A Four-Year Study of Student Contributions to OSS vs. OSS4SG with a Lightweight Intervention

Zihan Fang
zihan.fang@vanderbilt.edu
Vanderbilt University
USA

Madeline Endres
endremad@umich.edu
University of Michigan
USA

Thomas Zimmermann
tzimmer@microsoft.com
Microsoft Research
USA

Denae Ford
denae@microsoft.com
Microsoft Research
USA

Westley Weimer
weimerw@umich.edu
University of Michigan
USA

Kevin Leach
kevin.leach@vanderbilt.edu
Vanderbilt University
USA

Yu Huang
yu.huang@vanderbilt.edu
Vanderbilt University
USA

ABSTRACT

Modern software engineering practice and training increasingly rely on Open Source Software (OSS). The recent growth in demand for professional software engineers has led to increased contributions to, and usage of, OSS. However, there is limited understanding of the factors affecting how developers, and how new or student developers in particular, decide which OSS projects to contribute to, a process critical to OSS sustainability, access, adoption, and growth. To better understand OSS contributions from the developers of tomorrow, we conducted a four-year study with 1,361 students investigating the life cycle of their contributions (from project selection to pull request acceptance). During the study, we also delivered a lightweight intervention to promote the awareness of open source projects for social good (OSS4SG), OSS projects that have positive impacts in other domains. Using both quantitative and qualitative methods, we analyze student experience reports and the pull requests they submit. Compared to general OSS projects, we find significant differences in project selection ($p < 0.0001$, effect size = 0.84), student motivation ($p < 0.01$, effect size = 0.13), and increased pull-request acceptance rates for OSS4SG contributions. We also find that our intervention correlates with increased student contributions to OSS4SG ($p < 0.0001$, effect size = 0.38). Finally, we analyze correlations of factors such as gender or working with a partner. Our findings may help improve the experience for new developers participating in OSS4SG and the quality of their contributions. We also hope our work helps educators, project leaders, and contributors to build a mutually-beneficial framework for the future growth of OSS4SG.

ESEC/FSE '23, December 3–9, 2023, San Francisco, CA, USA

ACM ISBN 979-8-4007-0327-0/23/12…$15.00
https://doi.org/10.1145/3611643.3616250

CCS CONCEPTS

• **Software and its engineering** → *Open source model*; • **Human-centered computing** → *Empirical studies in HCI*.

KEYWORDS

Open Source Software, Social Good, CS Education

ACM Reference Format:
Zihan Fang, Madeline Endres, Thomas Zimmermann, Denae Ford, Westley Weimer, Kevin Leach, and Yu Huang. 2023. A Four-Year Study of Student Contributions to OSS vs. OSS4SG with a Lightweight Intervention. In *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE '23), December 3–9, 2023, San Francisco, CA, USA.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3611643.3616250

1 INTRODUCTION

Open source refers to both the concept and also the practice of making a program's source code openly available [35]. Since the concept of *open source software* (OSS) was first introduced in 1998, it has grown to involve not only academics and hobbyists, but also professional software engineers, influencing nearly 78% of US companies by 2015 [55]. Furthermore, in recent years, there has been significant interest in OSS projects that have the potential to benefit society in a broader way, such as through humanitarian efforts in domains like microfinance, healthcare, education, and disaster relief [31, 44]. This category of software projects is called *Open Source Software for Social Good* (OSS4SG) [31].

At the same time, there has been an increase in the demand for professional software engineers [5], as well as an associated growth in enrollment in undergraduate programs that prepare students for software jobs where OSS is common [40]. Ellis *et al.* claim that student developers have characteristics which make them particularly well-suited for participating in OSS [15]. In addition, as many computing students are future developers, how and why student developers choose OSS projects to contribute to may have implications for improving OSS sustainability and growth. However, student contributions to OSS are comparatively understudied.

There is limited understanding of how students can gain practical experience with more realistic software engineering tasks [25], as well as which methods educators can use to motivate students (e.g., for recruitment and retention, through the lens of OSS.

Understanding the motivations of new or student developers' OSS contributions may be of particular import for OSS4SG projects. Previous work demonstrates that, despite their societal potential, many OSS4SG projects are inhibited by a lack of advertisement and awareness among professional developers [31]. For example, Huang *et al.* found that it was more challenging for contributors to find OSS4SG projects to work on even though "there are many, many, many developers that might want to" [31, Sec. III-D].

We hypothesize that one way to increase OSS4SG project awareness is through *educational interventions*. At many universities, undergraduate software engineers contribute to OSS projects as part of a course assignment [12, 43]. OSS contribution can give students practical experience with many software development tasks such as understanding triaged issue reports, localizing faults in a legacy code base, reading and writing documentation, testing, and submitting a formal pull or change request [39]. As a result, many developers may first encounter OSS as undergraduates and such assignments may be an opportune time to raise OSS4SG awareness. However, prior studies of OSS in educational contexts focused on how students and educators *use* OSS (e.g., studying common mistakes in student contributions [30]), rather than on how students *choose* and *learn* from OSS projects. Furthermore, there have been no studies to-date that have investigated the impact that OSS4SG projects can have on students.

To help close this gap, we present the results of a four-year study (2018–2022) of OSS contributions from students across eight semesters of a senior software engineering course at University of Michigan. As the course final project, students had to contribute to an OSS project *of their choice*. To investigate our hypothesis that an educational intervention can increase OSS4SG contributions, for the last four semesters, we implemented a lightweight *intervention* introducing OSS4SG to students on the course website[1]. For all eight semesters, the course content, projects, and homework assignments were the same with the exception of intervention-related text. Additionally, the same two instructors taught the course, and student computing preparation remained consistent as enrollment and major declaration policies remained unchanged during the study period. This consistency permits analysis of trends across semesters.[2]

In total, we collected and analyzed 984 reports from 1,361 students[3] contributing to 443 different OSS projects. Our analysis provides an understanding of the life cycle of student contributions to OSS in general and to OSS4SG in particular. We also analyzed the effect of the lightweight intervention on OSS4SG participation. Following previous work [31], we determine which projects are OSS4SG using the combined criteria of Ovio [2] and the Digital Public Goods Alliance (DPGA) [1], which is derived from the United Nation's 17 Sustainable Development Goals [45].

Inspired by previous work that demonstrate key aspects in OSS participation [10, 16, 36], we focus on three key phases of the contribution life cycle when contrasting Non-social good (Non-SG) and OSS4SG projects:

(1) **Project selection**: Project topic, motivation, pair programming decisions, and the impact of our OSS4SG intervention.
(2) **Project experience**: Subjective experiences, lessons learned, and programming languages used.
(3) **Community reception**: Pull request acceptance rates and the factors that most influence them.

We find statistically-significant differences at all stages of this life cycle between students who engage with OSS4SG projects and those who do not. As some examples, student motivations and selected project topics are different between Non-SG and OSS4SG; our lightweight intervention (and to a lesser extent, the COVID-19 pandemic) influences the degree to which OSS4SG projects are selected; and pull request contributions to OSS4SG projects are more likely to be accepted and merged into the main project. To the best of our knowledge, this is the first large-scale (in terms of students, time, and projects) analysis of student OSS4SG contributions.

The main contributions of this paper are:

- The first large-scale study of student OSS contributions (including data from 1,361 students over four years).
- An analysis of how the life cycle of student OSS contributions (project selection, contribution experience, and community reception) differs between OSS4SG and Non-SG projects.
- A set of actionable recommendations and evidence, for educators and student developers, of which projects may be most effective for students, and the impact of a lightweight intervention to bring social good to student attention.

Ultimately, we believe OSS, and OSS4SG in particular, are both morally and pragmatically important: it helps software engineering as a community to contribute to humanitarian efforts more directly, and provides an additional avenue for attracting and preparing students from all backgrounds to succeed in the workforce. The goal of this research is to learn factors influencing student OSS contributions in educational settings, as well as with a goal of encouraging participation in OSS4SG, so that we can better support future OSS contributions in general. We believe our study also provides an important first step for leveraging the understudied potential of open source for social good.

2 BACKGROUND AND RELATED WORK

In this section, we discuss related work on open source software and social good (OSS4SG), the intersection of OSS and computing education, and related educational interventions.

