008.html

become contributors. Also, according to LPP, by successfully contributing peripheral tasks, apprentices should be gradually legitimized (i.e., acknowledged as project members) by experienced members. Instead, the student-OSS-project relationship in a Summer of Code context is mediated by a contract, which binds students and mentors for a few months. Thus, Summer of Code students have the time to dedicate themselves to the GSoC project, which provides them with an opportunity to develop strong social ties to mentors. Nevertheless, it is not clear from our data if relationships mediated by contracts could legitimize students. Therefore, our findings indicate that more research is necessary to understand how students can be legitimized as full project members in a Summer-of-Code context.

6.1 Implications

We list some implications of this study for different stakeholders.

6.1.1 OSS Projects

We claim that OSS project members should moderate their expectations about achieving long-term contributors by engaging in Summer of Code programs. Although GSoC increased participation in the assigned projects, the students in our sample did not remain considerably longer after the program (see Chapter 3). In a different study, we showed that several of the mentors in our sample learned to moderate expectations by experience (see Chapter 4). As it can be frustrating, OSS projects resolve this issue in different ways. Sometimes, it seemed to us that some OSS projects adopted an understanding that GSoC should be considered a short-to-medium-term investment in students' experience in exchange for software feature development. As a result, such projects typically hardened their selection process, meaning that they selected students that could contribute to OSS projects meaningfully. In other cases, OSS projects pondered whether participating was worth the costs, to the point of the disengagement in Summers of Code for some time.

We highlight that recruiting volunteers seems to be challenging for any volunteering program, regardless of domain. As GSoC administrators advise [The18], the engagement in Summer of Code programs is about building a positive learning experience for the students. After all, learning is a process that takes time [Fur96]. With our theory, we believe that OSS projects, especially mentors, can be better prepared for pondering whether they can afford some costs of participating in Summer of Code programs to try to bring more people into OSS.

6.1.2 Students

First-time students who want to engage in Summer of Code programs can benefit from our theory in many ways. First, our findings document that applicants are frequently encouraged to get involved with OSS projects before the selection process. In this way, first-time students can showcase their abilities and willingness, which seems to increase the odds of being accepted. Second, we could observe that Summers of Code bring rewards to the participants that go beyond what the typical applicant imagines. Students see such programs as great opportunities to build a portfolio and trigger their career, as can be observed in Table 4.2. Participants from developing countries report that participating in a program like GSoC increases students' visibility when seeking a job in large corporations.

Moreover, some students consider participating in GSoC as a chance of networking, enabling them to interact with OSS contributors and with "top of field people," as shown in Table 4.2. Third, students consider Summer of Code programs an excellent and flexible internship. It enables, for example, students that cannot commute or need to help their families during summer break to participate in internships.

6.1.3 Summers of Code administrators

The organizers must observe and value career advancements, by, for example, easing the access to the participants' list and providing certificates, something similar to what GSoC does. While looking online for the participants' email addresses, we analyzed the students' professional social network profiles and noted that they indeed list the participation at GSoC as job experience. We could observe that a significant part of the students' motives is not related to the stipends (see Table 4.2). Therefore, existing and potential new programs could offer the students a chance to participate without receiving stipends. By doing that, the projects would benefit from more newcomers, and the students would benefit from non-monetary rewards that the program offers. Besides, since one of the motives reported by the students was networking, Summers of Code programs could consider organizing regional meetups, inviting project members, and participants, so they have a chance to meet the regional project members in person. Lastly, an issue that needs reflection from the Summer of Code organizers side is that, as participants come from all over the world, and a significant part of them are not from the US (see statistics for 2017),[4] organizing the program in different periods, or making the calendar more flexible, would benefit students from different countries. It would help, for example, those countries in which the three-month break is from December to February.

6.1.4 Universities

Universities can also benefit from our results. Although Google does not classify GSoC as an internship,[5] *we evidenced that some universities use students' participation in the program for validating course credits*. Thus, provided that Summer of Code programs meet the universities' criteria and regulations, universities could use our results to provide incentives and support students to get into GSoC as a way to both help the students and contribute to OSS. The students would get coding experience in a real setting, and they would be exposed to real challenges. The experience of a GSoC student could potentially enrich the experience of other students. Besides, validating course credits would be especially interesting for universities away from major cities, in which the internship possibilities do not offer technical challenges to enable students to put what they learned in practice.

[4] https://developers.google.com/open-source/gsoc/resources/stats#2017

[5] https://developers.google.com/open-source/gsoc/faq

6.1.5 Researchers

As Ralph explained [Ral18], process theories offer a better foundation for the development of engagement methods. While process theories are concerned with the *actual* ways entities (i.e., OSS project or student motivation) change, methods "*prescribe specific practices, techniques, tools, or sequences that are ostensibly better than their alternatives*" [Ral18, p. 20]. As mentioned previously, OSS projects can differ significantly, such as concerning the number of members, their attractiveness, and their financial support possibilities. In this way, researchers could extend our results by studying efficient ways of engagement, perhaps taking into account OSS projects' peculiarities.

Mentors: We observed only students' motivation. However, to the best of our knowledge, mentors' motivation remains understudied. Understanding what drives mentors to support newcomers could benefit OSS projects and newcomers. Furthermore, it would be interesting to create an array of strategies that mentors could use to deal with common problems such as candidates' selection, project creation, mentoring guidelines, and others.

Demographics: Additionally, researchers could study students' demographics and how (or whether) potential differences influence students' motivation and contribution. For example, our data shows that there are more students from Asia (mainly from India) than students from South America or Europe. Also, additional research is necessary to understand how companies see the participation in Summers of Code in their hiring processes.

Another way researchers could extend our results is to study which OSS projects are selected by Summer of Code administrators. OSS projects that are rejected by Summer of Code administrators may have difficulties in finding newcomers as one of the benefits of participating in such programs is higher exposure. In a sense, the criteria Summer of Code administrators adopt to select OSS projects may determine which projects get to survive or evolve. This bias may impact small OSS projects (1-2 volunteers) as they may not be able to afford everything that is required to participate in Summer of Code program.

6.2 Set of guidelines

Based on OSS projects' applications for GSoC, interviews with students, and surveys with students and mentors, we drafted a set of guidelines useful to OSS projects that decide to engage in Summer of Code programs. We did not evaluate the guidelines; however, we consider it a significant outcome for both OSS projects willing to engage in such programs.

6.2.1 Guidelines on applicants' and mentors' selection

Set technical level expectations: Make clear to applicants what is the students' expected technical level for successful completion of the program. Some OSS projects require applicants to accomplish easy programming tasks, while others require them a contribution that follows OSS projet's norms. As we showed in our analysis of the theory's usefulness (in Section 5.3.3), there is a pool of students willing to contribute that do not know how (or whether) their practical knowledge can serve an OSS project.

Require a detailed schedule: Ask applicants to provide a clear and well-detailed schedule of how they plan to solve the proposed task. Having a detailed schedule can help applicants to organize themselves and to achieve their short/medium-term goals.

Select appropriate mentors: Prioritize mentors with proven involvement in the OSS project. Before selection, have mentors spend enough time with applicants for a general idea of how they work together. We suggest the reading of Shilling et al. [SLW12] for strategies on the students-team fit. Furthermore, make sure that mentors have excellent social skills.

Pair mentors: Designate two mentors per student to avoid problems of disappearing mentors. Consider holidays and vacations before allocation. Moreover, pair less experienced mentors with trusted and experienced ones. It provides students with a pool of mentors ready to jump in temporarily or permanently if needed.

Workload alert: When doing the call for mentors, make sure that volunteer mentors are aware of what is expected from them and that they are willing to work hard on their mentoring task.

6.2.2 Guidelines on mentor-student engagement

Encourage communication: Encourage mentors to have frequent one-on-one meetings with their students. Make it clear to mentors that having excellent communication with the students and coordinators is mandatory for a successful mentoring experience.

Establish a dedicated communication channel for mentors: Although mentors supposedly possess the technical knowledge to contribute, they may not be experienced at mentoring. In such cases, they may need specific guidance from experienced mentors. A dedicated communication channel may help conversations flow freely.

Balance mentor's workload: Mentoring is a time-consuming and labor-intensive endeavor. Whenever possible, OSS projects' administrators should balance the mentor's workload with other project members.

Offer a scale-up policy: Make students aware that any point in time an OSS project administrator is available to deal with any mentor related problems they have.

6.2.3 Guidelines on students' progress

Identify menaces to completion: Check with the students whether they have a suitable place to work (e.g., with internet access). Do students have exams? Do they have other activities during the Summer of Code? Require a physical address and a phone number and confirm the phone number during the selection.

Have a status update policy: Ask students to report their progress frequently. Following this suggestion may help to spot difficulties before students are lost. Lengthy reports are not necessary if the patch clearly shows that work is being done. Use email for quick updates and blog posts for more detailed updates.

Monitor students progress: Monitor students' progress through regular code submissions or blog posts.

Interfere when students stall: When the students are not making progress, encourage progress by requesting to commit code early and often, or file issues with collecting feedback on what code to write.

Allow minor deviations from original plan: Do not expect everything to go according to a plan. The important thing is that the students show steady progress over the program.

Encourage reflection on progress: Have students spend some time at the end of each working day to answer reflective questions such as: " *What was your plan for today? Did you accomplish that? Why? What is your plan for tomorrow?"*

Divide students' work into manageable chunks: Have mentors divide students' work into manageable chunks with measurable milestones. This action helps students learn how to manage and break up tasks on large scale projects.

