OSSDoorway: A Gamified Environment to Scaffold Student Contributions to Open Source Software

Italo Santos*, Katia Romero Felizardo†, Anita Sarma‡, Igor Steinmacher* and Marco A. Gerosa*

*Northern Arizona University, Flagstaff, AZ, USA
†Federal Technological University of Paraná, PR, Brazil
‡Oregon State University, Corvallis, OR, USA
Email: italo.santos@nau.edu, katiascannavino@utfpr.edu.br, anita.sarma@oregonstate.edu
igor.steinmacher@nau.edu, marco.gerosa@nau.edu

Abstract—Software engineering courses enable practical learning through assignments requiring contributions to open source software (OSS), allowing students to experience real-world projects, collaborate with global communities, and develop skills and competencies required to succeed in the tech industry. Learning software engineering through open source contribution integrates theory with hands-on practice, as students tackle real challenges in collaborative environments. However, students often struggle to contribute to OSS projects and do not understand the contribution process. Research has demonstrated that strategically incorporating game elements can promote student learning and engagement. This paper proposes and evaluates OSSDoorway, a tool designed to guide students contributing to OSS projects. We recruited 29 students and administered a self-efficacy questionnaire before and after their use of OSSDoorway, along with qualitative feedback to assess challenges, interface features, and suggestions for improvement. The results show that OSSDoorway boosts students' self-efficacy and provides a structured, gamified learning experience. Clear instructions, real-time feedback, and the quest-based system helped students navigate tasks like using GitHub features to submit pull requests and collaborating with the community. Our findings suggest that providing students with a supportive gamified environment that uses feedback and structured quests can help them navigate the OSS contribution process.

keywords: open source software, gamification, software engineering education.

I. Introduction

Teaching software engineering (SE) is challenging [1], [2]. Traditional SE courses tend to emphasize theoretical methodologies and concepts, offering a limited focus on preparing students to work with real-world, complex software systems [3], [4]. To better align education with industry practices, SE courses should go beyond teaching concepts, methods, and techniques to include practical skills and attitudes that reflect the current software development landscape [5]. One approach that bridges this gap is integrating student participation in Open Source Software (OSS) projects as part of the SE curriculum [5], [6]. OSS projects foster collaborative environments where a community works together to develop software systems, providing a real-world context for learning.

Supporting students to contribute to OSS projects helps to prepare the future SE workforce [6], [7], [8]. By contributing to OSS, students at the beginning of their careers gain practical experience in both technical and soft skills, enhancing their

confidence when pursuing industry positions [5], [8], [9], [10]. Successful contributions to OSS projects raise students' visibility among peers [11], [12] and help secure jobs [13], [14]. Additionally, OSS contributions serve society beyond personal and project-level benefits by improving widely-used software [12], [15], [16].

However, contributing to OSS is not easy, and students usually feel discouraged by various barriers that hinder their participation [6], [17], [18], [19], [20], [21], [22], and need orientation. Gamification is a promising strategy for engaging students by incorporating game elements into educational systems to enhance motivation [23]. By leveraging gamification, students are more likely to stay committed, enjoy their contributions, and gain educational benefits [24], [25], [26]. For instance, Diniz et al. [27] implemented gamification strategies in GitLab to boost motivation and collaboration among undergraduate students contributing to OSS projects. Their findings indicate that game elements like quests and points help sustain engagement while guiding and motivating students to participate. In this paper, we designed and evaluated OSSDoorway, a tool to support students' contributions to OSS projects by leveraging a gamified environment to scaffold students' contributions. We define scaffolding following the description in [28] as the support provided by a teacher, parent, peer, or a computer- or paper-based tool, enabling students to engage in and develop proficiency in a task they would otherwise be unable to complete independently. The following research question guided our research:

> **Research Question**
>
> How does OSSDoorway support students in contributing to OSS projects?

To address this research question, we followed two steps. In **Step 1, OSSDoorway Design and Development**, we conducted requirements elicitation, followed by the tools design, formative studies, and expert evaluations through an iterative process. Moving to **Step 2, Summative Evaluation**, we analyzed the data of 29 students who took the Open Source Software Development course and underwent a self-efficacy assessment before and after using OSSDoorway. We requested students' feedback, asking them to identify challenges they

faced during quests, a key gamification element of OSSDoorway, point out the features they found most useful, and suggest improvements for the tool.

This paper makes three primary contributions: (1) OSSDoorway, a gamified tool designed to support students in contributing to OSS projects; (2) an empirical evaluation of OSSDoorway, demonstrating its impact on enhancing students' self-efficacy and ability to complete OSS-related quests; and (3) insights into students' perceptions of the proposed approach, highlighting the benefits and challenges they experienced and suggesting areas for improvement.

Our findings indicate that OSSDoorway supports students in contributing to OSS projects by enhancing their self-efficacy and offering a structured, gamified learning experience. OSSDoorway was particularly effective for women, who initially demonstrated lower levels of self-efficacy than their men counterparts. After using the tool, the difference in self-efficacy was no longer statistically significant. The integration of clear instructions, real-time feedback, and a quest-based system helped students successfully navigate complex tasks, such as using GitHub features, submitting pull requests, and collaborating with the community. These results indicate that OSSDoorway is a valuable tool for empowering students to contribute to OSS projects and evidence the effectiveness of gamification in this context.

II. BACKGROUND AND RELATED WORK

This section outlines the background and related work involving gamification and its use in SE education.

Gamification. Dichev and Dicheva [23] describe gamification as a strategy to enhance student motivation in educational settings by integrating game elements. Gamification is increasingly being utilized in education to enhance the learning process [29], [30]. Its key benefits include simplifying complex topics [31], [32]. For example, Deterding et al. [29] propose that gamification simplifies complex material, making learning more accessible and promoting deeper understanding. They argue that games can enhance educational experiences by making them more interactive and effective for students. Toda et al. [33] proposed a framework for gamification strategies in educational settings based on a literature review and an evaluation with gamification and education experts. The resulting taxonomy included the description of 21 game elements, which we used as a baseline for the design of OSSDoorway game elements.

Gamification for SE education. Gamification has been effectively applied to SE education, including agile process [34], software testing [35], [36], [37], design pattern [26], software project management [38], [39], and business processes [40]. These applications highlight the broad potential of gamification to enhance educational experiences in the field. Su [41] developed a gamified framework to assess the impact of gamification on teaching SE, finding that gamified methods increased student motivation and improved academic performance. Similarly, Sheth et al. [42] showed that incorporating gamification into a SE course enhanced student engagement

in areas such as documentation, bug reporting, and testing. In the context of OSS, gamification strategies have been used to encourage contributions, such as through the OpenRank network algorithm and a monthly contribution leaderboard [43]. Santos et al. [44] identified software solutions that facilitate the onboarding of newcomers in software projects. Among these solutions, some used gamification techniques with newcomers, fostering engagement and boosting motivation [27], [45]. Diniz et al. [27] implemented four game elements, i.e., quests, points, ranking, and levels in GitLab, and assessed their ability to motivate students to overcome orientation barriers. Although their approach shares similarities with this study, our work differs by conducting a systematic user-centered design of a more comprehensive gamified environment to support students' contributions in GitHub, employing a bot that interacts with students and updates the environment.

III. METHOD

This section outlines our approach to developing and evaluating the OSSDoorway tool. We divided the process into two steps: (i) OSSDoorway design and development and (ii) Summative evaluation, as observed in Figure 1.