2.1 Open Source Software and Social Good

In recent years, the role played by OSS in key areas such as web services [13] has been the subject of increasingly-prominent discussions. Although some established software companies treat OSS as a threat [4], the more common view is that it generally positively impacts society, business, education, and research [33], and leads

[1]No requirement or extra credit was applied to students contributing to OSS4SG. Our Intervention Script is available in the supplementary materials.
[2]The COVID-19 pandemic occurred one semester prior to the implementation of the intervention. To validate our analysis considering the effect of the pandemic, we also analyzed the students' data prior to and following the onset of COVID-19 (see Section 6).
[3]Students were allowed but not required to work in teams of two.

to a more open society [23] because it partially addresses three key software challenges: cost, time scale, and quality [22].

OSS also changes the way people and organizations develop, acquire, use, and commercialize software. Ellis *et al.* emphasizes that OSS can help increase interest in computing [44] and provide career advancement benefits based on sustained participation [46]. As the OSS landscape has changed, participant motivations for joining OSS projects have changed as well [24]. Software developers are increasingly deciding to use their technical skills and OSS to benefit the common good of society [31]; computing for social good has become a common topic in technical circles [21], and developers are more likely to volunteer to write code that improves societal processes when given the opportunity [34]. Simultaneously, there has also been a growing global political interest in encouraging OSS usage and access as more policy makers learn how essential OSS is to our world's digital infrastructure [42].

The smooth integration of newcomers is critical for the sustainability and growth of OSS projects [52]. Approaches such as "Code for America" and various "Hackathon" events are one way innovative software solutions with a social conscience are encouraged [19]. However, unlike OSS focusing on technical development (non-SG), finding OSS projects with broader societal impact (OSS4SG) to contribute remains a top challenge for developers and OSS4SG has not yet been well studied by the research community in software engineering [31]. While prior studies explored general OSS contributor obstacles, none focused on student contributions to OSS4SG projects. Additionally, no research analyzed and compared developer disparities in OSS and OSS4SG within an educational context. In this work, we examine how student developers find and choose to contribute to OSS4SG projects as a first step to alleviating this challenge.

2.2 Open Source Software and Education

Open source has been seen as a natural fit for education because it encourages collaborators to pool resources and expertise so as to build new ways of solving difficult problems [7]. Some institutions integrate open source development into education [12], with the view that participating in OSS will help students to succeed in their careers [43]. Because of its broad appeal, OSS can help increase interest in computing [17] and it has also been shown to help retain involvement in computing over the long term [51], despite the challenges students may face when participating in large-scale software development projects [48]. Furthermore, students' involvement in large-scale software development projects enhances their academic curriculum through practical experience in real-life software engineering practices [29]. OSS4SG can be particularly appealing for students [12], including those from less-represented backgrounds (e.g., women students, first-generation students, LBGTQ+ students, etc.) [14].

However, project selection is a potential risk when using OSS contributions in CS education. Project selection facilitates if and how students improve professional skills when contributing to OSS [17, 44] — the student's motivation, perseverance, and acceptance by the community are inherently related to the project chosen. Studies are also undertaken to analyze the strategies by which students can be motivated to engage in OSS [50]. In addition, inexperienced student software engineers can also create drawbacks for projects. Hu *et al.* delineate 13 common mistakes based on 313 OSS projects that often occur in student contributions (e.g., not following existing designs, messy pull requests, etc.) [30]. Previous studies have also examined the factors that impact the acceptance of pull requests, including the quantity of comments received, the project's historical background and the developers' reputation [37]. However, no prior research has investigated the acceptance of pull requests specifically by students and the factors from the students' perspective that influence the outcomes of their pull requests.

Despite these challenges, it is often believed that maintaining and developing OSS can help students improve their resumes and their social and technical skills [43]. Large tech companies like Google, Yahoo, and Facebook have described the benefits of hiring people involved in the open source community [12]. In this study, we provide initial investigation of Non-SG and OSS4SG contributions by undergraduate software engineering students who are considered representative of the forthcoming OSS developers, which has not been examined in previous research.

2.3 Educational Interventions

An *intervention* is a purposeful action by a human agent to create change [41]. Interventions are commonly used in computing and STEM education to improve programming performance or foster certain behaviors [18]. Controlled experimental design is the primary method to explore whether such interventions have effects on specific populations [38].

The integration of social issues into computing curricula to guide students to be socially-responsible professionals is still a work in progress [28]. Bagli *et al.* and Goldweber *et al.* have proposed interventions related to incorporating social good into computing curricula [26]. The intervention we evaluate in this paper is most similar to that proposed by Goldweber *et al.*, where introductory computing assignments are reframed as social good problems to motivate students [26]. However, rather than modifying internal course projects, we propose an intervention exposing students to external OSS4SG projects that are likely to have real world impact.

3 STUDY DESIGN

We investigate the life cycle of student developer submissions to OSS projects with a focus on how this life cycle differs for contributions to OSS4SG projects compared to standard (i.e., Non-SG) OSS projects. To do so, we analyze 984 project reports submitted by 1,361 computer science students (794 working in 397 teams, 567 working alone) in a senior software engineering course at a large public university. Students were tasked with selecting an OSS project to contribute to, working on one or more active issues, and submitting a pull request with the results of their work. These project reports span eight semesters over four years (2018–2022) and include contributions to 443 unique OSS projects. We consider both statistical differences between Non-SG and OSS4SG efforts as well as the impact of a lightweight intervention (making students aware of the existence of OSS4SG) starting in 2020.

In the remainder of this section, we detail the contents of our dataset and discuss the qualitative methods we used to prepare our data for statistical analysis.

3.1 Dataset Overview

Our data is collected from students who have enrolled in *EECS 481 at University of Michigan* over the past four years. This course follows a similar model to offerings at other large US institutions, guiding the student developers to become professional software engineers by focusing on collaborating in a team, managing complexity, testing and quality assurance, mitigating risks, staying on time and budget, requirements elicitation, and so on.

During the semester-long course, students were required to contribute to an open source project. Students had to select an OSS project of their choice from GitHub, identify an open issue in the project (typically a bug report or feature request), understand the local development process, localize and make the desired change, then submit a pull request. Students documented their experiences with two written project reports: an initial project plan and a final reflection. In the initial report, students identified a project, proposed a schedule, discussed the project's build system, and described the intended defect to fix or feature to add. In the final reflection, students had to include: descriptions of how and why they selected the OSS project and task; the project context; the project governance (including processes and tools used to coordinate and communicate among developers); the task attempted; any submitted artifacts (e.g., code changes, new documentation, test case reports, etc.); a URL to any submitted pull requests; the student's subjective experience; and recommendations and advice for other students. Extra credit points were awarded for contributions merged into the project's codebase (i.e., pull requests accepted on GitHub) before the end of the semester. Students could choose to work alone or in pairs.

Our dataset consists of 984 textual project reports as well as any associated publicly-available pull requests. In addition, course staff provided the authors "recalled gender"[4] associated with each de-identified student (e.g., man, woman, do not recall, etc.). This was done based on recollections from lectures, office hours, online meetings, pronouns used by students, roster information, etc. Ultimately the course staff provided gender annotations for 93.9% of students. Other demographic information, such as race or ethnicity, could not be collected.

3.2 Intervention

To assess the degree to which a lightweight intervention could impact student project selection and promote OSS4SG, starting in the 5th semester (fall of 2020) the project description was updated to include a short (ten sentence) description of computing for social good as well as optional links to the DPGA [1] and Ovio [2] lists and descriptions of OSS4SG. In addition, the text clarified that choosing a social good project would have no impact on the assignment grade. Students were also asked to include a sentence in their final reports indicating whether they thought their project contributed to social good, but were again reminded that the answer would not affect their grade. The exact text of the intervention is available in the supplementary materials. The first half of our dataset (i.e., the first four semesters) *does not* include the intervention, while the second half *does* include it (see Table 1). Other than the intervention text, course materials were consistent across all eight semesters. Additionally, the same two instructors taught the class and students had similar computer science preparation throughout as university enrollment and major declaration policies were not changed. This stability helps minimize confounding variables, and permits comparison between pre-intervention and intervention semesters.

3.3 Analysis Methodology

Our study protocol was assessed by our local IRB and exempted from further review (HUM00220536). At a high level, de-identified project reports were provided to the study team by the course staff. After preprocessing, the study team conducted a series of qualitative and quantitative analyses of the reports and associated pull requests.