6.2.4 Guidelines on students' involvement during the Summer of Code

Embrace the students warmly: Embrace warmly the students who contribute to OSS projects, in any capacity. This gesture makes them feel part of the group from day one.

Notify members of students: Remember to notify other members of the presence of students. In this way, the other members can know they should devote more time to answering and clarifying technical details.

Use well-established communication channels: Prioritize public interactions with students. Making students use the same communication channels' as the other OSS project members contribute to making them feel they belong to the community. By encouraging students to communicate with the OSS project directly, instead of privately with their mentors, you will infuse them with the OSS process development.

Establish a dropout policy: Make sure the OSS project has a clear dropout policy. For example, in case of students stop communicating, try to contact them by any means. After an *agreed period* (e.g., one week), send a final warning. In case students do not respond within an *agreed period* (e.g., one day), fail the student.

Guidelines on students' involvement after the Summer of Code

Offer travel support: Encourage students to go to conferences, which are an opportunity to get in touch with companies interested in hiring, get to know all the people they worked with during the program, participate in the design sessions.

Ask for less demanding tasks: After Summers of Code, students are typically busy with school. Asking for less demanding tasks may help students keep involved.

Offer membership: For students with proven good knowledge of codebase, coding standards, and shows activity after the program, grant them commit access on the OSS projects' repository.

Mention students on team's page: Acknowledge students by mentioning them on the OSS project's web page.

Showcase success cases: Show the students testimonials of team members who started as Summer of Code students. Show how they joined the team and what made them decide to be active in that OSS project in particular and OSS projects in general.

Chapter 7

Conclusions

Engaging new contributors is vital to the sustainability of the Open Source Software (OSS) projects that depend mostly on a volunteer workforce [HRSF02]. Some OSS projects join Summer of Code programs in the hope of engaging new contributors or receiving meaningful contributions, after mentoring them in the projects' contribution norms. Nonetheless, the literature does not offer substantial evidence on what this engagement process entails for OSS projects and students. In this thesis, we identified student motivations, assessed student contributions quantitatively, structured, and discussed the engagement process in OSS projects via Summer of Code programs.

In Phase I of this thesis, our goal was to identify student motivations to engage in OSS via Summer of Code programs. We conducted two distinct studies to understand students' motivations. In the first study, we investigated several factors of students' contributions, such as the number of commits, code churn, and contribution date intervals. We found that although most OSS projects incorporated (i.e., merged) some code produced during Google Summer of Code (GSoC), the vast majority of newcomers stopped to contribute in a few months after the program. In the second study, we combined a survey (of students and mentors) and interviews (of students) to understand what motivates students to engage via GSoC. Combining the findings of both studies, our data suggest that students engaged in OSS via Summer of Code programs for work experience, and not necessarily to become frequent contributors. The main result of this phase was the empirical identification of the students' motivations to engage in OSS via Summer of Code programs.

In Phase II, our goal was to explain the process of engagement in OSS via Summer of Code programs for OSS projects and students. For explaining the engagement process of OSS projects, we used application forms for GSoC. In the case of students, we revisited the data from Phase I. Moreover, we used guides provided by program administrators and the literature. We employed grounded theory procedures to analyze all qualitative data and built a theory [Cha06]. Our findings reveal that, when OSS projects participate in such programs, their tasks converge towards planning their participation, socially integrating the students, rewarding contributions and achievements, and management. Additionally, the interplay of OSS projects' engagement via Summers of Code and students' contribution produces outcomes, some of which either increase students' intrinsic motivation or are instrumental for achieving their goals. With the theory that explains the engagement process for OSS projects and students, we achieve the main contribution of this phase.

We also claim that this thesis contains secondary contributions:

The method used to identify motivation factors can be reused in other studies, especially in other Summer of Code programs other than GSoC. We claim it is a minor and valuable contribution because although motivation factors have been extensively studied in the OSS context, we could still find new factors such as *taking advantage of currency conversion*; *obtaining course credits*, and; *lowering OSS projects' entry barriers.*

The set of guidelines is also a secondary contribution that can be used in practice by the OSS projects that want to provide a better contribution experience to students, and; new students that want to have the bigger picture of what motivated Summer of Code former students.

7.1 Future work

A variance theory of engagement in OSS projects via Summer of Code programs is a possible extension for this study. In this case, by the analysis of additional factors (i.e., independent variables), significant correlations could be found between the engagement level (i.e., dependent variable).

The study of the engagement of women in OSS projects via GSoC could reveal different motivation factors or other prioritization, which could help to bring more women into OSS development, bridging the gap of gender equality.

Designing a general method of the engagement in OSS projects via Summer of Code programs is also a possible extension of this thesis. While process theories only offer explanations, methods, in this context, are concerned with finding a specific set of tasks that increases the odds of achieving a particular goal such as placing meaningful contributions or retention.

The in-depth study of the entities in the engagement process could also be helpful in improving students' contribution and retention. For example, OSS projects are heterogeneous, differing concerning lots of factors including size, complexity, attractiveness, financial resources, and members. We believe that the engagement method of a small and unknown OSS project may be different from the method of a big known OSS project.

Appendix A

Summers of Code Programs

In the following, we present Summer of Code programs (A.1) and Semester of Code programs (A.2).

A.1 Summer of Code Programs

In the following sections, we present internships that aim, among other goals, to introduce students into OSS development.

A.1.1 Free Software Foundation Internship

Description: Interns work closely with Free Software Foundation staff members in the area of interest, such as campaign and community organizing, free software licensing, systems and network administration, GNU Project support, or web development.

Audience: All applicants must be: open-minded and tolerant of others, able to work as part of a team, and interested in the ethical ramifications of computing.

Stipend: No

Website: http://www.fsf.org/volunteer/internships

Status: Active

A.1.2 Google Code-In

Description: Pre-university students ages 13 to 17 are invited to take part in a global online contest that aims at introducing teenagers to the world of open source. With a wide variety of bite-sized tasks, it is easy for beginners to jump in and get started, no matter what skills they have. Mentors from our participating organizations lend a helping hand as participants learn what it is like to work on an OSS project.

Audience: Pre-university students ages 13 to 17 are invited to take part

Stipend: Prizes for winners

Website: https://developers.google.com/open-source/gci/

Status: Active

A.1.3 Google Summer of Code

Description: Google Summer of Code (GSoC) is a comprehensive Google program that offers students a stipend to write code for OSS for three months. GSoC started in 2005, and Google pays the students who complete all three phases of the program. GSoC has five goals.

(*i*) “Create and release OSS code for the benefit of all."

(*ii*) “Inspire young developers to begin participating in OSS development"

(*iii*) “Help OSS projects identify and bring in new developers and committers."

(*iv*) “Provide students the opportunity to do work related to their academic pursuits (flip bits, not burgers)"

(*v*) “Give students more exposure to real-world software development scenarios."

For participating in the selection process, applicants must write and submit up to five project proposals to the OSS organizations (previously approved by Google) they wish to work for, such as the Apache Software Foundation and Debian. The organizations' mentors—who are regular contributors—rank and decide which proposals to accept. Next, Google announces accepted student proposals on the GSoC's site. After that, there is a bonding period, in which students get to know their mentors, read documentation, and do some environment setup. When students effectively begin coding for their GSoC projects, Google issues the students an initial payment. After the first half of the program, the mentors assess their students' work and submit a mid-term evaluation to Google. For the passing students, Google issues their mid-term payments. In the second half of the program, the students are advised not only to conclude their coding but also to improve documentation and write tests. In the end, the mentors submit their final evaluations, and the students are required to submit their code to Google. Only passing students receive the rest of their payment. They also get invited to the summit in California.

Audience: Students

Stipend: Yes

Website: https://summerofcode.withgoogle.com/

Status: Active

A.1.4 Julia Summer of Code

Description: Julia Summer of Code (JSoC) is a Summer of Code program that aims at finding contributors to the Julia Language. JSoC started in 2013. In the editions 2014, 2016, and 2017, JSoC was part of GSoC.

Audience: Anyone that can apply for GSoC

Stipend: Yes

Website: https://julialang.org/soc/ideas-page.html

Status: Active

A.1.5 Outreachy

Description: Outreachy helps people from groups underrepresented in free and open source software get involved. We provide a supportive community for beginning to contribute any time throughout the year and offer focused internship opportunities twice a year with several free software organizations.

Audience: Currently, internships are open internationally to women (cis and trans), trans men, and genderqueer people. Additionally, they are open to residents and nationals of the United States of any gender who are Black/African American, Hispanic/Latin@, American Indian, Alaska Native, Native Hawaiian, or Pacific Islander. They are planning to expand the program to more participants from underrepresented backgrounds in the future.

Stipend: $5,500

Website: https://www.gnome.org/outreachy/

Status: Active

A.1.6 Rails Girls Summer of Code

Description: With their fellowship program, they aim to foster diversity in Open Source since 2013. Selected teams receive a three-month scholarship to work on Open Source projects of their choice.

Audience: All people with non-binary gender identities or who identify as women are welcome to apply.

Stipend: The scholarship is based on where the students live, how much their set expenses are, and any exceptional circumstances.

Website: http://railsgirlssummerofcode.org/

Status: Active

A.1.7 Season of KDE

Description: Season of KDE is a community outreach program, much like Google Summer of Code, that has been hosted by the KDE community for seven years.

Audience: Everyone can apply for the Season of KDE. They give preference to those who have applied for Google Summer of Code and to students, but consider applications from anybody interested in contributing to KDE.