Step 1 – OSSDoorway design and development. First, we carried out the requirements elicitation through a survey to gather student perceptions of game elements to be implemented in OSSDoorway (published elsewhere [46]). Then, we conducted interviews to further validate our findings and select the game elements to be implemented. Afterward, we followed an iterative user-centric design process that involved designing the tool, conducting formative evaluations, and assessing the proposed tasks with experts, i.e., instructors with experience teaching about OSS. In this process, we created and refined paper prototypes, which evolved into high-fidelity prototypes. Subsequently, we developed the actual tool. More details in Section IV.

Step 2 – Summative evaluation. We conducted an evaluation with 29 undergraduate students to assess OSSDoorway. In the study, students completed three different quests, each consisting of 3 to 5 different tasks. We administered a self-efficacy questionnaire before and after the students used OSSDoorway. The pre- and post-questionnaire responses were compared using the Wilcoxon signed-rank test [47]. At the end of the study, participants were asked to describe any difficulties they encountered while completing tasks, what aspects of the interface were most helpful, and provide suggestions for improving OSSDoorway. We qualitatively analyzed participants' responses to the open-ended questions using open coding procedures [48]. Section V provides additional details about this process.

IV. OSSDOORWAY DESIGN AND DEVELOPMENT

This section describes the OSSDoorway design and development process, including the requirements elicitation, tool design, formative studies, and expert evaluation.
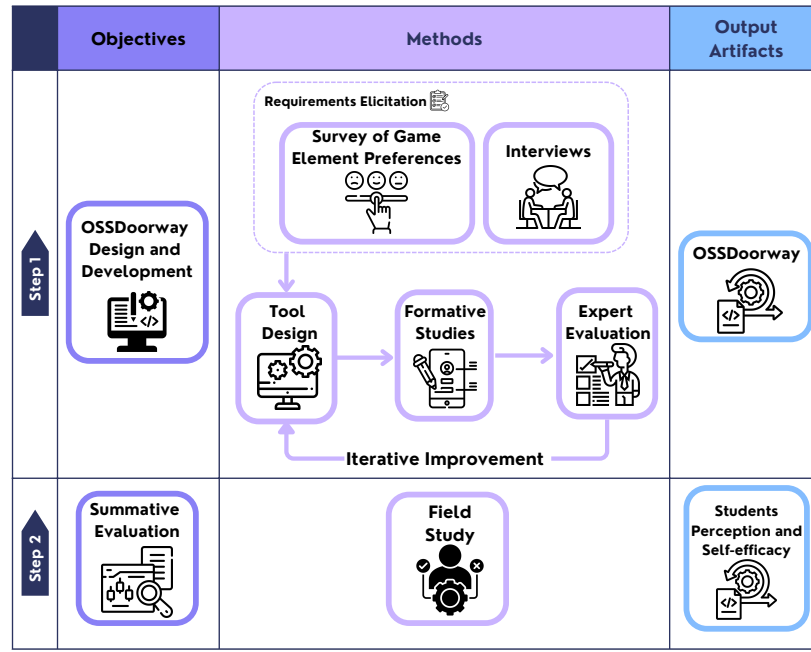
Figure 1. Method overview.

A. Requirements elicitation

In previous work [46], we surveyed 115 computer science students to identify their preferences for various game elements extracted from the taxonomy proposed by Toda et al. [33]. The survey was designed to provide insights from a user-centered design perspective before implementing the tool. In the survey, we tailored the description of the game elements within the context of the OSS contribution process.

The findings revealed that performance-oriented game elements, such as *quests, stats, maps, levels, points, progress bars,* and *badges*, were favored according to students. These elements offer valuable feedback on performance and foster a sense of progression and achievement. We paid particular attention to how perceptions of game elements varied across demographic groups, and we found no significant difference. Further details on this study can be found in [46].

After this preliminary work, we recruited eight computer science students for observation sessions. Participants in the observation sessions are designated as OB<X>, where X represents their assigned participant number. The participants comprised four undergraduate and four graduate students (3 women and 5 men). With their consent, the sessions were audio-recorded for note-taking purposes. During the sessions, we observed how the participants interacted with GitHub while performing two key tasks: (i) editing the Readme file and submitting a pull request and (ii) selecting and solving a non-code contribution task. These tasks were chosen because they exercise part of the process to contribute to an OSS project [21].

We employed the think-aloud method [49], where participants openly verbalized their decisions and reasoning. This allowed us to ask follow-up questions during the observation to clarify their decision-making process, providing valuable insight into how GitHub facilitates student contributions to OSS projects. Each session lasted approximately one hour.

During the interviews, we asked participants about the possibility of using game elements to support and contribute to a project on GitHub. Overall, the interviewees recommended including the same game elements identified in the survey while highlighting additional important elements, such as streaks. For instance, OB1 mentioned, "*I like quests a lot because they are fun, and I expect the system to have streaks to keep me engaged*", and OB4 noted, "*I would like a streak feature in the environment to give more points to players who access the system every day*". OB3 also emphasized, "*I like how a gamified environment breaks complex tasks into simpler steps through quests*".

Summary of Requirements Elicitation

After the survey and the interviews, we identified the following features that OSSDoorway needs to provide:
OSSDoorway features:

- Implement game elements that provide performance-oriented feedback: *stats, maps, levels, points, progress bars,* and *badges*.
- Implement streaks that allow users to earn additional rewards.
- Implement quests to support students in accomplishing the contribution goals.

B. Tool design

OSSDoorway was designed to guide students through a series of quests, each aligned with specific OSS contribution goals to ensure students gain practical knowledge and skills in contributing to OSS projects. We established three contribution goals based on previous research on the barriers that newcomers face when contributing to OSS projects, with emphasis on the key stages of the contribution process [21]. Through these quests, students should be able to (i) understand the GitHub contribution process, (ii) interact with other project members, and (iii) make a contribution.

To address goal *(i) understand the GitHub contribution process*, we designed **Quest 1: Exploring the GitHub World**. This quest familiarizes students with GitHub essential features by guiding them through the following tasks:

- Task 1 – Explore the issue tracker;
- Task 2 – Explore the pull-request;
- Task 3 – Explore the fork;
- Task 4 – Explore the README file;
- Task 5 – Explore the project contributors;

To address *goal (ii) interact with other project members*, we designed **Quest 2: Introducing Yourself to the Community**, which focus on guiding students on how to communicate with project contributors, as described below:

- Task 1 – Choose an issue to work on;
- Task 2 – Assign your user[id] to work on the issue;
- Task 3 – Post a comment in the issue introducing yourself;
- Task 4 – Mention a contributor to help you to solve the issue;

Finally, for the third contribution goal, *(iii) make a contribution*, we designed **Quest 3: Making Your First Contribution**. This quest involves students completing a non-code contribution, as described below:

- Task 1 – Solve the Issue (non-code contribution) and Submit a Pull Request;
- Task 2 – Post in the issue asking for someone to review it;
- Task 3 – Close the issue;

Through this process, students gain hands-on experience with the contribution workflow in GitHub. For the first version of OSSDoorway, we concentrated on non-code contributions to accommodate users without a technical background and to focus on the process and environment challenges, which are common across projects. The literature has shown that non-code contributions are a great way to join OSS [50]. We plan to expand OSSDoorway to include code review and refactoring quests.