Student Report Qualitative Preprocessing: Student reports were originally in free-form text. Therefore, it was necessary to manually read and code the reports to extract the data needed for statistical analysis. To do so, we constructed an initial codebook with seven categories for the reports using the assignment specification as a guideline. For any given report, this coding identified basic information about the OSS project and tasks picked, why the students selected this project, their experiences working on the project, and their advice for future students. The construction of the codebook was thoroughly reviewed to ensure accuracy and completeness, with multiple authors (including former course staff) identifying potential confounding factors and suggesting modifications. Specifically, the authors used `Atlas.ti`[5], a commonly used platform for qualitative analysis, to conduct the thematic analysis. Initially, the first author independently analyzed ten transcripts, identifying recurring open codes by labeling significant participant statements. Discussions with co-authors followed to establish connections, resolving category disagreements through extensive discourse, transcript review, and refinement. The first author then labeled data based on the refined code book. Collaboratively, relationships among codes were identified, leading to organized, meaningful themes. Furthermore, finalizing the codebook was a dynamic process; Following best practices [20], the codebook was modified whenever the annotator observed a concept that was not yet included. The first author reached code-book saturation (i.e., no new modifications of the codebook) after approximately 500 reports. These first 500 reports were re-coded with the finalized codebook by the first author.

To gain confidence in the accuracy of our coding, 50 project reports were randomly selected and thoroughly re-coded by the second author. This resulted in 13-14 re-coded high-level categories per project (or 686 in all). To quantify agreement, we present a contextualized percentage agreement (as the labels to categories are not mutually exclusive, we can not use the common Cohen's Kappa to calculate agreement); Excluding minor typos, the first and second authors agreed exactly in 623/686 cases, or 90.8% of the time, indicating generally high agreement. For non-mutually exclusive categories, the exact agreement indicates that both authors picked the same subset of codes as labels for a project. The 10% disagreement was generally caused by the second author including one or more additional complimentary labels to a multi-label category.

[4] Our original IRB-approved protocol did not explicitly ask for student-reported gender. As a result, our IRB suggested using *recalled gender* based on instructor-accessible information about students.

[5] https://web.atlasti.com

There were only 9 cases (1.3%) where the second author's coding was in direct conflict with the first author's. Once the coding process was completed, the remaining authors double-checked 25% of the coding results and conducted a high-level review of the pull request section of the qualitative data to ensure its quality and integrity.

We include our complete codebook in the supplementary materials. However, as a clarifying example, we provide our categorization of students' contribution motivations. We identified these five categories of motivations pointed out by the students in their reports:

- **Useful or Helpful**: the OSS project would be useful and benefit human daily lives.
- **Active**: number of recent issues, pull requests, contributors, etc.
- **Easy to Work On**: the organization of the project, the general difficulty level of the project, students' familiarity with the programming language, and the skills required in the project.
- **Interest**: the interests of the students aligned with the topic of the project.
- **Project Influence**: the popularity or impact of the project.

For many categorizations, including this one, multiple tags can apply simultaneously, resulting in overlap among categories.

Student Pull Request Qualitative Preprocessing: To assess how student contributions were received by OSS communities, we also investigated how many of the pull requests have been accepted and merged. To do so, the authors manually visited all pull request URLs provided in the project reports. 772 out of 984 reports contained at least one valid pull request link, and in total there 989 submitted requests (some groups submitted multiple times). For the remaining 212 reports, either the team did not submit a pull request (25/212) or did not include a valid link (187/212).

The authors read all pull requests and labeled them as accepted (e.g., merged or force-merged into the project by project owners/maintainers) or not (e.g., closed and not merged, still open). To do so, we used a two-phase approach: if the pull request was marked as Merged in the GitHub interface, we coded it as accepted. If the pull request was marked as Open or Closed, we manually inspected the comments and activity (e.g., moving the changes to a new request, referencing the pull request in another commit, or other project-specific processes). We did not rely on GitHub's interface alone as it can underestimate acceptance rates [32]. Our process may still not spot all accepted contributions (e.g., those incorporated into later pull requests by non-student developers, etc). However, we believe we identify most student-written accepted pull requests, aligning with the educational context of our work.

For each pull request, we also recorded the dates of the pull request submission and acceptance (if applicable) as well as the category of the issue the student was addressing (e.g., bug fix, feature request, test cases, refactoring, etc.). This pull request annotation data is used in our analysis of how student OSS contributions are received by the OSS community.

Project Topic and Social Good Labels: Beyond coding the student reports and associated pull requests, to conduct our analysis, we also labeled both the *topic* of a project and if the project is qualified as *OSS4SG*. We used the labels provided by Ovio [2] as general topics of projects. We used labels provided by DPGA [1] as the criteria to determine whether a project qualifies as supporting social good. Ovio [2] is an authority for matching software developers with projects; their site includes an explicit list of OSS projects associated with social good or technology, broadly construed. However, not all projects from Ovio [2] are social good projects per se. Notably, many computing projects, such as Microsoft's open source `VSCode` editor, would officially fall in the loose Ovio [2] categories of '`industry-innovation-infrastructure`' or '`reduced-inequality`' but do not align with conventional notions of OSS4SG [31] (e.g., VSCode does create infrastructure for industry and *could* be used in a social good context, and since it is free it *could* be seen as reducing inequality, but it is not inherently or essentially a social good project by more conventional definitions). Informally, almost every OSS project could be seen reducing inequality (since OSS is free to download) and/or as providing innovative infrastructure (since it is software), so those tags are not informative in this context. As a result, we followed best practices [20, 31] and used DPGA [1] criteria to help determine social good scoping more precisely; DPGA [1] makes use of the United Nation's 17 Sustainable Development Goals which have been found to more closely align with what computing practitioners mean by the term "social good" [31, 45].

For our study, after collecting labeled category data for each project, if a project meets any criterion other than '`industry-innovation-infrastructure`' and '`reduced-inequality`,' it is classified as OSS4SG. The final classification of each project was cross-checked by the authors. Our dataset is available and includes all of the original tags (so researchers can rerun the analysis with those tags included if desired).

4 ANALYSIS RESULTS

In this section, we discuss the results of our analysis of the lifecycle of student contributions to OSS. We organize our results around three key phases of this lifecycle:

- *Phase 1 — Project Selection:* What projects and topics do students choose when electing to contribute to OSS? What factors determine that decision? Examining developers' motivation is crucial for understanding OSS development and achievements, as highlighted in prior research [36]. In this phase, we analyze the preferences and choices of students as well as the impact of our intervention promoting the selection of OSS4SG projects.
- *Phase 2 — Contribution Experience:* How do students describe their experiences working on open source projects? Does this experience differ by project topic? OSS projects play a crucial role in providing individuals with practical software development skills that are often challenging to acquire through formal education [16]. In this phase, we explore students' contribution experiences and the lessons they have learned.
- *Phase 3 — Community Reception:* How are student submissions received by the open source community? Understanding the dynamics of collaboration and participation in open source communities requires recognition of community acceptance as a vital factor [10]. In this phase, we explore the factors that correlate with whether or not a student-submitted pull request was ultimately accepted.

Table 1: Student and report summary. "OSS4SG" counts reports associated with a social good project while "Non-SG" counts reports for non-social good projects. "Intervention" indicates the presence of a light pedagogical awareness intervention.

Semester	Students	Reports	OSS4SG	Non-SG	Intervention
1	95	70	7	63	✗
2	113	83	12	71	✗
3	124	105	30	75	✗
*4	130	92	28	64	✗
5	218	163	35	127	✓
6	249	171	55	116	✓
7	146	100	37	62	✓
8	291	200	57	143	✓
Total	1,361	984	262	722	

* Semester 4 coincided with the onset of COVID-19 (early 2020). See Section 6.

Table 2: Breakdown of student self-reported motivation categories for selecting 984 projects (for all OSS projects and OSS4SG) We find a significant difference of motivation of selection on Non-SG between SG.

Motivation	All Projects	OSS4SG	OSS4SG(%)
Useful or Helpful	379	116	30.6
Active	179	37	20.7
Easy to Work On	134	31	23.1
Interest	105	31	29.5
Project Influence	57	7	12.3
*N/A	130	39	30.0

* Students did not state their motivation of selection in reports.

In each phase, we directly compare and contrast between OSS4SG projects and OSS projects that are not explicitly related to social good (Non-SG OSS). Table 1 summarizes our dataset over the eight semesters covered by this study, broken down by non-social good contributions ("Non-SG") and social good contributions ("OSS4SG").

4.1 Phase 1 — Project Selection

For this phase, we first give an overview of student motivations and project topic selection. We also investigate if there are differences related to student gender or if the student worked alone or in a pair. We then compare OSS4SG and Non-SG projects across topics and evaluate our intervention's effectiveness.