Stipend: No

Website: https://season.kde.org/

Status: Active

A.1.8 The X.Org Endless Vacation of Code

Description: The basic terms and conditions of this program are quite similar to Google's GSoC. The key differences are that: An EVoC mentorship can be initiated at any time during the calendar year, and The Board can fund as many of these mentorships as it sees fit.

Audience: Everyone. They will also consider a broader range of proposals than GSoC: technical documentation is a specific area of interest for them.

Stipend: $5,000 for students over 18.

Website: https://www.x.org/wiki/XorgEVoC/

Status: Active

A.1.9 Tor Summer of Privacy

Description: The Tor Project, in collaboration with The Electronic Frontier Foundation, has taken part in Google Summer of Code for 2007 through 2014, mentoring a total of 53 students. This year the program was trimmed back, and the room was needed for new organizations. So they decided to launch Tor Summer of Privacy!, which is a pilot program we hope will grow and guarantee support for students who want to collaborate with privacy tools.

Audience: Anyone that qualifies for Google Summer of Code. They invite and welcome many kinds of students from many kinds of backgrounds to apply.

Stipend: $5,500.

Website: https://trac.torproject.org/projects/tor/wiki/org/TorSoP

Status: Innactive

A.2 Semester of Code Programs

Service-learning programs are designed to balance students' learning goals to the contribution to OSS communities. In the following, we present four programs that explicitly included as goal students' learning and contribution to OSS communities.

A.2.1 Facebook Open Academy

Description: At the beginning of the semester, students and mentors from open source projects come together for a weekend of learning and hacking. After this kickoff session, students return to their universities and continue to work in virtual teams. Open source mentors support their teams by helping students find and understand tasks and review code contributions. The course instructors at each university meet with student teams at regular intervals to review progress. Some instructors overlay a lecture series to provide further learning opportunities to students.

Audience: Students enrolled in accredited universities.

Stipend: No

Website: https://www.facebook.com/OpenAcademyProgram/

Status: Innactive

A.2.2 Humanitarian FOSS Summer Internship

Description: The Humanitarian FOSS (HFOSS) Project offers summer internships for undergraduate computing students as part of the annual HFOSS Summer Institute. The program is open to students who are currently enrolled in American (US) academic institutions and who want to get involved in building free and open source software for use by humanitarian organizations. Institute participants will learn the fundamentals of FOSS development by contributing to open-source projects, and will typically work in teams of 2-3 with mentoring by computer science faculty and advisers from the open source community.

Audience: HFOSS internships are open to US citizens, permanent residents, or international students who have work authorization (OPT). Applicants must be enrolled in an undergraduate degree program in the US.

Stipend: No

Website: http://www.hfoss.org/index.php/undergraduate-internships

Status: Innactive

A.2.3 Mozilla Winter of Security

Description: The Winter of Security (MWOS) is a program organized by Mozilla's Security teams to involve students with Security projects. Students who have to perform a semester project as part of their university curriculum can apply to one of the MWOS projects. Projects are guided by a Mozilla Adviser, and a University Professor. Students are graded by their University, based on success criteria identified at the beginning of the project. Mozilla Advisers allocate up to 2 hours each week to their students, typically on video-conference, to discuss progress and roadblocks

Projects are focused on building security tools, and students are expected to write code which must be released as Open Source. Universities are free to specify their requirements for projects, such as written reports. Mozilla does not influence the way grades are allocated, but advisers will provide any information professors need in order to grade their students.

Audience: Students who have to perform a semester project as part of their university curriculum.

Stipend: No

Website: https://wiki.mozilla.org/Security/Automation/Winter_Of_Security_2016

Status: Innactive

A.2.4 Undergraduate Capstone Open Source Projects

Description: Undergraduate Capstone Open Source Projects (UCOSP) is a program that brings together students from across Canada to work together on open source projects. Students learn how open source software development takes place, practice distributed development, and have the opportunity to work on a project with knowledgeable industry or faculty mentors on software with real users.

Audience: Canadian students.

Stipend: No

Website: http://ucosp.ca/

Status: Active

Appendix B

Instruments

B.1 Survey on Students' Motivation

B.1.1 Survey description

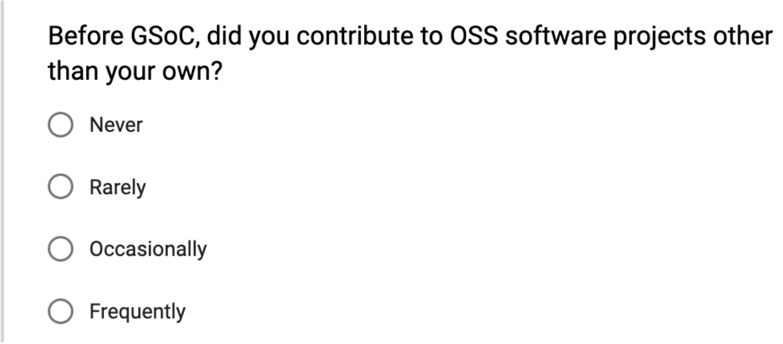
The following exerpt contains the survey description we sent to students and mentors.

We are researchers at the University of São Paulo, Brazil, investigating newcomers engagement into open source software projects. This survey is composed of 4 pages and should take approximately 7-10 minutes to complete. Your participation is completely voluntary and will be very important for this research. All information collected is strictly confidential, and the researchers on this project are the only ones who will have access to this data. Our research is independent from Google. No information collected on the identity of participants will be made available for individuals outside of this research project. For more information about our project: http://bit.ly/2JkIJ19

B.1.2 Student perspective questionnaire

Here we present the questionnaire we sent to students.

Question 01



Before GSoC, did you contribute to OSS software projects other than your own?

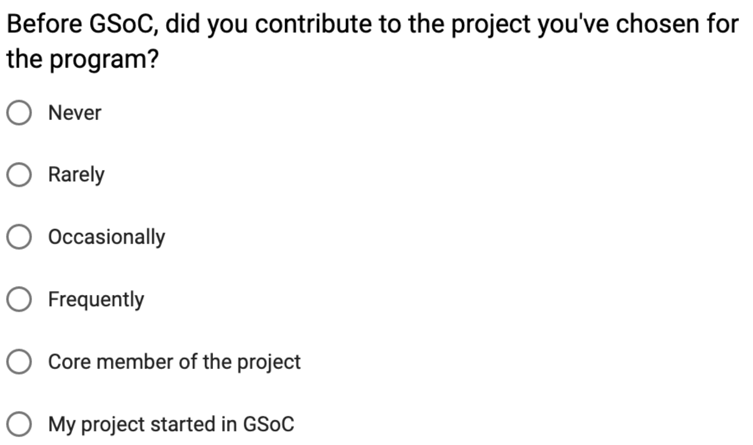
○ Never

○ Rarely

○ Occasionally

○ Frequently

Question 02



Before GSoC, did you contribute to the project you've chosen for the program?

○ Never

○ Rarely

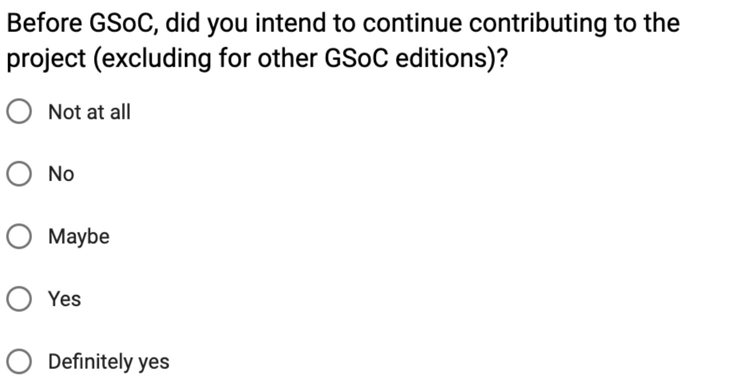
○ Occasionally

○ Frequently

○ Core member of the project

○ My project started in GSoC

Question 03



Before GSoC, did you intend to continue contributing to the project (excluding for other GSoC editions)?

○ Not at all

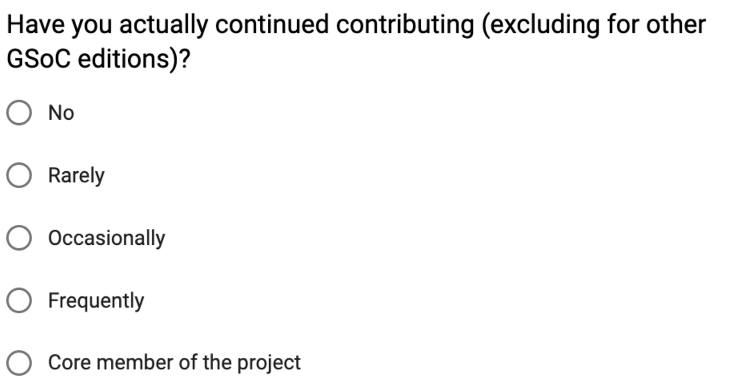
○ No

○ Maybe

○ Yes

○ Definitely yes

Question 04



Have you actually continued contributing (excluding for other GSoC editions)?