Tool description. The students interact with the sandbox GitHub repository, where they can monitor their progress through the GitHub README, which serves as the "home" page, displaying all available tasks and user progress. Students develop and test their contributions in isolated sandbox environments, providing a safe learning space to experiment, make mistakes, and refine their code without impacting the original OSS projects. Task completion is guided by quest instructions provided in GitHub Issues, where the system facilitates interaction and provides feedback through a bot implemented to check students' responses (OSS Bot). The OSS Bot verifies the student's actions, updates the sandbox

repository, and records progress in a MongoDB database. These updates trigger changes that are continuously reflected on the front page while generating feedback for students and collecting evaluation data to assess the tool's effectiveness. In addition to working within the sandbox environment, the student can interact with a public OSS repository, contributing to real-world OSS projects and gaining hands-on experience. The OSS bot communicates with the user through two GitHub pages: the main repository page, where the README is visible, and GitHub Issues, where the quest interactions occur. OSSDoorway has been developed with a NodeJS [51] backend that utilizes the Probot [52] framework to create a bot for GitHub, making the solution more approachable for managing and automating GitHub tasks. Figure 2 illustrates the key components of the OSSDoorway tool and their interactions.
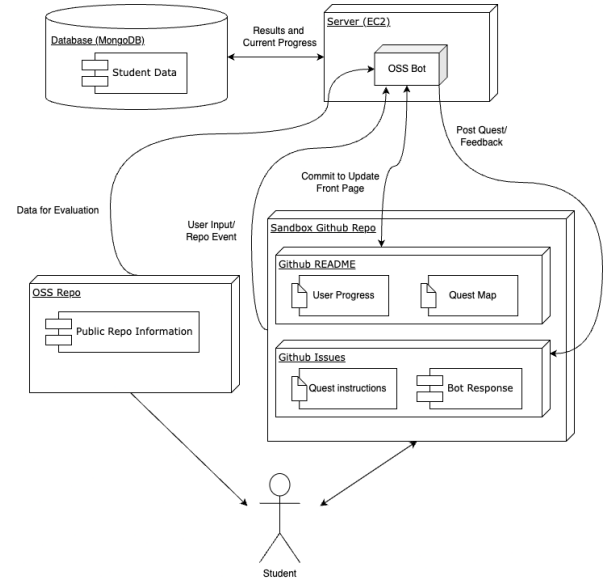

Figure 2. OSSDoorway Design Overview.

The interface of the OSSDoorway tool enables students to track their progress, view available quests and tasks, and interact with game elements such as progress bars, experience points, levels, badges, and streaks, as shown in Figure 3. In this figure, the task description for Quest 1 is displayed, where students are instructed to explore the issue tracker. Finally, the bot's feedback is presented, and the student points are awarded upon successful task completion.

OSSDoorway gamification elements. OSSDoorway implements eight game elements previously identified in the requirement elicitation (Section IV-A). Students can track their *Quests completed* while earning *Experience points* based on task difficulty. As they accumulate points, they *Level up*, unlocking new quests and rewards. *Badges* are awarded for completing quests and maintaining *Streaks*, earned by completing three tasks in a row. The tool also provides a snapshot of *Student stats*, summarizing completed quests, total points, and current level. At the same time, a *Map* helps students visualize their journey, and a *Progress bar* shows students' progress.
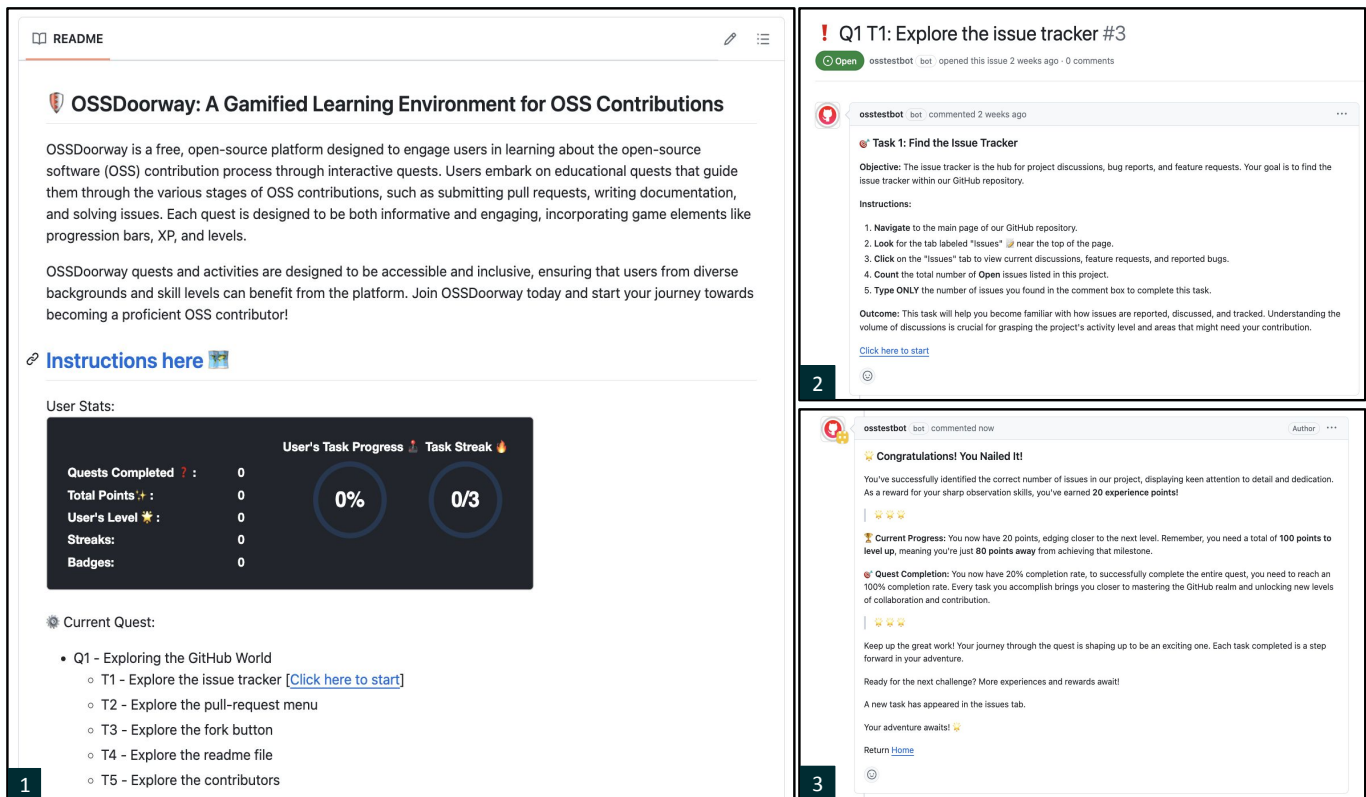
Figure 3. The left side (1) displays the OSSDoorway README page, where students can track their progress and view available tasks along with game elements such as progress bars, experience points, levels, badges, and streaks. The top right side (2) shows the GitHub Issues page with detailed instructions for Task 1. The bottom right side (3) presents feedback from the OSS Bot after the student successfully completes the task, providing progress updates and awarding experience points.

Moreover, OSSDoorway includes a correctness check through the integrated OSS Bot, which assesses whether students have completed a task and are ready to proceed to the next one.

To make OSSDoorway accessible for users with different cognitive styles, we employed directives suggested by Burnett et al. [53]. For example, we emphasized the practical impact of completing tasks, appealing to purpose-driven users while providing enjoyment for those *motivated* by technology. We considered comprehensive and selective *information processing styles*, offering detailed instructions for those who prefer step-by-step guidance while allowing others to explore the GitHub environment more flexibly. To support users with lower *computer self-efficacy*, OSSDoorway provides encouraging feedback and achievable tasks. It also addresses *risk aversion* by allowing students to experiment with low-stakes tasks, guided by the OSS Bot, which ensures mistakes can be corrected. Finally, OSSDoorway supports structured, process-oriented *learning style* as well as tinkering, offering a flexible environment with clear instructions.