4.1.1 Overview of Student Motivations and Project Topics. Table 2 summarizes the project selection motivations described in student reports (see Section 3.3 for a description of qualitative methods used to identify themes and code reports). Notably, while conventional wisdom [8] might suggest that many students are looking for projects that align with their personal interests or that will be easy, the dominant motivation mentioned for project selection was that the project was Useful or Helpful in some way (to the student personally or to others). *In fact, whether a project was seen as useful or helpful was cited more than twice as often as the next most common motivation for choosing a project.* While this notion of "helpful" is not strictly the same as "social good" (see Section 3.3), our results are consistent with other findings that suggest a broader humanitarian impact can be a strong motivator for students [6]. In addition, a prior investigation revealed gender bias within OSS communities [53], especially for women who are underrepresented in OSS [54].

Table 3: General topics of projects selected by students.

General Topic	Projects	OSS	OSS4SG
Technology	394	394	0
Management and Productivity	210	181	29
Creativity and Entertainment	172	145	27
Education	117	1	116
Society and Culture	68	9	59
Health and Well Being	40	0	40
Science	40	37	3
Civic Tech	34	3	29
Economics	19	5	14
Humanitarian	17	12	5
Environment and Nature	16	0	16
Literature and Journalism	7	5	2

Table 4: Motivation for project selection as a function of working alone vs. as a pair. We find a significant difference on students' motivation of selection between working alone and working in pairs (χ^2 = 10.9, p< 0.05 , Cramer's V = 0.11).

Motivation	Alone		In Pairs
Useful or Helpful	218 (42.7%)	↗	161 (47.0%)
Active	97 (19.0%)	↗	82 (23.9%)
Easy to Work On	85 (16.6%)	↘	49 (14.3%)
Interest	76 (14.9%)	↘	29 (8.5%)
Project Influence	35 (6.9%)	–	22 (6.4%)

However, our results hold for students generally: we find no evidence that student gender (our dataset's main demographic label) influenced the ranking or proportion of these categories ($p > 0.05$).

Table 3 summarizes the projects selected by students according to general topics (see Section 3.3). The most common project topics are Technology (e.g., `VSCode`) and Management and Productivity (e.g., `Zulip`). As a whole, most students prefer Technology and Entertainment projects to other topics. In addition, we speculate that these results may help to identify suitable projects to which undergraduate students are more likely to express interest.

As shown in Table 4, we also investigated whether or not collaboration impacts project selection. We found a statistically-significant difference in project selection motivations for students working alone vs. working in pairs (χ^2 = 10.9, p < 0.05, Cramer's V = 0.11). Students working in pairs are more likely to prioritize Useful or

Table 5: Motivation for project selection for Non-SG vs. OSS4SG projects. We find a significant difference on motivation for selection between Non-SG vs. OSS4SG (χ^2 = 13.9, p< 0.01 , Cramer's V = 0.13).

Motivation	Non-SG		OSS4SG (262)
Useful or Helpful	263 (41.6%)	↗	116 (52.3%)
Active	142 (22.5%)	↘	37 (16.3%)
Easy to Work On	103 (16.3%)	↘	31 (14.1%)
Interest	74 (11.7%)	↗	31 (14.1%)
Project Influence	50 (7.9%)	↘	7 (3.2%)

Helpful or Active projects than are students working alone. We hypothesize that students working together are more likely to prioritize the project itself (e.g., as a shared choice) than to focus on their personal suitability or interest for the project (Easy to Work On, Interest, etc.). This is important because students may feel they are not capable of accomplishing as much if working alone, influencing the type of project they decide to pursue.

By far, the most common motivation cited by students was if the project was Useful or Helpful, indicating students take personal or societal impact into account when choosing projects. This trend was even more pronounced for students who worked in a pair. The most common project topics were Technology and Entertainment.

4.1.2 Motivations for Contributing to OSS4SG. Overall, 26.6% of projects selected by students were classified as Open Source for Social Good following our methodology (Section 3.3). Critically, we find that project motivations are significantly different between Non-SG and OSS4SG projects (χ^2 = 13.9, p < 0.01, Cramer's V = 0.13). We conducted a logistic regression analysis on all selection factors. The results revealed that students' motivation significantly contributed to their selecting OSS4SG projects or not (p < 0.05). Useful or Helpful is an even more common motivation for OSS4SG, and Interest is more significant for OSS4SG as well. We hypothesize that one reason student developers choose to contribute to OSS4SG is that they attach importance to the personal or global significance of the contribution (Useful or Helpful, Interest) rather than to the likelihood of success (Active, Easy). Table 5 summarizes this result.

4.1.3 OSS4SG Project Topics. We investigate topical differences between selected OSS4SG and Non-SG projects. To analyze this, we consider topics that overlap between Non-SG and OSS4SG projects; Figure 1 shows these results. Management and Productivity is the most popular project topic among students selecting Non-SG projects, followed by Creativity and Entertainment and then Science. By contrast, OSS4SG contributions focused on topics such as Education, Health and Well-Being, and Society and Culture. In addition, to a lesser degree, OSS4SG project topics touched on Civic Technology as well as Environment and Nature. These differences are strongly significant (χ^2 = 455.5, p < 0.0001, Cramer's V = 0.84). In a linear regression, we found that topic also significantly impacts (p < 0.0001) contributing to Non-SG vs. OSS4SG. These

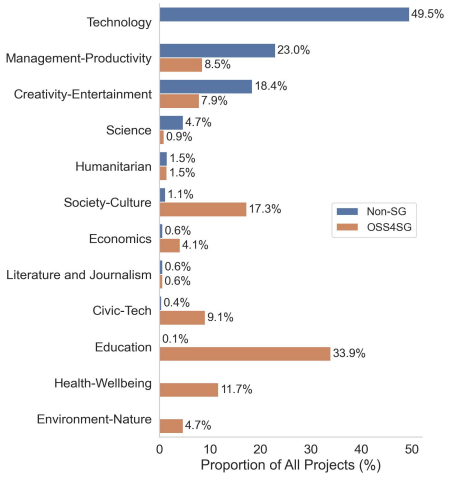

Figure 1: Topics for student-selected OSS and OSS4SG projects (note that all projects labeled Technology belong to Non-SG and all projects labeled Health and Well Being or Environment and Nature belong to OSS4SG).

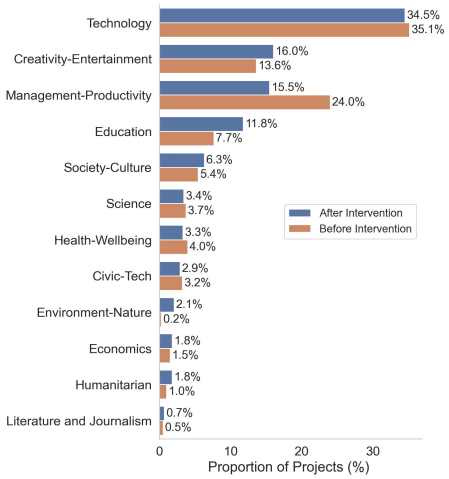

Figure 2: The effect of a lightweight pedagogical intervention (mentioning social good but not grading on it in any way) on student project selection.

results align with conventional wisdom [8] (e.g., students favoring OSS4SG focus on topics more likely to benefit society, while students favoring Non-SG focus more on personal development), and they provide concrete guidance for educators and trainers selecting topics or examples when providing advice about project selection.

4.1.4 Gender Effect and Social Good. As discussed in the overview of Phase 1, we do not observe a significant difference in topic selection by gender (e.g., men vs. women, men vs. not-identifying-as-men, etc., p = 0.54—956 men, 319 women, 3 non-binary, and 83 unknown or unreported). In addition, while a greater fraction of women students contributed to OSS4SG projects than did men, the effect was not significant (p = 0.053). However, we draw attention

Table 6: Effect of gender on selecting a Non-SG vs. OSS4SG project in specific (956 men, 365 women, 3 non-binary).

Gender	Non-SG	OSS4SG
Men	709 (74.2%)	247 (25.8%)
Women	218 (68.3%)	101 (31.7%)
Non-Binary	3 (100.0%)	0 (0.0%)

to the potential impact of social good aspects in OSS on individuals' access to and involvement to OSS in general. Gender-related factors can influence opportunities and contributions within specific fields, emphasizing the need to consider gender as a relevant aspect in the study of OSS4SG participation [54]. By delving into the interplay between gender and project selection, we can work towards creating more inclusive and equitable environments that foster diverse contributions to OSS community. Moreover, we hope future work can be conducted to investigate the potential of OSS4SG projects to stimulate participation of underrepresented minorities in OSS community.