○ No

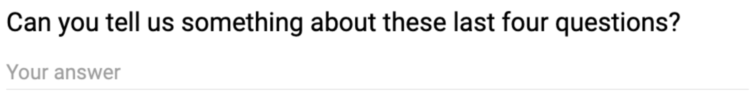
○ Rarely

○ Occasionally

○ Frequently

○ Core member of the project

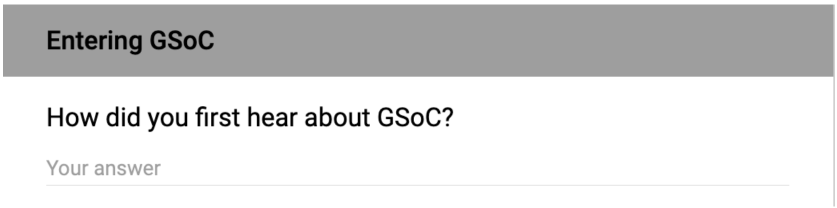
Question 05



Can you tell us something about these last four questions?

Your answer

Question 06

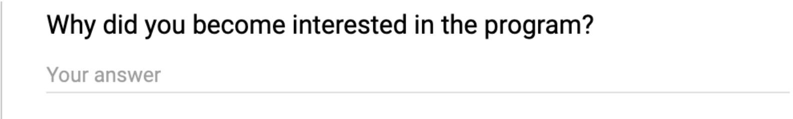


Entering GSoC

How did you first hear about GSoC?

Your answer

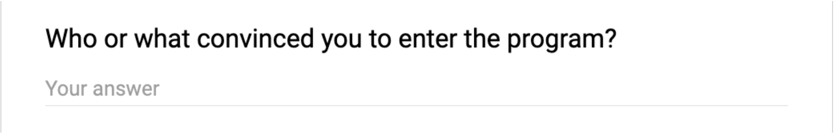
Question 07



Why did you become interested in the program?

Your answer

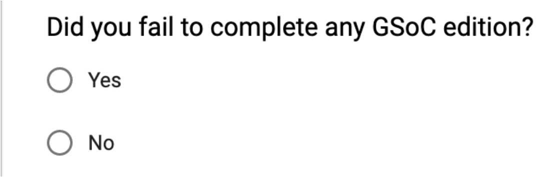
Question 08



Who or what convinced you to enter the program?

Your answer

Question 09

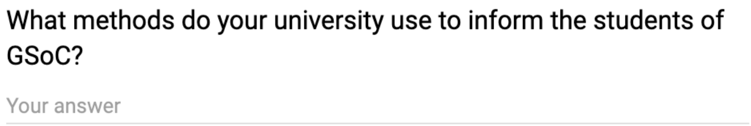


Did you fail to complete any GSoC edition?

○ Yes

○ No

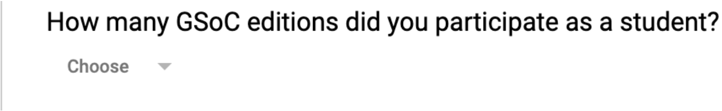
Question 10



What methods do your university use to inform the students of GSoC?

Your answer

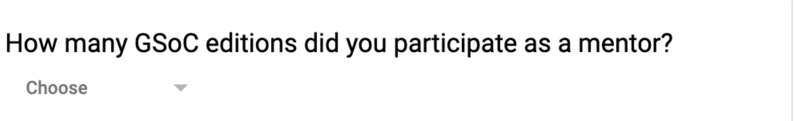
Question 11



How many GSoC editions did you participate as a student?

Choose ▾

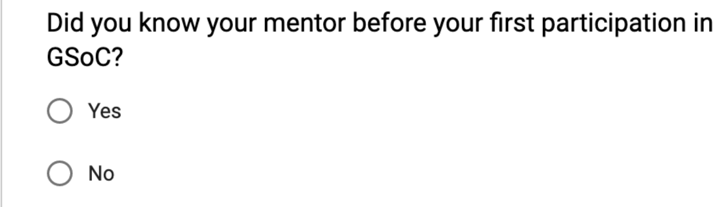
Question 12



How many GSoC editions did you participate as a mentor?

Choose ▾

Question 13

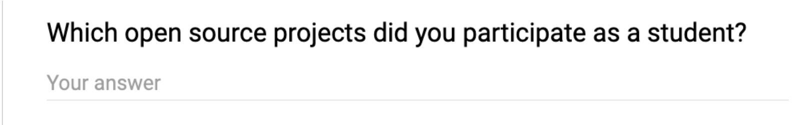


Did you know your mentor before your first participation in GSoC?

○ Yes

○ No

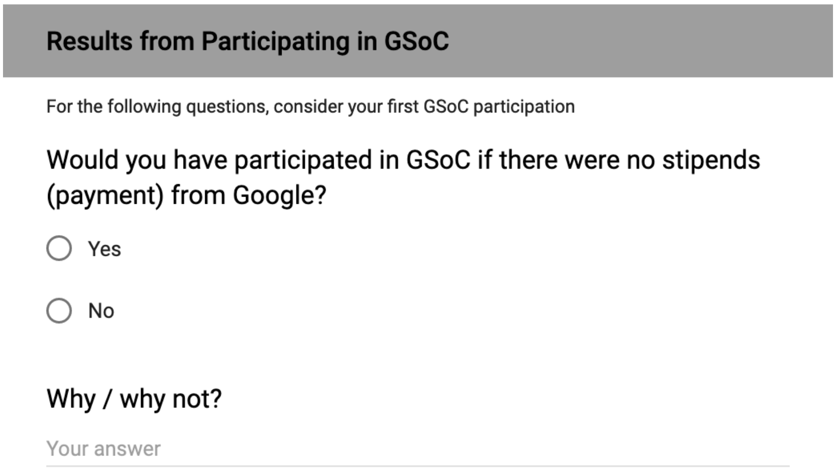
Question 14



Which open source projects did you participate as a student?

Your answer

Question 15



Results from Participating in GSoC

For the following questions, consider your first GSoC participation

Would you have participated in GSoC if there were no stipends (payment) from Google?

○ Yes

○ No

Why / why not?

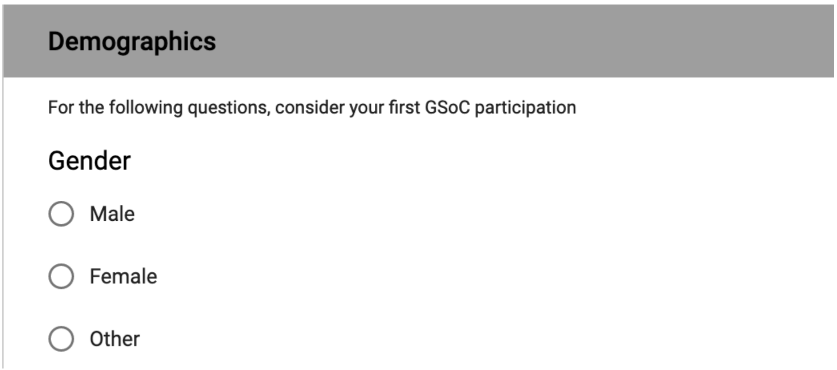
Your answer

Question 16

About the time you decided to enter GSoC, how much do you agree with the following statements. I would have given up entering GSoC if...

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
... I could not continue to contribute to open source software	○	○	○	○	○
... I would not gain respect from my peers at work, university, community etc	○	○	○	○	○
... there were no stipends (payment) from Google.	○	○	○	○	○
... there was no technically challenging work involved	○	○	○	○	○
... I could not announce this experience on my résumé	○	○	○	○	○

Question 17



Demographics

For the following questions, consider your first GSoC participation

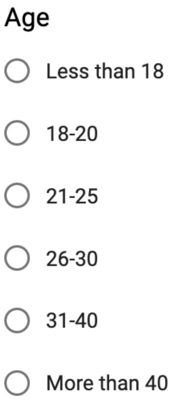
Gender

○ Male

○ Female

○ Other

Question 18



Age

○ Less than 18

○ 18-20

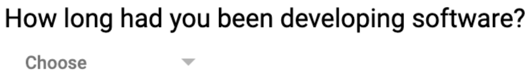
○ 21-25

○ 26-30

○ 31-40

○ More than 40

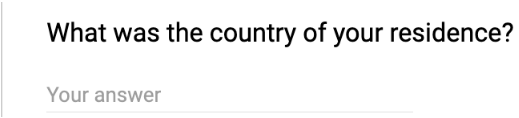
Question 19



How long had you been developing software?

Choose ▾

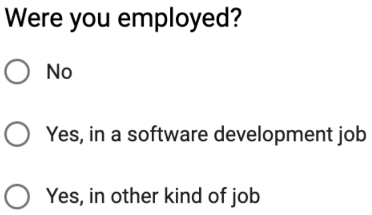
Question 20



What was the country of your residence?

Your answer

Question 21



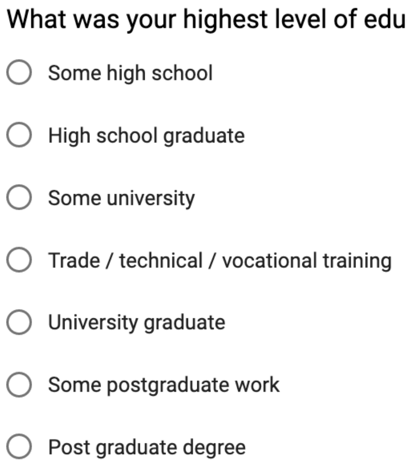
Were you employed?

○ No

○ Yes, in a software development job

○ Yes, in other kind of job

Question 22



What was your highest level of education?