C. Formative studies

OSSDoorway prototypes. We created paper prototypes of OSSDoorway and conducted two evaluation rounds. Participants in the prototype sessions are designated as PS<X>, where X represents their assigned participant number. With their consent, the sessions were audio-recorded for note-taking purposes.

During the first round, the group consisted of one undergraduate student, two graduate students, and one industry professional, all men. Overall, participants highlighted the need for clearer instructions to navigate the environment. PS1 emphasized that clear explanations for each game element were essential for understanding OSSDoorway. PS3 noted that some quests were unclear and required rewording or more detailed descriptions to clarify user actions. PS2 found the navigation confusing and suggested adding visual cues for better guidance. Based on this feedback, we revised the prototype by adding instructional guidelines on interacting with OSSDoorway and improving task descriptions to provide feedback.

During the second round of tests, the group was composed of four graduate students (one woman and three men). Participants indicated more satisfaction with the prototype. They suggested that they would like to see some competition elements in the environment. Moreover, PS5, PS7, and PS8 indicated that competitive aspects, such as leaderboards or rewards, could significantly enhance motivation. Specifically, PS3 and PS7 desired a more competitive system, suggesting rewards like GitCoins or leaderboards similar to those found in Duolingo, a popular gamified language-learning app. Further-

more, P08 also suggested real-world rewards as an incentive.

After refining these two paper prototypes, we developed a high-fidelity prototype, as shown in Figure 4. The paper prototypes are available in the supplemental material [54]. In the high-fidelity prototype, we incorporated all participant suggestions, except for the competitive elements, which were excluded based on survey results indicating that competition and pressure were viewed less favorably [46].
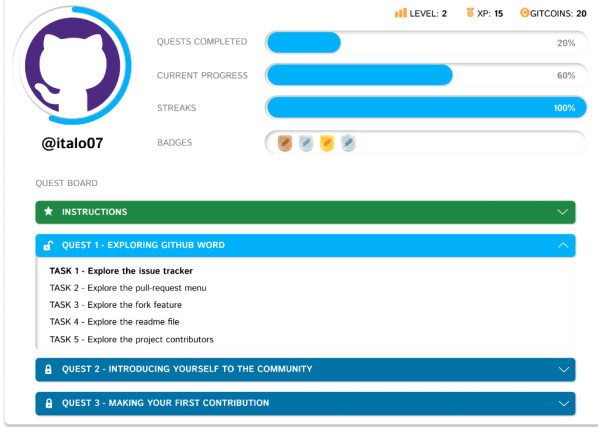

Figure 4. OSSDoorway Page - High Fidelity Prototype.

D. Expert evaluation

During the development of OSSDoorway, we evaluated the quests included in OSSDoorway to ensure that they align with our intended three contribution goals defined previously in Subsec. IV-A. To assess how well the proposed quests matched the contribution goals, we interviewed twelve experts with experience teaching OSS in their computer science curriculum. Among the 12 participants, all hold doctoral degrees and have significant experience in SE and OSS development, with some specializing in Human-Computer Interaction (HCI) and Artificial Intelligence (AI). Over half of the participants have more than 20 years of teaching experience. Their experience with OSS started as early as 2001. Seven out of twelve participants are full professors at well-known universities, teaching courses related to OSS development, software engineering, and capstone projects.

Overall, the experts' feedback was highly positive. Several participants noted that the quests closely mirrored the tasks they already assign to students in their courses, reinforcing that the *contribution goals are aligned with education goals* in the classroom and that the proposed quests support students in contributing to OSS projects. Experts also offered suggestions for improvement, such as reordering the tasks in quest 1 to place the exploration of the README file first, and including quests that require technical knowledge. We plan to incorporate this feedback in a second development cycle, adding more advanced quests to OSSDoorway.

V. SUMMATIVE EVALUATION

We adopted OSSDoorway over two editions of an OSS development course at a US university, each led by a different in-

structor and attended by master's students. Our study evaluated how OSSDoorway supports students in contributing to OSS projects. We administered a self-efficacy questionnaire before and after the students used the platform, using a 5-point Likert scale for each question. Additionally, we gathered qualitative feedback at the end of the study by asking participants to describe any challenges they faced, the most helpful aspects of the interface, and their suggestions for improvement.

Pilot. We conducted a pilot study with two researchers to gather feedback on the field study instruments (questionnaires) and overall design. We revised the questionnaires based on the feedback received, including removing certain questions to reduce the length of the study.

Participants. We applied OSSDoorway in the classroom, in two different semesters. Out of the 37 students who took the courses and used the environment, 29 agreed to have the data analyzed. The students ranged from 20 to 25 years old; 13 were women, and 16 were men. All students were enrolled in a Master's degree program in Information Technology or Computer Science. We asked participants about their experience with GitHub and OSS. While some mentioned having used GitHub before, upon further inquiry, it became clear that their interaction was limited to account creation without actual project contributions. Therefore, none had prior experience contributing to OSS and using GitHub.

Data analysis. *1) Likert scale questions:* We asked the level of agreement of participants using Likert scale questions that run from *strongly disagree* (encoded as 1) to *strongly agree* (encoded as 5), to rate their self-perception about their ability to contribute to OSS projects. The questionnaire was based on the work of Bandura [55] and had seven questions. Participants answered those questions before and after the experiment. The goal was to capture the students' self-perceived efficacy. The questions were prefixed with "I am confident that I can:" followed by:

(i) . . . use GitHub issue tracker to find open issues;
(ii) . . . understand GitHub pull-requests;
(iii) . . . fork GitHub repositories;
(iv) . . . find someone to help me with an issue using the GitHub web interface;
(v) . . . open a pull request using GitHub web interface;
(vi) . . . find an issue to work on and assign it to me;
(vii) . . . use GitHub to contribute to projects.