Table 6 summarizes the proportions of project selection by gender. These results further indicate that our results regarding OSS4SG project selection generalize regardless of gender.

4.1.5 OSS4SG Intervention Effect. Finally, we investigate whether our lightweight educational intervention (described in Section 3.2) impacts student project selection in terms of both direct proportions (are more OSS4SG projects chosen?) and specific topics. The four pre-intervention semesters include 350 reports, 24% of which are OSS4SG projects (77 OSS4SG vs. 273 Non-SG). The four post-interventions semesters have 632 reports, 29% of which are OSS4SG projects (183 OSS4SG vs. 448 Non-SG). This increase is statistically significant ($z = 4.4$, $p < 0.0001$, Cohen's $h = 0.38$). As shown in Figure 2, the intervention resulted in an increase in projects labeled as Education, Creativity and Entertainment and Environment and Nature, with the corresponding reduction coming almost entirely from Management and Productivity.

These results are important because OSS4SG can interest students from a variety of backgrounds and help with longer-term interest in computing (see Section 2.2). However, educators may be wary of heavyweight approaches that may be falsely perceived as "lowering standards" or "sacrificing rigor" (cf. [26]). The 5% increase in uptake from adding ten sentence of awareness intervention is thus a very promising concrete path forward.

> Student motivation and project topic differ significantly between OSS4SG and Non-SG projects. Additionally, our lightweight intervention was correlated with increased OSS4SG project selection. Such an intervention is thus a promising way for instructors to promote OSS4SG contribution.

4.2 Phase 2 — Contribution Experience

After students select projects, they work on those projects, which involves both the manipulation of software engineering artifacts (e.g., source code, documentation) and also interaction with other developers (e.g., via email, Discord or IRC chats, pull requests, etc.).

Table 7: Self-reported experiences from 1,361 students of contributing to OSS (including both Non-SG and OSS4SG).

Experience Category	Count	Proportion
Learning a lot	451	45.8%
Feeling stressed and challenged	409	41.6%
Being positive about this experience	371	37.7%
Observing friendliness and responsiveness	241	24.5%
Being a first-time contributor	206	20.9%
Experiencing organized project	157	16.0%
Finding it easy to participate	154	15.6%
Finding similarities with industry work	118	12.0%
Finding issue resolved	84	8.5%
Feeling nervous or fearful	74	7.5%
Recognizing the importance of testing	69	7.0%
Observing unfriendliness or apathy	61	6.2%
Experiencing disorganized project	59	6.0%
Continuing to contribute	49	5.0%
Engaging in pair programming	47	4.8%
Having a terrible experience	16	1.6%

Among the 1,361 students, the most commonly-reported experience was one of learning a lot and the finding the activity stressful and challenging, but ultimately positive. An indicate sentence from a report notes, "*our team gained quite a few insights into real-world software development practices. Overall, I enjoyed this project and was able to learn a lot about contributing to an open source project and was able to apply many of the skills I have learned in class.*" Around 42% students mentioned the activity to be more stressful and challenging than they originally expected. As stated by one student, "*I personally found it incredibly difficult being thrown into an already existing project with little guidance/background on how it worked. With every previous school assignment, we were given [...] a set of starter files or we started an assignment from scratch.*" Despite these challenges, about 38% of students explicitly stated that their attitude toward the contribution activity was positive, and only 1.6% found contributing to OSS to be terrible. These results reflect existing work that suggests incorporating contributions to OSS is generally well-received by students.

4.2.1 Interactions with OSS Developers. From the perspective of collaboration, as shown in Table 7, far more students perceived OSS to be friendly and responsive than unfriendly or not responsive (24.5% vs. 6.2%). One student contributing to `Zulip` noted, "*Communicating with other developers was very straightforward, especially with the incredibly active [project] channel. There were always opportunities to ask questions and get help. The culture of this open source community is rooted in encouraging input from all, no matter your skill level.*" We draw particular attention to the comment about skill level, which aligns with prior work on the appropriateness of OSS for all students, including those who might have less incoming preparation (e.g., first-generation or transfer students, etc.).

Table 7 also summarizes students' self-reported experiences of contributing to OSS. Around 4.8% of students explicitly described pair programming as particularly useful. Alternatively, among the

Table 8: Categories of advice offered by students to future students (including both Non-SG and OSS4SG).

Advice	Count	Proportion
Start Early	346	35.1%
Take the Assignment Seriously	305	31.0%
Choose a Good Project	174	17.6%
Practice for Future Career	143	14.5%
Communication and Collaboration	72	7.3%
Planning is Important	57	5.8%
Follow Coding Conventions	10	1.0%

379 teams for which we have reports, 12% of teams noted a positive pair programming interaction. A negative pair programming interaction experience was not an identified or present in this dataset. One team stated, "*We found it was more beneficial for both of us to instead utilize pair programming for all tasks, as we could bounce ideas off each other and work more efficiently.*" In addition, very few students who worked alone suggested that future students also work alone (see Section 4.2.2). Together, these reported experiences may suggest that students enjoy or benefit from the pair interactions when approaching a new software project, despite the communication overhead required in teamwork.

4.2.2 Advice from Student Contributors. We also summarize what lessons students reported learning from this experience in Table 8. Most students encouraged others to start early or to take the assignment seriously. However, beyond such general academic advice, over one-sixth of students specifically gave advice suggesting to choose a good project. This is particularly relevant to Phase 1 of the open source contribution lifecycle (Section 4.1.1). For example, advice to choose a good project may be reflected in the motivations students report (e.g., Useful or Helpful) when selecting a project as well as the projects' community and organization (e.g. well-organized code, clear issues, quality documentation, helpful community). Finally, 14.5% of the advice suggests that contributing to OSS serves as practice for a future software development career.

4.2.3 Differences in OSS4SG Projects. When comparing student experiences between Non-SG and OSS4SG, the only statistically-significant difference centered around the programming language(s) used in the projects. We find that JavaScript is more common in student-selected OSS4SG projects than is other languages. This is in contrast to Non-SG, where languages such as Python and C++ are more common in student-selected projects. JavaScript is not taught in introductory courses in the school in question.We hypothesize that social good projects to which students contribute provide services through websites or similar front-ends with straightforward UIs as indicated by our collected data, both because their audiences are often more general (e.g., for broad-reaching humanitarian benefits) and also because many OSS4SG projects explicitly focus on democratizing or publicizing information in an accessible way (e.g., making local governance records easier to view [31]). While decisions about which languages to cover in CS curricula are nuanced and multi-faceted [56], our findings do provide an additional angle by which languages may be evaluated, given the desire to admit OSS4SG interactions to appeal to broader audiences as well as impact the community's accessibility and inclusiveness. Moreover, this factor can contribute to the improvement of students' practical programming skills and equip them for their future professional paths. As for self-reported experiences, there were no statistically-significant differences between Non-SG and OSS4SG. However, there were significant differences in pull request acceptance rates (i.e., in outcomes), see Section 4.3.

> Students report learning from the experience (e.g., because they advise others to start early and practice for future career), and that it may have helped to improve skills such as time management. Other than programming language, there are no significant differences in the reported experiences of Non-SG and OSS4SG contributors. These results suggest that, regardless of whether students contribute to Non-SG or OSS4SG projects, valuable learning outcomes can be realized.

4.3 Phase 3 — Community Reception

In this last phase, we consider how student contributions are viewed by the OSS community. To do so, we primarily investigate the reception of the pull requests submitted by students as part of their projects (see Section 3.3 for more information on how we annotated and analyzed student pull requests).

We first investigate the percentage of pull requests submitted by students that were ultimately accepted and merged into the project. We find that 43.7% (432/989) of pull requests submitted by students are accepted and merged as of August 2022. While general statistics about contributions accepted on GitHub are not widely published, this ratio is significantly lower than the 85% observed in other studies of non-student-specific pull requests [27]. We discuss this discrepancy further in Section 5.2.

This overall percentage masks the large variation of acceptance rates on a project-to-project basis. While students submitted pull requests to 443 projects, only ten projects were submitted to at least 15 times. However, among those projects, pull request acceptance rates varied widely from 10.5% for `Zulip` a "*team collaboration tool*", to 78.9% for `Habitica`, a "*habit building program which treats your life like a Role Playing Game.*" Other popular software infrastructure projects, such as `VSCode` and `Pandas`, fall in the middle with 31.0% and 34.1% respectively. One possible explanation for this variability is that some projects may be larger and more actively maintained than others. However, at time of writing, all ten are large and active projects (regular commits in the last week, over 200 contributors), so other factors such as community culture or project set-up and contribution requirements are likely driving this diversity.