○ Some high school

○ High school graduate

○ Some university

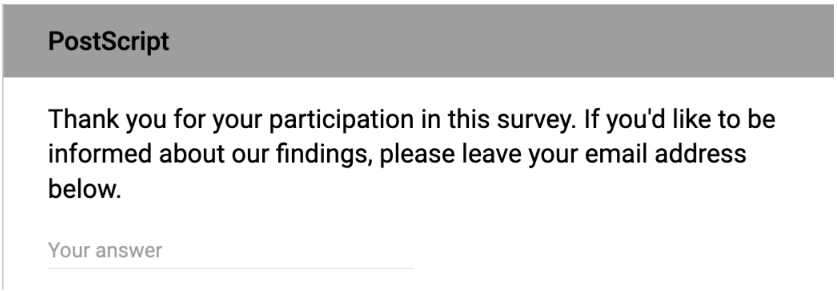
○ Trade / technical / vocational training

○ University graduate

○ Some postgraduate work

○ Post graduate degree

Question 23

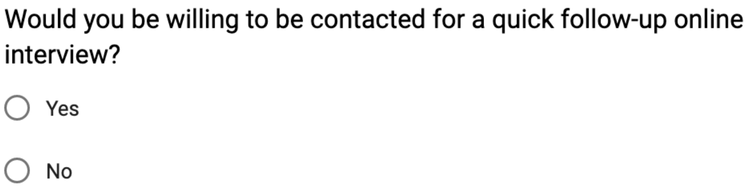


PostScript

Thank you for your participation in this survey. If you'd like to be informed about our findings, please leave your email address below.

Your answer

Question 24

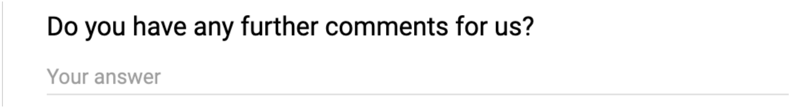


Would you be willing to be contacted for a quick follow-up online interview?

○ Yes

○ No

Question 25



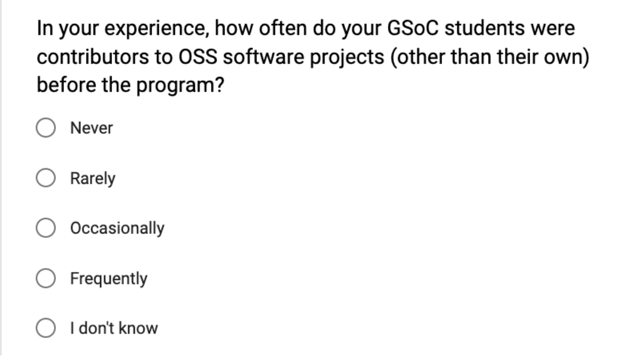
Do you have any further comments for us?

Your answer

B.1.3 Mentor perspective questionnaire

In this section, we present the questionnaire we sent to mentors.

Question 01



In your experience, how often do your GSoC students were contributors to OSS software projects (other than their own) before the program?

○ Never

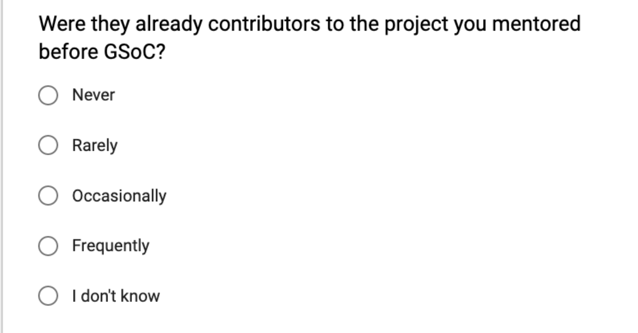
○ Rarely

○ Occasionally

○ Frequently

○ I don't know

Question 02



Were they already contributors to the project you mentored before GSoC?

○ Never

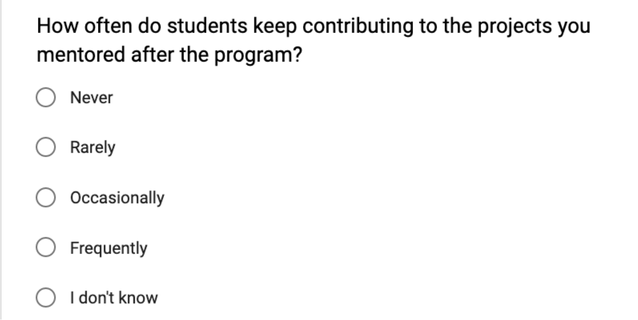
○ Rarely

○ Occasionally

○ Frequently

○ I don't know

Question 03



How often do students keep contributing to the projects you mentored after the program?

○ Never

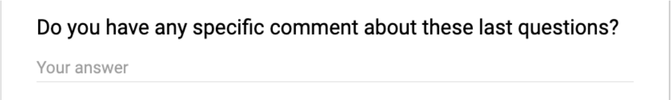
○ Rarely

○ Occasionally

○ Frequently

○ I don't know

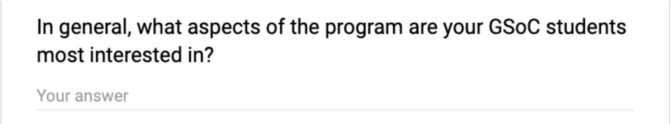
Question 04



Do you have any specific comment about these last questions?

Your answer

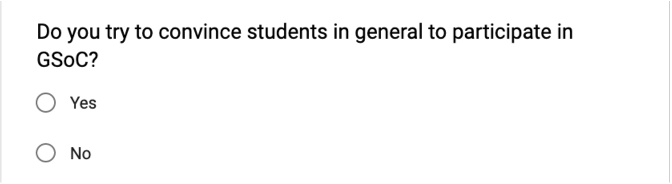
Question 05



In general, what aspects of the program are your GSoC students most interested in?

Your answer

Question 06

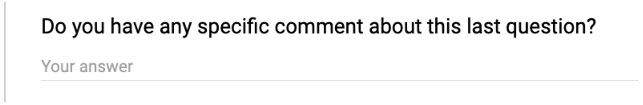


Do you try to convince students in general to participate in GSoC?

○ Yes

○ No

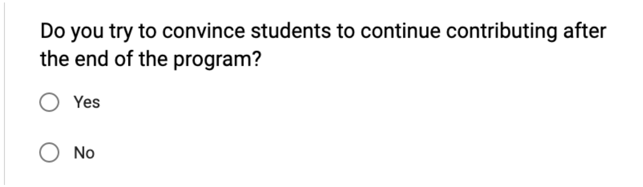
Question 07



Do you have any specific comment about this last question?

Your answer

Question 08

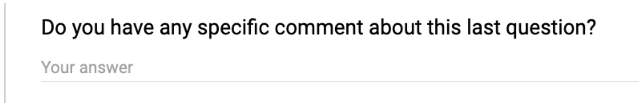


Do you try to convince students to continue contributing after the end of the program?

○ Yes

○ No

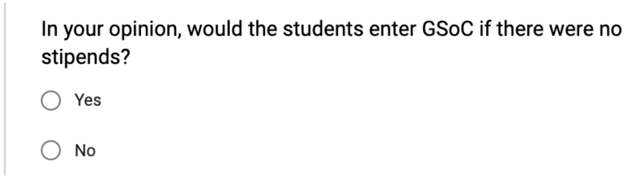
Question 09



Do you have any specific comment about this last question?

Your answer

Question 10

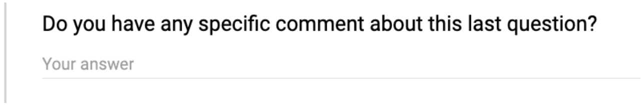


In your opinion, would the students enter GSoC if there were no stipends?

○ Yes

○ No

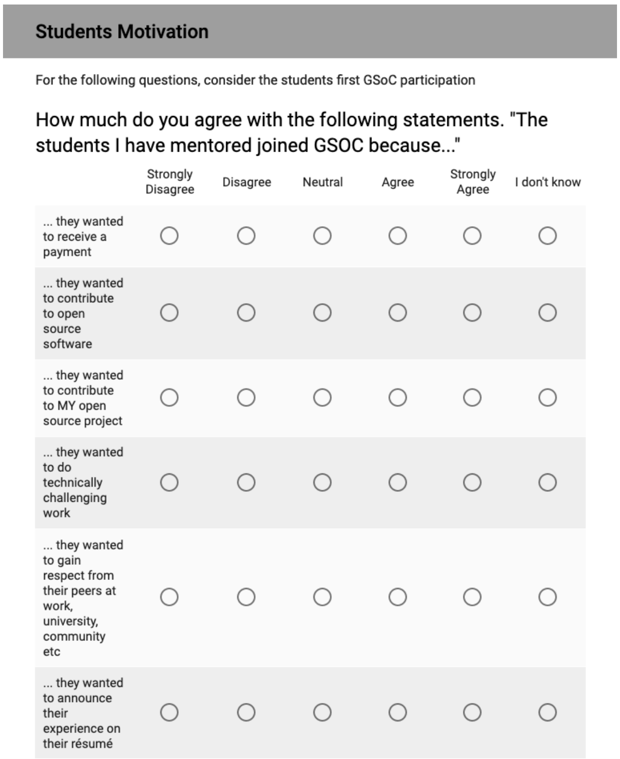
Question 11



Do you have any specific comment about this last question?