2) Open questions: We used open questions to ask participants to describe difficulties they encountered while completing tasks, what aspects of the interface were most helpful, and to provide suggestions for improving OSSDoorway. We qualitatively analyzed participants' comments following open coding procedures [48]. The process was conducted using continuous comparison and discussion until reaching a consensus. Two researchers jointly analyzed the sets of answers to establish common ground, discussing the applied codes and disagreements until reaching a consensus. Finally, a third researcher inspected the classification.
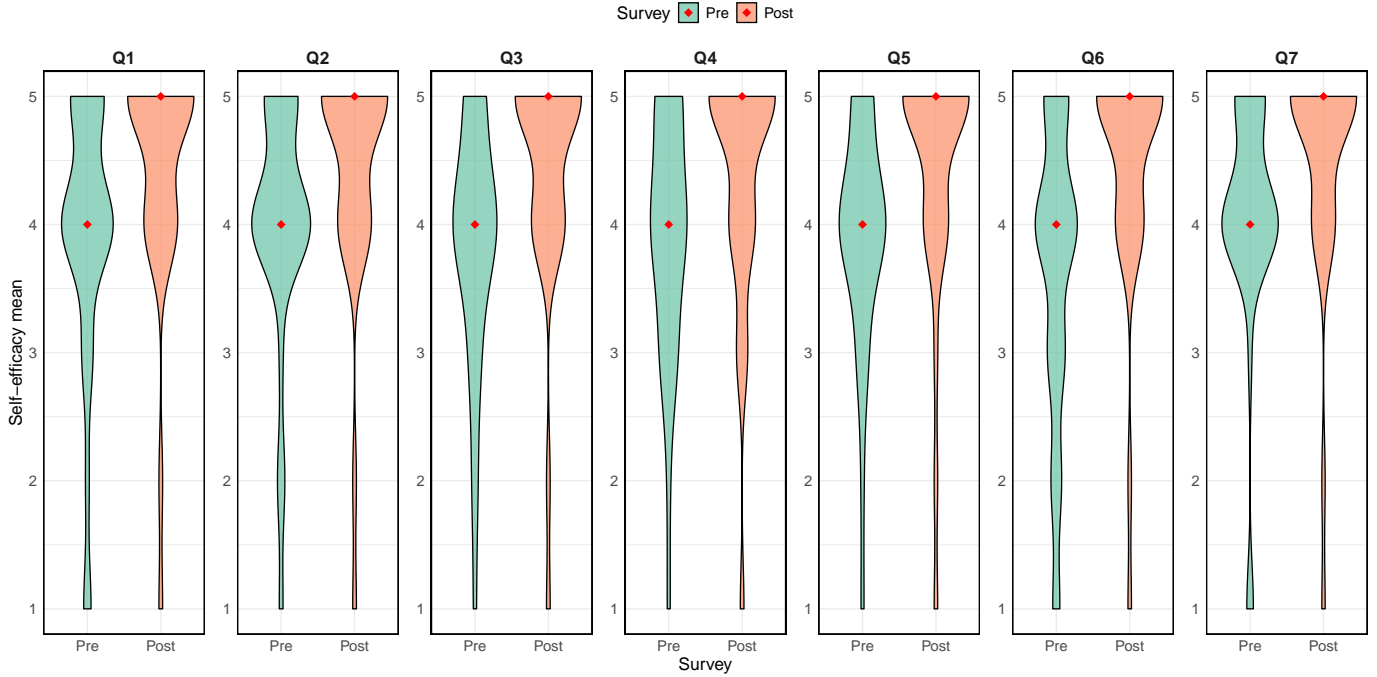
Pre vs Post Self−efficacy

Figure 5. Responses to the 5-point Likert scale questions for the self-efficacy questions. The left side ("Pre") shows students' responses to the questions before using OSSDoorway, while the right side ("Post") shows their responses after using OSSDoorway. The red dot indicates the median response for each question.

Question description: *I am confident that I can...* ***Q1:*** *use GitHub issue tracker to find open issues;* ***Q2:*** *understand GitHub pull requests;* ***Q3:*** *fork GitHub repositories;* ***Q4:*** *find help with an issue using the GitHub web interface;* ***Q5:*** *open a pull request using the GitHub web interface;* ***Q6:*** *find an issue to work on and assign it to myself;* ***Q7:*** *use GitHub to contribute to projects.* The left side (Pre) shows students' responses to the questions before using OSSDoorway, while the right side (Post) shows their responses after using OSSDoorway.

A. Results

All participants successfully completed all quests within the study time, with no discernible differences among demographics. Therefore, we answer RQ: *How does OSSDoorway support students in contributing to OSS projects?* by analyzing (1) the results of the self-efficacy questionnaire, which was administered before and after the student's interaction with OSSDoorway, and (2) the students' perceptions of using OSSDoorway.

1) Students' self-efficacy: Figure 5 presents the results of the self-efficacy questionnaire that students filled out before ("pre") and after ("post") using OSSDoorway to contribute to OSS projects. Our findings show increased students' self-efficacy after using OSSDoorway, with post-scores consistently higher than pre-scores across all seven survey questions (Q1 to Q7). This indicates that students felt more confident in contributing to OSS projects after using OSSDoorway to complete the tasks. While the degree of improvement varied, the data suggests that OSSDoorway successfully enhanced students' self-efficacy in areas where they initially were less confident.

We calculated the Wilcoxon signed-rank test, illustrated in Table I, a frequently used nonparametric test for paired data (e.g., pre-and post-treatment measurements) [47], and applied the Bonferroni correction for multiple comparisons. The dif-

ference in improvement ("pre" vs. "post") using OSSDoorway is significant for four out of the seven tasks, which saw an increase in both mean and median scores, with all p-values below 0.05. The most substantial improvements were observed in tasks like finding help with issues (Q4) and assigning issues (Q6), which had the lowest pre-assessment scores and the most significant changes (p-values 0.024 and 0.003, respectively). These results suggest that OSSDoorway boosts students' confidence in navigating GitHub and contributing to OSS projects.

Segmented analysis. We analyzed the data by segmenting it to compare men and women and applied the Mann-Whitney U test, a widely used nonparametric test for independent samples [56]. We compared them in both the pre-and post-self-efficacy assessments (i.e., comparing men pre vs. women pre and men post vs. women post). The results revealed a statistically significant difference in the pre-test (p-value = 0.003), indicating that men had a higher self-efficacy before using OSSDoorway (mean = 4.09) compared to women (3.69). This result is consistent with the literature [53], which found that women tend to have lower self-efficacy than men in their peer group. In the post-test, there was no significant difference between men and women (p-value = 0.914), with both groups achieving a median score of 5 and similar means (men: 4.5, women: 4.3). These findings suggest that while men

| | OSSDoorway | | | | |
| | Pre Self-efficacy | | Post Self-efficacy | | |
	mean	median	mean	median	p-value*
Q1...use GitHub issue tracker to find open issues	3.9	4	4.4	5	0.085
Q2...understand GitHub pull-requests	4	4	4.4	5	0.036
Q3...fork GitHub repositories	3.8	4	4.4	5	0.045
Q4...find someone to help me with an issue using the GitHub web interface	3.9	4	4.4	5	0.024
Q5...open a pull request using GitHub web interface	3.9	4	4.4	5	0.073
Q6...find an issue to work on and assign it to me	3.6	4	4.4	5	0.003
Q7...use GitHub to contribute to projects	4	4	4.5	5	0.080

p-value after applying the Bonferroni correction.

initially exhibited higher self-efficacy, *the use of OSSDoorway contributed to leveling self-efficacy for men and women*.

2) Students' perception about OSSDoorway: As mentioned, we adopted open coding in our qualitative analysis to identify recurring themes across students' responses.

Difficulties using OSSDoorway. We asked participants to share the most difficult aspects they encountered while using OSSDoorway. Feedback revealed that many students faced *challenges to solve an issue* during Quest 3, which requires the student to solve an issue and submit a pull request. For example, P1 noted that "*solving the issue and submitting a pull request was a bit hard*", while P9 shared that "*The most difficult is the implementation of pull request and committing the code to the main as it is my first time to learn and implement such thing.*" Some participants also mentioned difficulties with navigation and task clarity. Overall, while some students found the process manageable with clear instructions—P25 remarked that "*The process was really easy. I didn't find anything difficult. The instructions were very detailed which helped me complete the quest on time.*"—others encountered specific challenges, particularly related to Quest 3 and navigating GitHub features.