4.3.1 Other General Findings for Pull Requests. We also investigate whether students' pull request (PR) acceptance rates vary by project topic (see Section 3.3). Using a χ^2-test, we find the acceptance rate of pull requests varies significantly by project topic ($p = 0.01$, Cramer's $V = 0.21$). Furthermore, we also collected all the pull requests from all the GitHub projects involved in this study (i.e., 443 GitHub Projects, 1,900,681 Pull Requests, 1,363,533 Accepted Pull Requests) and found no significant difference on PR acceptance rate by project topics ($p = 0.23$) among all the PRs from general

contributors. To understand which factors drive the significant result among students' PRs, we analyze standardized residuals from the χ^2-test. We find that pull requests to Management and Productivity projects are significantly less likely to be accepted than expected, while pull requests to Creativity and Entertainment and Education projects are significantly more so. Specifically, 60.5% of Creativity and Entertainment and 52.2% of Education pull requests are accepted (17% and 9% above the average of 43.7%) while only 33.9% of Management and Productivity pull requests are accepted (10% below average). These results are particularly interesting as they pertain to the Creativity and Entertainment category, a category composed primarily of games. We hypothesize that online games often support a culture of modification and bug fixing that appreciates contributions from invested participants.

We also investigate factors about the students themselves that could influence the likelihood that their pull requests are accepted. In particular, we investigate correlations with student gender and whether the students worked in a team or alone. We do not find any evidence of significant differences in pull request acceptance rates by gender: 45.9% of pull requests submitted by men are accepted compared to 40.4% of pull requests submitted by women ($p = 0.22$).

We did, however, observe evidence that choosing to work on the pull request alone or with a partner *does* correlate with pull request acceptance: pull requests submitted by students who work alone are significantly more likely to be accepted than are those from partners ($p < 0.01$, Cohen's $h = 0.17$). 47.4% (266/561) of pull requests submitted by solo students were accepted, compared to only 38.8% (166/428) of pull request submitted by pairs. Initially, this result may appear to run contrary to established wisdom on pair programming [9]. However, due to the context of this being observed in student submissions, we hypothesize that this difference is the result of hidden communication costs between the partners; for students not trained in pair-programming, it may be these communication and time management difficulties outweigh the benefit from working with a partner, at least in the context of a semester-long software engineering course. We also note that student subjective experiences (Section 4.2) and advice (Section 4.2.2) both favor pair programming, suggesting that students prefer the experience, even if it is less effective for this one outcome. Given the importance placed on student retention, and thus a desire to avoid a perceived conflict between what is optimal and what students desire, this leads to the direct advice to not structure course grades or student assessments around whether or not pull requests are accepted (and instead leave them ungraded or as optional extra credit).

> 44% of student pull requests are accepted and merged. Students who work alone are *significantly more likely* to have their pull requests accepted than are those that work in pairs (47% to 39%, $p < 0.01$, Cohen's $h = 0.17$), a result that may add nuance to standard considerations of pair programming in an educational setting. Faculty and instructors are encouraged to avoid mandatory grading of whether pull requests are accepted.

4.3.2 Pull Requests and Social Good. Finally, we analyze how community reception of student pull requests differs for OSS4SG projects compared to Non-SG projects. We find a significant difference between the pull request acceptance rates. In particular, we find that pull requests for OSS4SG projects are 13% more likely to be accepted than those of Non-SG projects (53% or 145/276 vs. 40% or 287/713, $p < 0.001$, Cohen's $h = 0.25$). Furthermore, we calculated the overall acceptance rate of all PRs in all of the GitHub projects involved in this study (see Section 4.3.1). Our results show that there is no significant difference between the acceptance rates of Non-SG and SG projects in general on GitHub (70.6% and 72.4%, respectively) when using the same methodology. This is particularly interesting because it might hint that while there is no significant difference in general PR acceptance rate between non-SG and SG projects (e.g., we cannot assume it has lower standards in SG projects or SG projects are easier), students' contribution to SG projects might project a higher quality or effort compared to non-SG projects. This is also relevant in helping students evaluate the impact they may have when contributing to OSS projects in this setting — that they may feel more included by OSS4SG communities.

We find these results to be particularly compelling for two reasons. First, instructors of undergraduate software engineering courses can simultaneously raise awareness about OSS4SG projects and communities while indicating the increased likelihood that their contributions will be well-received and incorporated into such software projects. As a result, this encourages not only broader participation from students finding OSS4SG appealing, but also helps students learn more about the complete process in OSS projects. We anticipate that these results will help inform educators about best practices concerning the discussion of OSS4SG projects.

> Student pull requests for social good projects are *significantly more likely* to be accepted and merged than pull requests to Non-SG projects (53% to 40%, $p < 0.001$, Cohen's $h = 0.25$). This suggests that educators can truthfully indicate that their contributions are more likely to be well-received by OSS4SG communities to promote participation in such projects without undermining the educational value of using this intervention.

5 DISCUSSION

The findings in the previous section demonstrate that students differ greatly in their motivations when choosing whether to contribute to Non-SG or OSS4SG projects, as well as in the project topics they select. In addition, we find that external factors, such as interventions, can have an impact on student contribution choices. We also find a nuanced interaction with pair programming as well as a more direct benefit, in terms of pull request acceptance, for OSS4SG contributions. In this section, we will propose approaches for leading more future developers to participate in OSS4SG and helping them find the effective projects and topics for their contributions.

5.1 Leveraging External Guidance

In our four-year study, we find that a lightweight intervention (based on raising awareness without changing grading) increased student involvement in OSS4SG by over 5%. Although the community is still debating the exact criteria for "social good," we believe OSS projects that address people and society, broadly construed, would benefit from greater exposure and more participating contributors. We encourage instructors of similar classes not only to make use of such interventions (such as the one proposed here

or that discussed by Goldweber *et al.* [26]), but also to use third-party agencies (e.g., Ovio [2], DPGA [1], GitHub). For example, to improve the stability of contributions, we suggest discussing and linking resources like Ovio [2] and DPGA [1] in class alongside lecture materials for ethics and professionalism.

We find that 21% of student developers self-identified as first-time contributors to OSS. Thus, discussing OSS4SG communities in class and discussing resources like Ovio [2] and DPGA [1] can also have the added effect of awareness for these first-time contributors, ultimately raising the visibility of OSS4SG projects.

5.2 Improving Student Experiences and Community Reception

Student experiences will ultimately influence continued contributions to OSS4SG, so the community culture and organization is quite important. While we read and coded student reports, many first-time developers reported struggling to find the source code file required, or they reported that there was no clear set of contribution instructions for beginners. Therefore, we recommend organizers of OSS4SG projects be particularly vigilant about providing supportive scaffolding and documentation for first-time developers. In addition, in our dataset, many student developers mentioned that they would be more willing to keep contributing in the future if they were working with a more friendly and responsive culture.

While we noted a significant difference in pull request acceptance between Non-SG and OSS4SG projects, less than half of all requests were ultimately accepted. This rate is lower than that observed in non-student-specific studies of pull request acceptance [27]. This difference may be due in part to designing our pull request annotation methodology to align with an educational context rather than to catch all accepted contributions (see section 3.3). However, it is reasonable that pull requests from students for a course would have a lower acceptance rate than those of OSS developers at large. We encourage future investigation both in understanding what would make student pull requests more likely to be accepted, and in understanding why pull requests for OSS4SG projects are more accepted. Increasing the likelihood of accepted pull requests may contribute to a more positive outcome for student participants.