Your answer

Question 12



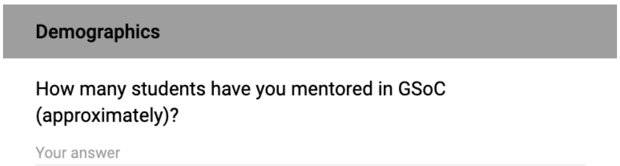
Students Motivation

For the following questions, consider the students first GSoC participation

How much do you agree with the following statements. "The students I have mentored joined GSOC because..."

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree	I don't know
... they wanted to receive a payment	○	○	○	○	○	○
... they wanted to contribute to open source software	○	○	○	○	○	○
... they wanted to contribute to MY open source project	○	○	○	○	○	○
... they wanted to do technically challenging work	○	○	○	○	○	○
... they wanted to gain respect from their peers at work, university, community etc	○	○	○	○	○	○
... they wanted to announce their experience on their résumé	○	○	○	○	○	○

Question 13

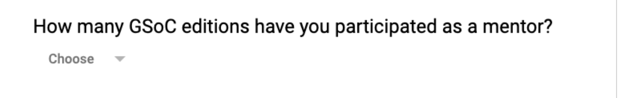


Demographics

How many students have you mentored in GSoC (approximately)?

Your answer

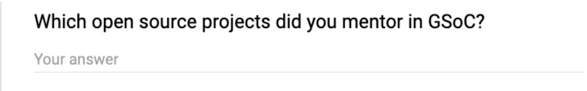
Question 14



How many GSoC editions have you participated as a mentor?

Choose ▾

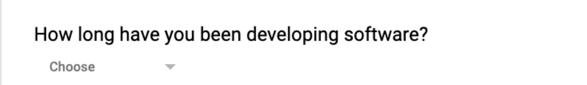
Question 15



Which open source projects did you mentor in GSoC?

Your answer

Question 16



How long have you been developing software?

Choose ▾

Question 17

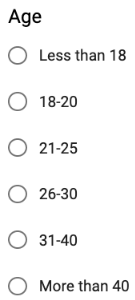
Gender

○ Male

○ Female

○ Other

Question 18



Age

○ Less than 18

○ 18-20

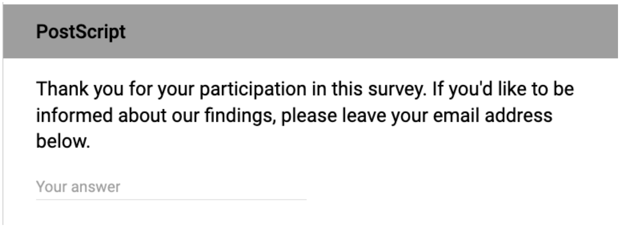
○ 21-25

○ 26-30

○ 31-40

○ More than 40

Question 19

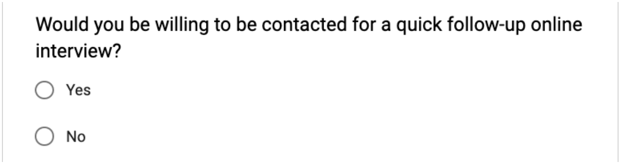


PostScript

Thank you for your participation in this survey. If you'd like to be informed about our findings, please leave your email address below.

Your answer

Question 20

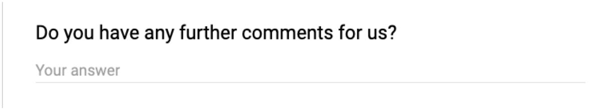


Would you be willing to be contacted for a quick follow-up online interview?

○ Yes

○ No

Question 21



Do you have any further comments for us?

Your answer

Bibliography

[BB10] Hind Benbya e Nassim Belbaly. Understanding Developers' Motives in Open Source Projects: A Multi-Theoretical Framework. *Communications of the Association for Information Systems*, 27(October):589–610, 2010. 14

[BBH$^+$08] Sarah Beecham, Nathan Baddoo, Tracy Hall, Hugh Robinson e Helen Sharp. Motivation in Software Engineering: A systematic literature review. *Information and Software Technology*, 50(9-10):860–878, 2008. 14, 40

[BD18] Sebastian Baltes e Stephan Diehl. Towards a Theory of Software Development Expertise. Em *Proceedings of the 26th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE '18)*, página 14, Lake Buena Vista, FL, 2018. ACM. 66, 71

[Ben01] Yochai Benkler. Coase's Penguin, or, Linux and the Nature of the Firm. *The Yale Law Journal*, 112(3), 2001. 15

[BL01] M. Bergquist e J. Ljungberg. The Power of Gifts: Organizing Social Relationships in Open Source Communities. *Information Systems Journal*, 11(4):305–320, 2001. 16

[Bla64] P. Blau. *Exchange and Power in Social Life.* Wiley, New York, 1964. 14

[BSKK02] Brian Butler, Lee Sproull, Sara Kiesler e Robert E Kraut. Community Effort in Online Groups: Who Does the Work and Why? *Leadership at a Distance*, páginas 1–32, 2002. 74

[BSS07] J. Bitzer, W. Schrettl e P. J. Schröder. Intrinsic Motivation in Open Source Software Development. *Journal of Comparative Economic*, 35(1):160–169, 2007. 15

[CAKL10] Boreum Choi, Kira Alexander, Robert E Kraut e John M. Levine. Socialization tactics in Wikipedia and their effects. Em *CSCW '10 Proceedings of the 2010 ACM conference on Computer Supported Cooperative Work*, páginas 107–116, Savannah, Georgia, 2010. 10

[CCC16] Johanne Charbonneau, Marie Soleil Cloutier e Élianne Carrier. Why Do Blood Donors Lapse or Reduce Their Donation's Frequency? *Transfusion Medicine Reviews*, 30(1):1–5, 2016. 10

[CF09] Jorge Colazo e Yulin Fang. Impact of License Choice on Open Source Software Development Activity. *Journal of the American Society for Information Science and Technology*, 60(5):997–1011, 2009. 30

[CFM08] E. Capra, C. Francalanci e F. Merlo. An empirical study on the relationship between software design quality, development effort and governance in open source projects. *IEEE Transactions on Software Engineering*, 34(6):765–782, 2008. 5

[Cha06] Kathy Charmaz. *Constructing Grounded Theory*. SAGE Publications, London, UK, 2006. 37, 58, 61, 71, 81

[CIA13] Laura Christopherson, Ray Idaszak e Stan Ahalt. Developing Scientific Software through the Open Community Engagement Process. Em *First Workshop on Sustainable Software Science: Practice and Experiences*, Denver, CO, 2013. 32

[Cif03] Andrea Ciffolilli. Phantom authority, Self-selective recruitment and retention of members in virtual communities: The case of wikipedia. *First Monday*, 8(12), 2003. 11

[Cre12] John W Creswell. *Educational Research: planning, conducting, and evaluating quantitative and qualitative research*. Pearson, 4th ed edição, 2012. 37

[CSK99] D Constant, L Sproull e S Kiesler. The Kindness of Strangers: The Usefulness of Electronic Weak Ties for Technical Advice. *Organization Science*, 7(2):119–135, 1999. 11

[CSSD10] Indushobha Chengalur-Smith, Anna Sidorova e Sherae Daniel. Sustainability of Free/Libre Open Source Projects: A Longitudinal Study. *Journal of the Association for Information Systems*, 11(11), 2010. 16

[Dan15] Pocock Daniel. Want to be selected for Google Summer of Code 2016?, 2015. 26

[DAW03] Paul a David, Seema Arora e Andrew Waterman. FLOSS-US: The Free/Libre Open Source Software Survey for 2003, 2003. 15, 16

[DJ15] Sylvain Dejean e Nicolas Jullien. Big from the beginning: Assessing online contributors' behavior by their first contribution. *Research Policy*, 44(6):1226–1239, 2015. 12, 30, 74

[DKR99] Edward L Deci, Richard Koestner e Richard M Ryan. A meta-analytic review of experiments examining the effects of extrinsic rewards on intrinsic motivation. *Psychological Bulletin*, 125(6):627–668, 1999. 13, 14, 64, 66

[Don13] Anne Van Dongen. *Retention of New Blood Donors*. Tese de Doutorado, Maastricht University, 2013. 10

[DR87] E. L. Deci e R. M. Ryan. The support of autonomy and the control behavior. *Journal of Personality and Social Psychology*, 53(6):1024–1037, 1987. 16

[DS08] Paul A. David e Joseph S. Shapiro. Community-Based Production of Open Source Software: What Do We Know about the Developers Who Participate? *Information Economics and Policy*, 20(4):364–398, 2008. 15

[dSCK$^+$11] Carlos Denner dos Santos, Marcos Cavalca, Fabio Kon, Julio Singer, Victor Ritter, Damaris Regina e Tamy Tsujimoto. Intellectual Property Policy and Attractiveness : A Longitudinal Study of Free and Open Source Software Projects. Em *Proceedings of the 2011 ACM Conference on Computer Supported Cooperative Work, CSCW*, páginas 705–708, Hangzhou, China, 2011. 12

[Duc05] Nicolas Ducheneaut. Socialization in an Open Source Software Community: A Socio-Technical Analysis. *Computer Supported Cooperative Work*, 14(4):323–368, 2005. 13, 18

[Dug12] Vishal Dugar. Google Summer of Code. Introduction, Myths and How-Tos, 2012. 60

[FDKP11] Rosta Farzan, Laura Dabbish, Robert E. Kraut e Tom Postmes. Increasing Commitment to Online Communities by Designing for Social Presence. Em *Proceedings of the ACM 2011 conference on Computer supported cooperative work - CSCW '11*, página 321, Hangzhou, China, 2011. 11