Tool support. Our results highlight what students found helpful while using the OSSDoorway environment that supported them through task completion. One indicated feature was the README file in the OSSDoorway that shows to the user the game elements to help them track their progress, which participants frequently mentioned as crucial for their success. For example, P1 stated that "*accessing information through the README file was very helpful*". The quest system and its process-oriented design also played a significant role in facilitating the learning experience. P25 noted that the quests helped them "*All the elements were easy to locate. The quest was interactive and was well defined*", making the process easier to navigate. Similarly, P10 highlighted that the quests' structure provided a clear and logical progression, helping them track their progress effectively. Highlighted instructions within the tasks were also indicated as useful, P26 elaborated:

"*I think the step-by-step instructions were very helpful*", ensuring clarity and reducing confusion during task execution. Students also indicated that the OSS Bot offered real-time feedback and support. As P9 mentioned, "*Using of Chatbot to see if we have done the changes or not is the most helped part in the whole process for me.*". At the same time, P11 emphasized "*Chat bot helped me a lot to confirm whether my answer is correct or not and description of each question of quest and directions in the question helped me to complete the assignment*".

Feedback to improve OSSDoorway. The theme *improve quests description* emerged in students feedback. P13 highlighted this with the remark: "*Even though the instructions were clear, it was a little difficult for a newbie to explore everything on his own.*" This suggests that, while the instructions were adequate, a more detailed guide may be needed to better support students' needs. Overall, we received positive feedback, such as P2's comment: "*I think the process is as simple as it can be and I can't think of anything to specifically change to improve the process*", and P25's observation: "*The quest can increase their level and add more interesting topics. It was really fun to learn about GitHub through this quest. I personally liked it.*" These responses indicate that students found the experience using OSSDoorway enjoyable.

Research Question

How does OSSDoorway support students in contributing to OSS projects?

- - - - - - - - - -

Answer: All students accomplished the tasks, and OSSDoorway significantly improved students' self-efficacy in contributing to OSS projects. Even though men had a higher self-efficacy than women before using OSSDoorway, this difference disappeared afterward. According to the students, OSSDoorway provides clear instructions, real-time feedback, and a structured quest-based learning environment.

VI. DISCUSSION

This section highlights the insights gained from our research findings and explores potential opportunities for further investigation.

Guiding Students with Quests. Our study found that using OSSDoorway's structured, sequential tasks helped mitigate navigation and task clarity challenges. This aligns with previous research, which shows that step-by-step guidance, such as that provided by quests and points, can reduce barriers for newcomers in OSS environments [27]. Our participants also highlighted the value of quests in effectively tracking their progress and guiding their navigation through the OSS contribution process. Moreover, similar to the findings from Diniz et al. [27], quests can effectively orient students toward their first contribution, keeping them engaged throughout their journey.

Gamification in OSS education. Applying game elements to OSS contributions can support contribution. Even though gamification can also help newcomers overcome barriers [17],

especially those related to process and environment, other barriers can still impact student participation, such as toxic culture and exclusionary practices, as highlighted by Trinkenreich et al. [20]. While gamification can enhance engagement, it must complement broader efforts to address systemic issues in OSS communities. Future research should explore how gamification can help mitigate these barriers.

Motivations to contribute to OSS projects. Previous research has explored the motivations behind developers' contributions to OSS [57], [58], [59], [60]. Developers are extrinsically motivated when they seek external rewards, such as career advancement or financial compensation [61]. These external motivators can become internalized, leading developers to perceive them as self-regulated behaviors driven by factors like reputation, reciprocity, learning, and personal use [59], [62]. Intrinsic motivations, by contrast, arise from acting for its inherent satisfaction, such as ideology, altruism, or fun [59], [63]. Gerosa et al. [60] found that intrinsic and internalized extrinsic motivations largely drive contributors, while other studies suggest extrinsic rewards, like potential monetary gain, also play a role [64]. Past work has also investigated students' motivation to contribute to OSS [7]. Future research should explore how to motivate students to contribute to OSS projects by aligning gamification with these motivational drivers.

Students' self-efficacy improvement. Previous work [18], [19] has demonstrated that the information architecture of OSS project pages (e.g., project descriptions and issue tracker details) typically appeals to individuals with high self-efficacy, driven by motivations like intellectual stimulation, competition, and learning technology for enjoyment. Our findings indicate that using OSSDoorway significantly enhanced students' self-efficacy in completing tasks related to contributing to OSS projects. These results align with previous studies that demonstrated how redesigning GitHub through a web browser plugin also increased students' self-efficacy [65]. Future research could explore additional strategies to boost students' confidence, such as refining gamified elements or integrating personalized feedback mechanisms to better support learners in the OSS contribution process.

VII. IMPLICATIONS

In this section, we outline the implications of our study. ***Implications for students.*** For students, especially those with lower self-efficacy, OSSDoorway's structured environment significantly increases self-efficacy in contributing to OSS projects. The results show that quests and real-time feedback help reduce the fear of making mistakes and provide a clear path to contribute to OSS projects. Students can benefit from these features by following guided steps to learn concepts in OSS tools, such as issue tracking and pull requests. Our findings emphasize the need to provide students with an environment that offers support through feedback and structured questions, which can alleviate common frustrations and help students build the skills necessary to contribute effectively.

Implications for educators. Incorporating OSS contributions into the computer science curriculum offers valuable real-world experience to students. Still, educators should recognize that some students—particularly those with lower self-efficacy—may struggle more when using platforms like GitHub. Educators could introduce tools like OSSDoorway to provide a more scaffolded learning experience, offering clear, step-by-step tasks and real-time feedback to ease students into OSS contributions.

Implications for social coding platforms. Our findings indicate that structured guidance, such as quests and real-time feedback, benefits students, increasing their self-efficacy. Social coding platforms like GitHub could incorporate similar features to support new users.

Implications for maintainers of OSS projects. Projects could benefit from student contributions, especially those who struggle to attract external contributors. Including contextualized quests or guided tutorials could also help lower the entry barriers for students. Additionally, maintainers should consider providing real-time feedback with automated bots, similar to OSSDoorway, to guide new contributors through their first contributions.

Implications for researchers. Our study adds to the current literature on gamification in OSS by exploring its impact on self-efficacy and supporting students in contributing to OSS projects. Future research should further investigate how specific game elements, like quests and feedback systems, affect different students' learning styles and motivations. Additionally, understanding the long-term effects of using gamified tools like OSSDoorway on student retention and performance in real-world OSS projects would be valuable.

VIII. LIMITATIONS

Even though we developed OSSDoorway based on the outcomes of each step in our research method and conceptualized this environment using surveys, and interviews, the requirements elicitation process may not fully cover all the nuances of contributing to OSS projects, particularly in capturing various contribution types and strategies. To address this, we evaluated the tasks implemented in OSSDoorway with experts. Nevertheless, while OSSDoorway aims to support students in the initial steps of the contribution process, such as requiring students to make non-code contributions, it does not encompass the full range of contribution methods or delve deeply into technical specifics. Our focus was primarily on the contribution process and the tools rather than the technical details of coding or development. Future studies can explore how OSSDoorway can support non-technical contributors (e.g., writers) in engaging with OSS projects.

While OSS promotes teamwork, gamification can sometimes prioritize competition and personal milestones, potentially overlooking the social and OSS cooperative aspects. In this version of OSSDoorway, we chose not to incorporate competitive elements to ensure the primary focus remained on learning and skill development. However, future studies

could explore how gamified approaches might better integrate collaborative and competitive mechanics.

Concerning the generalizability of our study, our results can be partially influenced by the specific sample of participants. OSSDoorway may yield different outcomes if used by students with varying skill levels in OSS tools. Additionally, our study is limited to the use of GitHub. We cannot assume that the benefits observed with OSSDoorway will be as effective in real-world OSS projects as they were identified in the controlled sandbox environment of OSSDoorway. Future work can investigate how students transfer the acquired knowledge to real project settings.