6 LIMITATIONS AND THREATS TO VALIDITY

Although we present a number of statistically-significant results from a dataset of over 1,300 students over eight semesters, our results may not generalize to all situations. We discuss a few such threats to validity. First, our four-year study was conducted in a classroom setting, so the results may not generalize to new developers in industry. We mitigated this by focusing on senior computer science students (i.e., who will likely enter the workforce shortly) engaging in an activity that is seen as more similar to industrial software engineering than are "toy examples" (although single contributions to GitHub may still not be a perfect match). Second, similar to previous studies, we lack an official principle or automated method to determine whether an OSS project is associated with social good or not [31]. We mitigated this threat by making use of both the Ovio [2] and DPGA [1] guidelines and follow best practices as described in Section 3.3, and by making use of multiple separate annotators (i.e., separate authors) when employing qualitative methods. Finally, the partial overlap of our intervention and the outbreak of COVID-19 is a complication (COVID-19 happened one semester before intervention was applied). To mitigate this, we analyzed students' project selections before and after COVID-19, finding that the pandemic's effect weakened over time: for example, the percentage of Health and Well being projects is 1–3% in the first three semesters, but jumps to 9% in the fourth semester (when COVID-19 occurred), then drops back to 5% in the fifth semester (when the intervention was delivered) and stays low at 1–3% again after that. We also calculated our statistics with the presence of COVID-19 as another independent variable potentially impacting the outcome, and still find that our intervention is statistically significant (even when COVID-19 is an explicit factor). These results give confidence that COVID-19 effects may be minimal compared to the intervention on students' contributions. Finally, we want to point it out that biases might exist in retrospective recall when assessing students' motivation [11, 47], which can also overlap with an individual's self-beliefs [49].

7 CONCLUSION

We present an analysis of 984 project reports submitted by 1,361 senior computer science students in a software engineering course. We investigated how we can train these future developers by exploring contributions to OSS and OSS4SG projects (in contrast to Non-SG projects). Our aim is to investigate effective strategies for fostering and supporting emerging contributors to OSS especially OSS4SG through educational contexts. We find significant differences in project selection ($p < 0.0001$, effect size = 0.84), reported student motivation ($p < 0.01$, effect size = 0.13), and reception from software communities. Students who work in pairs are more likely to choose projects that are Useful and Helpful compared to students working alone, and students report favorable experiences with pair programming activities. We also find women are more likely to select OSS4SG than men. Our lightweight intervention has a statistically-significant improvement in student social good selection ($p < 0.0001$, effect size = 0.38). In addition, students contributing to OSS4SG projects are significantly more likely to have contributions accepted by their communities. Combined with the fact that there are no significant differences in reported experiences between Non-SG and OSS4SG contributors, this suggests that we can raise awareness and appreciation for OSS4SG projects while retaining valuable educational outcomes. We believe our results can not only help improve the experience of participating in social good OSS but also improve the quality of new developers' contributions. We hope that educators, OSS project organizers, and contributors can form a mutually-beneficial framework for developing Open Source Software for Social Good communities in the future.

8 DATA AVAILABILITY

We make our intervention text, qualitative code books, and statistical scripts publicly available on Zenodo [3]. DOI 10.5281/zenodo.8264614 Our data, with potentially sensitive information (e.g., links to public pull requests that are connected to educational assignments and grades), can't be public due to IRB restrictions. Interested researchers can contact the first author for data sharing.

REFERENCES

[1] 2023. *Digital Public Goods Alliance.* https://ovio.org/

[2] 2023. *Ovio.* https://ovio.org/

[3] 2023. *Supplementary Material for A Four-Year Study of Student Contributions to OSS vs. OSS4SG with a Lightweight Intervention.* Zenodo. https://doi.org/10.5281/zenodo.8264614

[4] Mohammad AlMarzouq, Li Zheng, Guang Rong, and Varun Grover. 2005. Open source: Concepts, benefits, and challenges. *Communications of the Association for Information Systems* 16, 1 (2005), 37. https://doi.org/10.17705/1CAIS.01637

[5] Donald Bagert and Steve Chenoweth. 2005. Future growth of software engineering baccalaureate programs in the United States. In *2005 Annual Conference*. 10–653. https://doi.org/10.18260/1-2--14753

[6] Angela R Bielefeldt and Nathan E Canney. 2016. Humanitarian aspirations of engineering students: Differences between disciplines and institutions. *Journal of Humanitarian Engineering* 4, 1 (2016).

[7] Judith Bishop, Carlos Jensen, Walt Scacchi, and Arfon Smith. 2016. How to use open source software in education. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*. 321–322. https://doi.org/10.1145/2839509.2844665

[8] Kevin Buffardi. 2016. Localized open source software projects: Exploring realism and motivation. In *2016 11th International Conference on Computer Science & Education (ICCSE)*. IEEE, 382–387. https://doi.org/10.1109/ICCSE.2016.7581611

[9] Alistair Cockburn and Laurie Williams. 2000. The costs and benefits of pair programming. *Extreme programming examined* 8 (2000), 223–247.

[10] Kevin Crowston and James Howison. 2003. The social structure of open source software development teams. (2003).

[11] Susan E Cutler, Randy J Larson, and Scott C Bunce. 1996. Ropressive coping style and the experience and recall of emotion: A naturalistic study of daily affect. *Journal of Personality* 64, 2 (1996), 379–405. https://doi.org/10.1111/j.1467-6494.1996.tb00515.x

[12] Gregory DeKoenigsberg. 2008. How successful open source projects work, and how and why to introduce students to the open source world. In *2008 21st Conference on Software Engineering Education and Training*. IEEE, 274–276. https://doi.org/10.1109/CSEET.2008.42

[13] Blagoj Delipetrev, Andreja Jonoski, and Dimitri P Solomatine. 2014. Development of a web application for water resources based on open source software. *Computers & Geosciences* 62 (2014), 35–42. https://doi.org/10.1016/j.cageo.2013.09.012

[14] Sarah Sakha Durva Trivedi. 2019. Women in Tech: How They are Using Data and Tech for Social Good. https://www.rockefellerfoundation.org/blog/women-data-tech-social-good/, accessed 2022-08-30.

[15] H Ellis, Ralph A Morelli, and GW Hislop. 2008. Support for educating software engineers through humanitarian open source projects. In *2008 21st IEEE-CS conference on software engineering education and training workshop*. IEEE, 1–4. https://doi.org/10.1109/CSEETW.2008.5

[16] Heidi JC Ellis, Gregory W Hislop, Stoney Jackson, and Lori Postner. 2015. Team project experiences in humanitarian free and open source software (HFOSS). *ACM Transactions on Computing Education (TOCE)* 15, 4 (2015), 1–23. https://doi.org/10.1145/2684812

[17] Heidi JC Ellis, Gregory W Hislop, Stoney Jackson, and Lori Postner. 2015. Team project experiences in humanitarian free and open source software (HFOSS). *ACM Transactions on Computing Education (TOCE)* 15, 4 (2015), 1–23. https://doi.org/10.1145/2684812

[18] Madeline Endres, Madison Fansher, Priti Shah, and Westley Weimer. 2021. To read or to rotate? comparing the effects of technical reading training and spatial skills training on novice programming ability. In *ESEC/FSE '21: 29th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering, Athens, Greece, August 23-28, 2021*, Diomidis Spinellis, Georgios Gousios, Marsha Chechik, and Massimiliano Di Penta (Eds.). ACM, 754–766. https://doi.org/10.1145/3468264.3468583

[19] Maria Angela Ferrario, Will Simm, Peter Newman, Stephen Forshaw, and Jon Whittle. 2014. Software engineering for'social good': integrating action research, participatory design, and agile development. In *Companion Proceedings of the 36th International Conference on Software Engineering*. 520–523. https://doi.org/10.1145/2591062.2591121

[20] Casey Fiesler, Jed R Brubaker, Andrea Forte, Shion Guha, Nora McDonald, and Michael Muller. 2019. Qualitative methods for CSCW: Challenges and opportunities. In *Conference companion publication of the 2019 on computer supported cooperative work and social computing*. 455–460. https://doi.org/10.1145/3311957.3359428

[21] Douglas H Fisher, Jacqueline Cameron, Tamara Clegg, and Stephanie August. 2018. Integrating social good into CS education. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*. 130–131. https://doi.org/10.1145/3159450.3159622

[22] Brian Fitzgerald. 2004. A critical look at open source. *Computer* 37, 7 (2004), 92–94. https://doi.org/10.1109/MC.2004.38

[23] Brian Fitzgerald. 2005. Has open source software a future. *Perspectives on free and open source software* 1 (2005), 93–106.