[FHFM04] J Fulk, A J Heino, P R Flanagin e F. Monge. A Test of the Individual Action Model for Organizational Information Commons. *Organization Science*, 15(5):569–585, 2004. 11

[Fin95] Arlene G. Fink. *How to Ask Survey Questions.* SAGE Publications, Inc, vol 2 edição, 1995. 36, 60

[FJ11] Samer Faraj e Steven L. Johnson. Network Exchange Patterns in Online Communities. *Organization Science*, 22(6):1464–1480, 2011. 12

[FL13] Andrea Forte e Cliff Lampe. Defining, Understanding and Supporting Open Collaboration: Lessons from the Literature. *American Behavioral Scientist*, 57(5):535–547, 2013. 10

[FM92] K. Forsberg e H. Mooz. The Relationship of System Engineering to the Project Cycle. *Engineering Management Journal*, 4(3):36–43, 1992. 57

[FN09] Yulin Fang e Derrick Neufeld. Understanding Sustained Participation in Open Source Software Projects. *Journal of Management Information Systems*, 25(4):9–50, apr 2009. 1, 16, 18

[Fur96] Andrew Furco. Service-learning: A balanced approach to experiential education. *Expanding Boundaries: Serving and Learning*, 1:1–6, 1996. 75

[Gho05] Rishab Aiyer Ghosh. Understanding free software developers: Findings from the FLOSS study. *Perspectives on Free and Open Source Software*, páginas 23–45, 2005. 15, 16

[GK05] Robert J. Grissom e John J. Kim. *Effect Sizes for Research.* Routledge, 2005. 24

[GPVF15] Mohammad Gharehyazie, Daryl Posnett, Bogdan Vasilescu e Vladimir Filkov. Developer initiation and social interactions in OSS: A case study of the Apache Software Foundation. *Empirical Software Engineering*, 20(5):1318–1353, 2015. 3

[Gre06] Shirley Gregor. The nature of theory in information systems. *MIS Quarterly*, 30(3):611–642, 2006. 58

[Hem02] Andrea Hemetsberger. Fostering Cooperation on the Internet: Social Exchange Processes in Innovative Virtual Consumer Communities. *Advances in Consumer Research*, 29:354–356, 2002. 15, 16

[HK03] Eric Von Hippel e Georg Von Krogh. Open Source Software and the "Private-Collective" Innovation Model: Issues for Organization Science. *Organization Science*, 14((2)):209–223, 2003. 16, 17

[HNH03] Guido Hertel, Sven Niedner e Stefanie Herrmann. Motivation of Software Developers in Open Source Projects : An Internet-Based Survey of Contributors to the Linux Kernel. *Research Policy*, 32(7):1159–1177, 2003. 15

[HO02] A. Hars e S. Ou. Working for free? Motivations of participating in open source projects. *International Journal of Electronic Commerce*, 6:25–39, 2002. 15, 16

[HPS03] E. Haruvy, A. Prasad e S. Sethi. Harvesting Altruism in Open-Source Software Development. *Journal of Optimization Theory and Applications*, 118(2):381–416, 2003. 15

[HRSF02] Il-Horn Hann, Jeff Roberts, Sandra Slaughter e Roy Fielding. Why do developers contribute to open source projects? First evidence of economic incentives. 2002. 81

[L. 59] Allen L. Edwards. The Social Desirability Variable in Personality Research. *Journal of the American Statistical Association*, 54(03), 1959. 48

[Lan99] Ann Langley. Strategies for theorizing from process data. *Academy of management review*, 24(4):691–710, 1999. 71

[LJ07] Benno Luthiger e Carola Jungwirth. Pervasive Fun. *First Monday*, 12(1):5, 2007. 14, 15

[LMS13] N. Lacetera, M. Macis e R. Slonim. Economic Rewards to Motivate Blood Donations. *Science*, 340(6135):927–928, 2013. 55

[LRS17] Bin Lin, Gregorio Robles e Alexander Serebrenik. Developer Turnover in Global, Industrial Open Source Projects: Insights from Applying Survival Analysis. Em *Proceedings of the 12th International Conference on Global Software Engineering*, páginas 66–75, Buenos Aires, Argentina, 2017. 26

[LS05] Christoph Lattemann e Stefan Stieglitz. Framework for Governance in Open Source Communities. Em *Proceedings of the 38th Hawaii International Conference on System Sciences*, páginas 1–11, 2005. 15

[LV03] Karim R. Lakhani e Eric Von Hippel. How open source software works: "free" user-to-user assistance. *Research Policy*, 32(6):923–943, 2003. 14, 16

[LW91] Jean Lave e Etienne Wenger. *Situated learning: Legitimate Peripheral Participation.* Cambridge University Press, 1991. 18, 74

[LW05] Karim R. Lakhani e Robert G. Wolf. Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects. Em *Perspectives on Free and Open Source Software*, chapter Chapter 1, página 570. The MIT Press, Cambridge, MA, 2005. 14, 15, 16, 48, 66

[Mar14] Marylene Gagne. *The Oxford Handbook of Work Engagement, Motivation, and Self-Determination Theory.* Oxford University Press, New York, NY, USA, 2014. 66

[Mas12] Michelle M. Mason. Motivation, satisfaction, and innate psychological needs. *International Journal of Doctoral Studies*, 7:259–277, 2012. 13

[MBWS13] Jonathan T. Morgan, Siko Bouterse, Heather Walls e Sarah Stierch. Tea and sympathy. Em *Proceedings of the 2013 conference on Computer supported cooperative work - CSCW*, páginas 839–848, San Antonio, Texas, 2013. 12

[MD03] Terence R. Mitchell e Denise Daniels. Motivation. Em *Handbook of Psychology.* John Wiley & Sons, Inc., Hoboken, NJ, USA, apr 2003. 66

[Mer09] Sharan B. Merriam. *Qualitative Research: A Guide to Design and Implementation*, volume 1. Jossey-Bass, 3rd editio edição, 2009. 38

[MSM$^+$10] Paulo Meirelles, Carlos Santos Jr, João Miranda, Fabio Kon, Antonio Terceiro e Christina Chavez. A Study of the Relationships between Source Code Metrics and Attractiveness in Free Software Projects. Em *Proceedings - 24th Brazilian Symposium on Software Engineering, SBES 2010*, páginas 11–20, Salvador, Bahia, Brazil, 2010. IEEE. 18

[NCM18] Tuan Dinh Nguyen, Marisa Cannata e Jason Miller. Understanding student behavioral engagement : Importance of student interaction with peers and teachers. *The Journal of Educational Research*, 111(2):163–174, 2018. 5

[Ndu16] Ivange Larry Ndumbe. GSoC 2016 Final Evaluation, 2016. 66, 67

[Nic84] J.G. Nicholls. Achievement motivation: Conceptions of ability, subjective experience, task choice, and performance. *Psychological Review*, 91(3):328–346, 1984. 14

[NYN$^+$02] Kumiyo Nakakoji, Yasuhiro Yamamoto, Yoshiyuki Nishinaka, Kouichi Kishida e Yunwen Ye. Evolution patterns of open-source software systems and communities. Em *Proceedings of the international workshop on Principles of software evolution - IWPSE*, number 1, página 76, Orlando, Florida, 2002. xi, 16, 17

[OR07] M. Osterloh e S. G. Rota. Open Source Software Development—Just Another Case of Collective Invention? *Research Policy*, 36(2):157–171, 2007. 16

[PHT09] Katherine Panciera, Aaron Halfaker e Loren Terveen. Wikipedians are born, not made: a study of power editors on Wikipedia. Em *Proceedings of the ACM 2009 international conference on Supporting group work*, páginas 51–60, Sanibel Island, Florida, 2009. 12

[PNA04] Jenny Preece, Blair Nonnecke e Dorine Andrews. The top five reasons for lurking: improving community experiences for everyone. *Computers in Human Behavior*, 20(2):201–223, 2004. 11

[PP03] M. Poppendieck e T. Poppendieck. *Lean Software Development: An Agile Toolkit.* 2003. 57

[Pre00] J Preece. *Online Communities: Designing Usability, Supporting Sociability.* John Wiley & Sons, Chichester, UK, 2000. 10

[PS07] Naren B. Peddibhotla e Mani R. Subramani. Contributing to Public Document Repositories: A Critical Mass Theory Perspective. *Organization Studies*, 28(3):327–346, 2007. 11

[QF10] Israr Qureshi e Yulin Fang. Socialization in Open Source Software Projects: A Growth Mixture Modeling Approach. *Organizational Research Methods*, 14(1):208–238, 2010. 1

[Ral15] Paul Ralph. Developing and Evaluating Software Engineering Process Theories. Em *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, Florence, Italy, 2015. IEEE. 57

[Ral18] Paul Ralph. Toward Methodological Guidelines for Process Theories and Taxonomies in Software Engineering. 5589(c), 2018. 58, 60, 67, 74, 77

[Ray99] Eric Raymond. *The Cathedral and the Bazaar.* O'Reilly Media, 1999. 16

[RD00] Richard M Ryan e Edward L Deci. Intrinsic and Extrinsic Motivations: Classic Definitions and New Directions. *Contemporary Educational Psychology*, 25(1):54–67, 2000. 6, 66

[RHS06] Jeffrey a. Roberts, Il-Horn Hann e Sandra a. Slaughter. Understanding the Motivations, Participation, and Performance of Open Source Software Developers: A Longitudinal Study of the Apache Projects. *Management Science*, 52(7):984–999, 2006. 5, 15, 55, 66