Regarding the reliability of the study's conclusions, we employed nonparametric statistical tests with minimal statistical assumptions and Bonferroni correction for multiple comparisons. For the qualitative analysis, we acknowledge the potential for bias in data interpretation. To mitigate subjectivity, the team engaged in continuous comparative analysis and reached conclusions through a process of negotiated agreement. Each team member has extensive experience in qualitative methods and OSS.

IX. Conclusion

In this study, we presented OSSDoorway, a gamified environment designed to support students in contributing to OSS projects, and evaluated it with 29 graduate students. Our study demonstrates that OSSDoorway effectively supports students in contributing to OSS projects by significantly improving their self-efficacy and providing a structured, gamified learning environment. The combination of game elements, real-time feedback from the OSS Bot, and the quest-based system helped students navigate complex tasks such as understanding GitHub features, submitting pull requests, and collaborating within a community. Our findings suggest that OSSDoorway can be a valuable tool for supporting students in contributing to OSS projects. Future work will focus on refining and expanding OSSDoorway to include more advanced contributions, such as code review and refactoring. Additionally, we plan to conduct an empirical study to evaluate the OSSDoorway's impact, considering students' different cognitive styles.

X. Data Availability

The replication package, with the research instruments and scripts, is available for public access [54].

References

[1] Y. Sedelmaier and D. Landes, "A research agenda for identifying and developing required competencies in software engineering," in *2012 15th International Conference on Interactive Collaborative Learning (ICL)*. IEEE, 2012, pp. 1–5.

[2] G. Cantone, M. Marchesi *et al.*, "Agile processes in software engineering and extreme programming," in *Proceedings of 15th international conference, XP*. Springer, 2014.

[3] R. Holmes, M. Allen, and M. Craig, "Dimensions of experientialism for software engineering education," in *Proceedings of the 40th International Conference on Software Engineering: Software Engineering Education and Training*, 2018, pp. 31–39.

[4] O. Cawley, S. Weibelzahl, I. Richardson, and Y. Delaney, "Incorporating a self-directed learning pedagogy in the computing classroom: Problem-based learning as a means to improving software engineering learning outcomes," in *Overcoming Challenges in Software Engineering Education: Delivering Non-Technical Knowledge and Skills*. IGI Global, 2014, pp. 348–371.

[5] D. M. Nascimento, K. Cox, T. Almeida, W. Sampaio, R. A. Bittencourt, R. Souza, and C. Chavez, "Using open source projects in software engineering education: A systematic mapping study," in *2013 IEEE Frontiers in Education Conference (FIE)*. IEEE, 2013, pp. 1837–1843.

[6] G. H. L. Pinto, F. Figueira Filho, I. Steinmacher, and M. A. Gerosa, "Training software engineers using open-source software: the professors' perspective," in *CSEE&T*. IEEE, 2017, pp. 117–121.

[7] J. O. Silva, I. Wiese, D. M. German, C. Treude, M. A. Gerosa, and I. Steinmacher, "Google summer of code: Student motivations and contributions," *JSS*, vol. 162, p. 110487, 2020.

[8] G. Pinto, C. Ferreira, C. Souza, I. Steinmacher, and P. Meirelles, "Training software engineers using open-source software: the students' perspective," in *ICSE-SEET 2019*. IEEE, 2019.

[9] G. Braught, J. Maccormick, J. Bowring, Q. Burke, B. Cutler, D. Goldschmidt, M. Krishnamoorthy, W. Turner, S. Huss-Lederman, B. Mackellar *et al.*, "A multi-institutional perspective on H/FOSS projects in the computing curriculum," *ACM Transactions on Computing Education (TOCE)*, vol. 18, no. 2, pp. 1–31, 2018.

[10] B. Morgan and C. Jensen, "Lessons learned from teaching open source software development," in *IFIP International Conference on Open Source Systems*. Springer, 2014.

[11] Y. Cai and D. Zhu, "Reputation in an open source software community: antecedents and impacts," *Decision Support Systems*, 2016.

[12] D. Riehle, "How open source is changing the software developer's career." *Computer*, 2015.

[13] L. Singer, F. Figueira Filho, B. Cleary, C. Treude, M.-A. Storey, and K. Schneider, "Mutual assessment in the social programmer ecosystem: An empirical investigation of developer profile aggregators," in *Proceedings of the 2013 Conference on Computer Supported Cooperative Work*, ser. CSCW '13. New York, NY, USA: ACM, 2013, pp. 103–116.

[14] J. Marlow, L. Dabbish, and J. Herbsleb, "Impression formation in online peer production: Activity traces and personal profiles in GitHub," in *Proceedings of the 2013 Conference on Computer Supported Cooperative Work*, ser. CSCW '13. New York, NY, USA: ACM, 2013, pp. 117–128.

[15] E. Parra, S. Haiduc, and R. James, "Making a difference: an overview of humanitarian free open source systems," in *ICSE 2016-Companion*. ACM, 2016.

[16] G. J. Greene and B. Fischer, "Cvexplorer: identifying candidate developers by mining and exploring their open source contributions," in *ASE 2016*. ACM, 2016.

[17] I. Steinmacher, T. Conte, M. Gerosa, and D. Redmiles, "Social barriers faced by newcomers placing their first contribution in open source software projects," in *ACM CSCW 2015*, 2015.

[18] C. Mendez, H. S. Padala, Z. Steine-Hanson, C. Hilderbrand, A. Horvath, C. Hill, L. Simpson, N. Patil, A. Sarma, and M. Burnett, "Open source barriers to entry, revisited: a sociotechnical perspective," in *ICSE 2018*, 2018.

[19] S. H. Padala, C. J. Mendez, L. F. Dias, I. Steinmacher, Z. S. Hanson, C. Hilderbrand, A. Horvath, C. Hill, L. D. Simpson, M. Burnett *et al.*, "How gender-biased tools shape newcomer experiences in OSS projects," *IEEE TSE*, 2020.

[20] B. Trinkenreich, I. Wiese, A. Sarma, M. Gerosa, and I. Steinmacher, "Women's participation in open source software: a survey of the literature," *ACM TOSEM*, 2022.

[21] I. Steinmacher, T. Conte, C. Treude, and M. Gerosa, "Overcoming open source project entry barriers with a portal for newcomers," in *ICSE 2016*, 2016.

[22] I. Santos, I. Wiese, I. Steinmacher, A. Sarma, and M. A. Gerosa, "Hits and misses: Newcomers' ability to identify skills needed for OSS tasks," in *IEEE SANER*. IEEE, 2022, pp. 174–183.

[23] C. Dichev and D. Dicheva, "Gamifying education: what is known, what is believed and what remains uncertain: a critical review," *International journal of educational technology in higher education*, vol. 14, pp. 1–36, 2017.

[24] O. Pedreira, F. García, N. Brisaboa, and M. Piattini, "Gamification in software engineering–a systematic mapping," *Information and Software Technology (IST)*, vol. 57, pp. 157–168, 2015.

[25] D. Dicheva, C. Dichev, G. Agre, and G. Angelova, "Gamification in education: A systematic mapping study," *ET&S*, vol. 18, no. 3, pp. 75–88, 2015.

[26] A. Bartel and G. Hagel, "Gamifying the learning of design patterns in software engineering education," in *EDUCON*. IEEE, 2016, pp. 74–79.

[27] G. C. Diniz, M. A. G. Silva, M. A. Gerosa, and I. Steinmacher, "Using gamification to orient and motivate students to contribute to OSS projects," in *IEEE/ACM 10th CHASE*. IEEE, 2017, pp. 36–42.