[24] Marco Gerosa, Igor Wiese, Bianca Trinkenreich, Georg Link, Gregorio Robles, Christoph Treude, Igor Steinmacher, and Anita Sarma. 2021. The shifting sands of motivation: Revisiting what drives contributors in open source. In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*. IEEE, 1046–1058. https://doi.org/10.1109/ICSE43902.2021.00098

[25] Carlo Ghezzi and Dino Mandrioli. 2005. The challenges of software engineering education. In *International Conference on Software Engineering*. Springer, 115–127. https://doi.org/10.1145/1062455.1062578

[26] Michael Goldweber, John Barr, Tony Clear, Renzo Davoli, Samuel Mann, Elizabeth Patitsas, and Scott Portnoff. 2013. A framework for enhancing the social good in computing education: a values approach. *ACM Inroads* 4, 1 (2013), 58–79. https://doi.org/10.1145/2432596.2432616

[27] Georgios Gousios, Martin Pinzger, and Arie van Deursen. 2014. An exploratory study of the pull-based software development model. In *Proceedings of the 36th international conference on software engineering*. 345–355. https://doi.org/10.1145/2568225.2568260

[28] Gregory W Hislop and Heidi JC Ellis. 2017. Humanitarian Open Source Software in Computing Education. *Computer* 50, 10 (2017), 98–101. https://doi.org/10.1109/MC.2017.3641643

[29] Reid Holmes, Meghan Allen, and Michelle Craig. 2018. Dimensions of experientialism for software engineering education. In *Proceedings of the 40th International Conference on Software Engineering: Software Engineering Education and Training*. 31–39. https://doi.org/10.1145/3183377.3183380

[30] Zhewei Hu, Yang Song, and Edward F Gehringer. 2018. Open-source software in class: students' common mistakes. In *Proceedings of the 40th International Conference on Software Engineering: Software Engineering Education and Training*. 40–48. https://doi.org/10.1145/3183377.3183394

[31] Yu Huang, Denae Ford, and Thomas Zimmermann. 2021. Leaving My Fingerprints: Motivations and Challenges of Contributing to OSS for Social Good. In *43rd IEEE/ACM International Conference on Software Engineering, ICSE 2021, 22-30 May 2021*. IEEE, Madrid, Spain, 1020–1032. https://doi.org/10.1109/ICSE43902.2021.00096

[32] Eirini Kalliamvakou, Georgios Gousios, Kelly Blincoe, Leif Singer, Daniel M German, and Daniela Damian. 2014. The promises and perils of mining github. In *Proceedings of the 11th working conference on mining software repositories*. 92–101. https://doi.org/10.1145/2597073.2597074

[33] Murtaza Ali Khan and Faizan UrRehman. 2012. Free and open source software: Evolution, benefits and characteristics. *International Journal of Emerging Trends & technology in Computer Science* 1, 3 (2012), 1–7.

[34] Antti Knutas, Victoria Palacin, Giovanni Maccani, Pablo Aragon, Annika Wolff, and Lukas Mocek. 2022. Civic Code for Social Change: Lessons in Civic Tech Grassroots for Software Engineers. *IEEE Software* (2022). https://doi.org/10.1109/MS.2022.3179670

[35] Shaheen E Lakhan and Kavita Jhunjhunwala. 2008. Open source software in education. *Educause Quarterly* 31, 2 (2008), 32.

[36] Karim R Lakhani and Robert G Wolf. 2003. Why hackers do what they do: Understanding motivation and effort in free/open source software projects. *Open Source Software Projects (September 2003)* (2003). https://doi.org/10.2139/ssrn.443040

[37] Valentina Lenarduzzi, Vili Nikkola, Nyyti Saarimäki, and Davide Taibi. 2021. Does code quality affect pull request acceptance? an empirical study. *Journal of Systems and Software* 171 (2021), 110806. https:

//doi.org/10.1016/j.jss.2020.110806
[38] Mark W Lipsey and David S Cordray. 2000. Evaluation methods for social intervention. *Annual review of psychology* 51, 1 (2000), 345–375. https://doi.org/10.1146/annurev.psych.51.1.345
[39] Ju Long. 2009. Open Source Software Development Experiences on the Students' Resumes: Do They Count?-Insights from the Employers' Perspectives. *Journal of Information Technology Education: Research* 8, 1 (2009), 229–242. https://doi.org/10.28945/618
[40] Stephanie M Mazerolle and Sarah S Benes. 2014. Factors influencing senior athletic training students' preparedness to enter the workforce. *Athletic Training Education Journal* 9, 1 (2014), 5–11. https://doi.org/10.4085/09015
[41] Gerald Midgley. 2000. Systemic intervention. In *Systemic intervention*. Springer, 113–133. https://doi.org/10.1007/978-1-4615-4201-8_6
[42] United States White House Office of Science and Technology Policy. 2022. OSTP Issues Guidance to Make Federally Funded Research Freely Available Without Delay. https://www.whitehouse.gov/ostp/news-updates/2022/08/25/ostp-issues-guidance-to-make-federally-funded-research-freely-available-without-delay/, accessed 2022-08-31.
[43] Gustavo Pinto, Clarice Ferreira, Cleice Souza, Igor Steinmacher, and Paulo Meirelles. 2019. Training software engineers using open-source software: the students' perspective. In *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)*. IEEE, 147–157. https://doi.org/10.1109/ICSE-SEET.2019.00024
[44] Lori Postner, Darci Burdge, Stoney Jackson, Heidi Ellis, George Hislop, and Sean Goggins. 2015. Using humanitarian free and open source software (HFOSS) to introduce computing for the social good. *Acm Sigcas Computers and Society* 45, 2 (2015), 35–35. https://doi.org/10.1145/2809957.2809967
[45] United Nations Development Programme. [n. d.]. Sustainable Development Goals. https://www.undp.org/content/undp/en/home/sustainable-development-goals.html, accessed 2022-08-01.
[46] Huilian Sophie Qiu, Alexander Nolte, Anita Brown, Alexander Serebrenik, and Bogdan Vasilescu. 2019. Going farther together: The impact of social capital on sustained participation in open source. In *2019 ieee/acm 41st international conference on software engineering (icse)*. IEEE, 688–699. https://doi.org/10.1109/ICSE.2019.00078
[47] Michael Ross. 1989. Relation of implicit theories to the construction of personal histories. *Psychological review* 96, 2 (1989), 341. https://doi.org/10.1037/0033-295X.96.2.341
[48] Larissa Salerno, Simone de França Tonhão, Igor Steinmacher, and Christoph Treude. 2023. Barriers and Self-Efficacy: A Large-Scale Study on the Impact of OSS Courses on Student Perceptions. *arXiv preprint arXiv:2304.14628* (2023). https://doi.org/10.1145/3587102.3588789
[49] Christie N Scollon, Chu Kim-Prieto, and Ed Diener. 2003. Experience sampling: Promises and pitfalls, strengths and weaknesses. *Journal of Happiness studies* 4, 1 (2003), 5–34. https://doi.org/10.1023/A:1023605205115
[50] Jefferson Silva, Igor Wiese, Daniel M German, Christoph Treude, Marco Aurélio Gerosa, and Igor Steinmacher. 2020. A theory of the engagement in open source projects via summer of code programs. In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 421–431. https://doi.org/10.1145/3368089.3409724
[51] Sulayman K Sowe and Ioannis G Stamelos. 2007. Involving software engineering students in open source software projects: Experiences from a pilot study. *Journal of Information Systems Education* 18, 4 (2007), 425. https://www.proquest.com/scholarly-journals/involving-software-engineering-students-open/docview/200102291/se-2
[52] Igor Steinmacher, Tayana Conte, Marco Aurélio Gerosa, and David Redmiles. 2015. Social barriers faced by newcomers placing their first contribution in open source software projects. In *Proceedings of the 18th ACM conference on Computer supported cooperative work & social computing*. 1379–1392. https://doi.org/10.1145/2675133.2675215
[53] Josh Terrell, Andrew Kofink, Justin Middleton, Clarissa Rainear, Emerson R Murphy-Hill, and Chris Parnin. 2016. Gender bias in open source: Pull request acceptance of women versus men. *PeerJ Prepr.* 4 (2016), e1733. https://doi.org/10.7287/peerj.preprints.1733v1
[54] Bianca Trinkenreich, Igor Wiese, Anita Sarma, Marco Gerosa, and Igor Steinmacher. 2022. Women's Participation in Open Source Software: A Survey of the Literature. *ACM Trans. Softw. Eng. Methodol.* 31, 4, Article 81 (aug 2022), 37 pages. https://doi.org/10.1145/3510460
[55] Kimberly Truong. 2022. Let's Talk Open-Source—An Analysis of Conference Talks and Community Dynamics. In *2022 IEEE/ACM 44th International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*. IEEE, 322–324. https://doi.org/10.1145/3510454.3522683
[56] David A Watt. 2000. Programming languages-Trends in education. In *Proceedings of Simposio Brasileiro de Linguagens de Programacao, Recife, Brazil, http://www. dcs. gla. ac. uk/~ daw/publications/PLTE. ps.* Citeseer.

Received 2023-02-02; accepted 2023-07-27