[Rie07] D. Riehle. The Economic Motivation of Open Source Software: Stakeholder Perspectives. *Computer*, 40(4):25–35, 2007. 15

[RK09] Paul; Resnick e Robert E. Kraut. *Building Successful Online Communities: Evidence-Based Social Design.* The MIT Press, 2009. 10, 11, 55

[Ros04] M.A. Rossi. Decoding the " Free / Open Source (F / OSS) Software Puzzle " a survey of theoretical and empirical contributions. *The Economics of Open Source Software Development*, página 40, 2004. 13

[Sal15] Johnny Saldana. *The coding manual for qualitative researchers.* SAGE Publications, Inc, 2015. 58

[SC98] Anselm Strauss e Juliet M. Corbin. *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory.* SAGE Publications, 2nd editio edição, 1998. 37

[Sca02] W Scacchi. Understanding the requirements for developing open source software systems. Em *IEEE Proceedings Software*, páginas 24–39, 2002. 1

[SCGR15] Igor Steinmacher, Tayana Conte, Marco Aurélio Gerosa e David Redmiles. Social Barriers Faced by Newcomers Placing Their First Contribution in Open Source Software Projects. Em *Proceedings of the ACM Conference on Computer-Supported Cooperative Work & Social Computing (CSCW)*, páginas 1379–1392, Vancouver, BC, Canada, 2015. ACM. 31

[Sch13] Wilmar Schaufeli. *Employee Engagement in Theory and Practice.* Routledge, London, 2013. 5

[SDPR92] C. Scott Rigby, Edward L. Deci, Brian C. Patrick e Richard M. Ryan. Beyond the intrinsic-extrinsic dichotomy: Self-determination in motivation and learning. *Motivation and Emotion*, 16(3):165–185, 1992. 14, 15, 66

[SHK08] Sebastian Spaeth, Stefan Haefliger e Georg Von Krogh. Communal Resources in Open Source Development. *Information Research*, 13(1), 2008. 16

[Ski53] B. F. Skinner. *Science and Human Behavior.* 1953. 14

[SKKP13] Carlos Santos, George Kuk, Fabio Kon e John Pearson. The attraction of contributors in free and open source software projects. *The Journal of Strategic Information Systems*, 22(1):26–45, 2013. 18

[SLMH13] Edward Smith, Robert Loftin e Emerson Murphy-Hill. Improving developer participation rates in surveys. Em *6th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)*, San Francisco, CA, USA, 2013. 37

[SLW12] Andreas Schilling, Sven Laumer e Tim Weitzel. Who Will Remain? An Evaluation of Actual Person-Job and Person-Team Fit to Predict Developer Retention in FLOSS Projects. Em *Proceedings of the Annual Hawaii International Conference on System Sciences (HICSS)*, páginas 3446–3455, Maui, HI, 2012. 2, 6, 19, 30, 71, 73, 78

[Smi14] Melanie R Smith. *Retention of Firefighters in Volunteer Fire Departments in Suburban Nebraska.* Tese de Doutorado, Capella University, 2014. 9, 55

[SS18] K. Schwaber e J. Sutherland. The Scrum Guide, 2018. 57

[Ste15] Igor Fabio Steinmacher. *Supporting newcomers to overcome the barriers to contribute to open source software projects.* Tese de Doutorado, University of São Paulo, 2015. 1, 16, 18, 20, 55, 67

[SWC+14] Igor Steinmacher, Igor Scaliante Wiese, Tayana Conte, Marco Aurélio Gerosa e David Redmiles. The hard life of open source software project newcomers. Em *Proceedings of the 7th International Workshop on Cooperative and Human Aspects of Software Engineering - CHASE 2014*, páginas 72–78, New York, NY, USA, 2014. ACM Press. 43

[SWCG12] Igor Steinmacher, Igor Scaliante Wiese, Ana Paula Chaves e Marco Aurélio Gerosa. Newcomers withdrawal in open source software projects: Analysis of Hadoop common project. Em *Proceedings - 9th Brazilian Symposium on Collaborative Systems, SBSC*, páginas 65–74, Sao Paulo, SP, 2012. 3

[SWG+17] Jefferson O. Silva, Igor Wiese, Daniel German, Igor Steinmacher e Marco A Gerosa. How Long and How Much : What to Expect from Summer of Code Participants? Em *Proceedings of 33rd International Conference on Software Maintenance and Evolution (ICSME)*, página 10, Shanghai, China, 2017. 1, 21, 54, 71

[SWG+19] Jefferson O. Silva, Igor Wiese, Daniel M German, Christoph Treude, Marco A Gerosa e Igor Steinmacher. Google summer of code: Student motivations and contributions. *Journal of Systems and Software*, dec 2019. 35

[SWSG15] Jefferson O Silva, Igor Wiese, Igor Steinmacher e Marco A Gerosa. Pagamento Atrai Colaboradores de Longo Prazo ou Prestadores de Serviço? Uma Investigação Inicial sobre o Google Summer of Code 2014. Em *XII Simpósio Brasileiro de Sistemas Colaborativos, SBSC*, página 8, Salvador, BA, Brazil, 2015. 4

[TCH14] Erik H Trainer, Chalalai Chaihirunkarn e James D Herbsleb. The Big Effects of Short-term Efforts: Mentorship and Code Integration in Open Source Scientific Software. *Journal of Open Research Software*, 2(1):e18, jul 2014. 1, 2, 19, 31, 32, 71, 73

[TCKH14] Erik H. Trainer, Chalalai Chaihirunkarn, Arun Kalyanasundaram e James D. Herbsleb. Community Code Engagements: Summer of Code & Hackathons for Community Building in Scientific Software. Em *Proceedings of the 18th International Conference on Supporting Group Work - GROUP*, number 10, páginas 111–121, Sanibel Island, Florida, 2014. 1, 2, 5, 6, 19, 32, 71, 73

[The18] The Google Summer of Code Guides Authors. Google Summer of Code Mentor Guide, 2018. 59, 62, 71, 75

[TKCH16] Erik H Trainer, Arun Kalyanasundaram, Chalalai Chaihirunkarn e James D Herbsleb. How to Hackathon: Socio-technical Tradeoffs in Brief, Intensive Collocation. Em *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing - CSCW '16*, páginas 1116–1128, 2016. 32, 71

[TL02] Jean Tirole e Josh Lerner. Some Simple Economics of Open Source. *The Journal of Industrial Economics*, 50(2):197–234, 2002. 15, 16, 66

[Tur01] J. C. Turner. Using context to enrich and challenge our understanding of motivation theory. Em Simone (Ed) Volet e Sanna Jarvela (Ed), editors, *Motivation in learning context: Theoretical advances and methodological implications*, chapter Using cont, página 334. Emerald Group Publishing Limited, 2001. 35

[VHH+99] C Veitch, J Harte, R Hays, D Pashen e S Clark. Community participation in the recruitment and retention of rural doctors: methodological and logistical considerations. *The Australian Journal of Rural Health*, 7(4):206–211, 1999. 10

[vKHSW12] G. von Krogh, S. Haefliger, S. Spaeth e M. W Wallin. Carrots and rainbows: Motivation and social practice in open source software development. *MIS Quarterly*, 36(2):649–676, 2012. 14, 54

[VSL03] Georg Von Krogh, Sebastian Spaeth e Karim R. Lakhani. Community, joining, and specialization in open source software innovation: A case study. *Research Policy*, 32(7):1217–1241, 2003. 13, 17

[WdSS$^+$16] Igor Scaliante Wiese, Jose Teodoro da Silva, Igor Steinmacher, Christoph Treude e Marco Aurelio Gerosa. Who is Who in the Mailing List? Comparing Six Disambiguation Heuristics to Identify Multiple Addresses of a Participant. Em *2016 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, páginas 345–355, Raleigh, NC, USA, 2016. IEEE. 22

[WF05] M M Wasko e Samer Faraj. Why should I share? Examining social capital and knowledge contribution in electronic networks of practice. *MIS Quarterly*, 29(1):35–38, 2005. 11

[Wit13] Bill Wittich. *Attract the Best Volunteers: Stop Recruiting and Start Attracting*. Knowledge Transfer Publishing, 2013. 9

[XJS09] B. Xu, D. R. Jones e B. Shao. Volunteers' Involvement in Online Community based Software Development. *Information & Management*, 46(3):151–158, 2009. 15

[YJC07] J. Yu, Z. Jiang e H. C. Chan. Knowledge Contribution in Problem Solving Virtual Communities: The Mediating Role of Individual Motivations. Em *Proceedings of the 2007 ACM SIGMIS CPR Conference on Computer Personnel Doctoral Consortium and Research Conference*, páginas 144–152, St. Louis, MO, 2007. 15, 16

[YK03] Yunwen Ye e K. Kishida. Toward an understanding of the motivation of open source software developers. Em *Proceedings of the 25th International Conference on Software Engineering, ICSE*, páginas 419–429, Portland, Oregon, 2003. 16

[ZM12] Minghui Zhou e Audris Mockus. What make long term contributors: Willingness and opportunity in OSS community. Em *ICSE '12 Proceedings of the 34th International Conference on Software Engineering*, páginas 518–528, Zurich, Switzerland, 2012. 16

[ZZH$^+$13] Haiyi Zhu, Amy Zhang, Jiping He, Robert E Kraut e Aniket Kittur. Effects of Peer Feedback on Contribution : A Field Experiment in Wikipedia. Em *CHI '13 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, páginas 2253–2262, Paris, France, 2013. 55, 56