[28] B. R. Belland, *Scaffolding: Definition, Current Debates, and Future Directions*. New York, NY: Springer New York, 2014, pp. 505–518.

[29] S. Deterding, R. Khaled, L. E. Nacke, D. Dixon *et al.*, "Gamification: Toward a definition," in *CHI*, vol. 12. Vancouver, 2011, pp. 1–79.

[30] M. Sailer and L. Homner, "The gamification of learning: A meta-analysis," *Educational psychology review*, vol. 32, no. 1, pp. 77–112, 2020.

[31] B. Marín, J. Frez, J. Cruz-Lemus, and M. Genero, "An empirical investigation on the benefits of gamification in programming courses," *ACM Transactions on Computing Education (TOCE)*, vol. 19, no. 1, pp. 1–22, 2018.

[32] M. Ayub, H. Toba, M. C. Wijanto, S. Yong, and B. Wijaya, "Gamification for blended learning in higher education," *World Transactions on Engineering and Technology Education*, vol. 17, no. 1, pp. 76–81, 2019.

[33] A. M. Toda, A. C. Klock, W. Oliveira, P. T. Palomino, L. Rodrigues, L. Shi, I. Bittencourt, I. Gasparini, S. Isotani, and A. I. Cristea, "Analysing gamification elements in educational environments using an existing gamification taxonomy," *Smart Learning Environments*, vol. 6, no. 1, pp. 1–14, 2019.

[34] C. R. Prause, J. Nonnen, and M. Vinkovits, "A field experiment on gamification of code quality in agile development." in *PPIG*, 2012, p. 17.

[35] J. M. Rojas and G. Fraser, "Teaching mutation testing using gamification," in *ECSEE*, 2016.

[36] G. Garaccione, T. Fulcini, and M. Torchiano, "Gerry: a gamified browser tool for GUI testing," in *Gamify'22*, 2022, pp. 2–9.

[37] S. Ozturk, "Gamification of exploratory testing process," in *Gamify'22*, 2022, pp. 14–17.

[38] G. Annunziata, S. Lambiase, F. Palomba, and F. Ferrucci, "Serge–serious game for the education of risk management in software project management," in *Proceedings of the 46th International Conference on Software Engineering: Software Engineering Education and Training*, 2024, pp. 264–273.

[39] E. O. Navarro and A. Van Der Hoek, "Comprehensive evaluation of an educational software engineering simulation environment," in *20th Conference on Software Engineering Education & Training (CSEET'07)*. IEEE, 2007, pp. 195–202.

[40] G. Garaccione, R. Coppola, L. Ardito, and M. Torchiano, "Gamification of business process modeling education: an experimental analysis," *Software and Systems Modeling*, pp. 1–26, 2024.

[41] C.-H. Su, "The effects of students' motivation, cognitive load and learning anxiety in gamification software engineering education: a structural equation modeling study," *Multimedia Tools and Applications*, vol. 75, pp. 10 013–10 036, 2016.

[42] S. K. Sheth, J. S. Bell, and G. E. Kaiser, "Increasing student engagement in software engineering with gamification," in *4th SSE*, 2012.

[43] S. Zhao15, X. Xia25, B. Fitzgerald, X. Li, V. Lenarduzzi, D. Taibi, R. Wang, W. Wang, and C. Tian, "Motivating open source collabora-tions through social network evaluation: A gamification practice from alibaba," in *IEEE/ACM 46th ICSE-SEIP*, 2024.

[44] I. Santos, K. R. Felizardo, I. Steinmacher, and M. A. Gerosa, "Software solutions for newcomers' onboarding in software projects: A systematic literature review," *Information and Software Technology*, vol. 177, p. 107568, 2025.

[45] C. Toscani, D. Gery, I. Steinmacher, and S. Marczak, "A gamification proposal to support the onboarding of newcomers in the flosscoach portal," in *17th IHC*, 2018, pp. 1–10.

[46] I. Santos, K. R. Felizardo, M. A. Gerosa, and I. Steinmacher, "Game elements to engage students learning the open source software contribution process," in *2024 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, 2024, pp. 59–70.

[47] B. Rosner, R. J. Glynn, and M.-L. T. Lee, "The wilcoxon signed rank test for paired comparisons of clustered data," *Biometrics*, 2006.

[48] A. Strauss and J. Corbin, *Basics of qualitative research techniques*. Sage Publications, 1998.

[49] D. W. Eccles and G. Arsal, "The think aloud method: what is it and how do i use it?" *Qualitative Research in Sport, Exercise and Health*, vol. 9, no. 4, pp. 514–531, 2017.

[50] B. Trinkenreich, M. Guizani, I. Wiese, A. Sarma, and I. Steinmacher, "Hidden figures: Roles and pathways of successful oss contributors," *ACM on Human-Computer Interaction*, vol. 4, no. CSCW, pp. 1–22, 2020.

[51] N. Foundation, "NodeJS," 2024, accessed: 2024-09-20. [Online]. Available: https://nodejs.org/

[52] P. Team, "Probot framework," 2024, accessed: 2024-09-20. [Online]. Available: https://probot.github.io/

[53] M. Burnett, S. Stumpf, L. Beckwith, and A. Peters, "The GenderMag kit: how to use the GenderMag method to find inclusiveness issues through a gender lens," 2018.

[54] Figshare, "Replication package," 2024, accessed: 2024-09-09. [Online]. Available: https://figshare.com/s/0b93fef5c7ef015e2ddf

[55] A. Bandura, "Social cognitive theory of moral thought and action," in *Handbook of Moral Behavior and Development*. Psychology press, 2014.

[56] T. W. MacFarland, J. M. Yates, T. W. MacFarland, and J. M. Yates, "Mann–whitney u test," *Introduction to nonparametric statistics for the biological sciences using R*, pp. 103–132, 2016.

[57] K. R. Lakhani and E. Von Hippel, *How open source software works:"free" user-to-user assistance*. Springer, 2004.

[58] J. Bitzer, W. Schrettl, and P. J. Schröder, "Intrinsic motivation in open source software development," *Journal of comparative economics*, vol. 35, no. 1, pp. 160–169, 2007.

[59] G. Von Krogh, S. Haefliger, S. Spaeth, and M. W. Wallin, "Carrots and rainbows: motivation and social practice in open source software development," *MIS Q*, 2012.

[60] M. Gerosa, I. Wiese, B. Trinkenreich, G. Link, G. Robles, C. Treude, I. Steinmacher, and A. Sarma, "The shifting sands of motivation: Revisiting what drives contributors in open source," in *ICSE 2021*. IEEE, 2021.

[61] B. S. Frey *et al.*, "On the relationship between intrinsic and extrinsic work motivation," *International journal of industrial organization*, vol. 15, no. 4, pp. 427–439, 1997.

[62] E. L. Deci and R. M. Ryan, "The support of autonomy and the control of behavior." *Journal of personality and social psychology*, vol. 53, no. 6, p. 1024, 1987.

[63] R. M. Ryan and E. L. Deci, "Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being." *American psychologist*, vol. 55, no. 1, p. 68, 2000.

[64] S. O. Alexander Hars, "Working for free? motivations for participating in open-source projects," *International journal of electronic commerce*, vol. 6, no. 3, pp. 25–39, 2002.

[65] I. Santos, J. F. Pimentel, I. Wiese, I. Steinmacher, A. Sarma, and M. A. Gerosa, "Designing for cognitive diversity: Improving the GitHub experience for newcomers," in *2023 IEEE/ACM 45th ICSE-SEIS*. IEEE, 2023, pp. 1–12.