

Perceptions of Undesirable Software Development Tasks Among Computer Science Students

Omar Chaaban, Mary Sánchez-Gordón, and Selina Demi

Østfold University College, 1757 Halden, Norway

ABSTRACT

Software development projects include both technical and non-technical tasks, some of which may be perceived as undesirable by developers. While prior research has examined how developers perceive and manage such tasks, students' perspectives remain unexplored. Therefore, this study aims to explore task undesirability for students. A survey-based approach was adopted to gather insights from 30 participants enrolled in IT-related programs at Østfold University College. The preliminary results revealed that back-end related tasks, bug fixing and writing documentation were among the most undesirable tasks, leading to negative emotional responses, such as frustration and demotivation among students. In addition, strategies such as "Just Do It" and working in groups were commonly employed to cope with these tasks. Although further research is needed to better understand support mechanisms that enhance students' engagement and well-being, findings suggest that coping strategies arise when students' experiences are not sufficiently supported. This provides an opportunity to design courses that enhance learning experiences by not only mitigating the negative impact of undesirable tasks but also making these tasks more desirable for students.

Keywords: Undesirable tasks, Software development education, Survey

INTRODUCTION

In software development, tasks refer to the work or activities that must be completed to achieve project objectives within a defined period (Schmidt, 2000). Overall, the type of task assigned to developers plays an important role in their personal satisfaction, morale, and project outcome. Various studies have been conducted in this research area. Afanasieva et al. (2019) introduced a framework for assessing professional growth among developers, which optimizes task distribution to align with developers' skill levels for improved project management, efficiency, and productivity. Bradley et al. (2022) identified sources of friction in software development tasks, advocating for tool and environment enhancements to streamline development workflows. Moreover, Nundlall & Nagowah (2021) identified different factors and challenges of task allocation and coordination in distributed agile teams to facilitate efficient task management and collaboration.

Etemadi et al. (2022) proposed a self-adaptive task assignment (SATA) approach that balances cost and knowledge diffusion by optimizing task assignment, thereby mitigating knowledge loss and reducing maintenance

costs. More recently, Kurian & Thomas (2023) and Lima et al. (2023) highlighted the importance of the quality of code refactoring and positive emotions in enhancing software testing performance and developers' creativity, motivation, and engagement in teamwork, which lead to better outcomes. However, the introduction of Agile methodologies and self-assignment practices creates challenges and opportunities in task allocation and execution. Similarly, Nundlall & Nagowah (2021) pointed out the limitations in current agile practices related to task assignment and coordination.

From the human perspective, only a few studies have focused on the impact of task desirability. Licorish & MacDonell (2018) discussed the links between task types, team attitudes, and completion performance, suggesting that task desirability could significantly affect project success. Consequently, Masood et al. (2022) provided an overview of the factors that contribute to the (un)desirability of software development tasks and developers' motivation to engage with certain types of work that are more appealing than others which lessen productivity. In addition, the significance of task desirability was discussed by Russo et al. (2023). These authors explored the satisfaction and performance of software developers working from home during the COVID-19 pandemic, revealing the impact of assigned tasks with developer preferences on productivity and well-being.

Based on the review conducted on this topic, it was observed that existing studies focused predominantly on professional environments, while the academic context remains underexplored. This study aims to bridge this gap by exploring how undesirable software development tasks, contributing factors, impact and strategies are perceived by students. Therefore, we formulate the following research questions:

RQ1: What are the main undesirable software development tasks for students?

RQ2: What factors contribute to students perceiving software tasks as undesirable?

RQ3: How does working on undesirable tasks impact students?

RQ4: How do students cope with undesirable software development tasks?

This study builds upon the foundational work in (Masood et al., 2022) which considered both technical tasks and non-technical/collaborative activities as "tasks." However, while the authors of that study conducted semi-structured interviews with 32 software developers from different age, experience, gender, role, company size, and project domain, our exploratory study is based on a survey that seeks to provide preliminary insight into undesirable software development tasks within the Norwegian academic context, particularly at Østfold University College.

The paper is structured as follows: Section 2 outlines the research approach, including the survey design and data analysis approach. Section 3 presents the results, Section 4 discusses them, and Section 5 provides some conclusions.

RESEARCH APPROACH

This research uses a survey-based approach to investigate the proposed research questions. The survey design is based on the empirically evaluated checklist proposed in (Molléri et al., 2020), which focuses on the importance of clear, concise, and relevant survey questions to ensure the accuracy and credibility of the collected data. The survey comprises questions about students' perceptions of task desirability (RQ1) and contributing factors to task desirability (RQ2), as well as feelings experienced by students when performing such undesirable tasks (RQ3). Finally, the survey addresses RQ4 by including questions aimed at understanding the strategies adopted to cope with undesirable tasks.

Participants. The sample was chosen from a population of students enrolled in IT-related programs, such as computer science and software engineering at Østfold University College. The aim was to reach a group of students across different stages of their academic level, from undergraduate to graduate students. The recruitment process ensured that students had engaged in at least one academic software development project, as some experience with such tasks is essential for forming valid opinions. Students without this experience were therefore excluded.

Ethical Considerations. This study follows ethical research guidelines provided by Østfold University College. The participants were assured of the confidentiality and anonymity of the data collected. They were informed about their right to withdraw from the survey at any time without any consequence. An email for direct contact was visibly placed, allowing participants to directly contact the party responsible if they had any doubts or concerns.

Questionnaire Design

The questionnaire was developed based on the guidelines proposed in (Molléri et al., 2020) and included both closed and open-ended questions. The closed-ended questions used 5-point Likert scale questions to measure the perception of students on undesirability of tasks and the strategies adopted to manage them. Moreover, optional open-ended questions allowed participants to explain or describe freely in their own words, providing more profound insights and feedback.

The survey consists of five sections, beginning with participants' basic demographic information, which can help contextualize the finding. The remaining survey questions were formulated based on the findings identified in (Masood et al., 2022) focusing on tasks, factors, feelings, and strategies, and were organized into four parts, following the categorization proposed in (Masood et al., 2022). The questionnaire was then reviewed by a senior researcher, which led to the inclusion of frequency-related questions in each section. These questions aimed to provide deeper insights into participants' opinions and habits, such as how often they experience a feeling or use a strategy to deal with undesirable tasks.

In what follows, the five parts of the survey are described. The full version of the survey, including all questions, is available and accessible in a public repository (*Software Development Task Management Survey*, 2024).

First part: Demographic information. This part includes an optional question about participants' gender (Q1), program of study (Q2), and the year of study (Q3). Moreover, participants reported their experience with software development projects (Q4) and their software development knowledge level (Q6). Finally, an open question is included to identify what technologies/tools were used in their projects (Q5).

Second part: Perception of tasks' undesirability (RQ1). Participants were asked to indicate the extent to which they dislike software development tasks based on a 5-point scale (1: Not at all – 5: Very much) (Q7). Examples of such tasks are front-end related tasks, back-end related tasks, developing/adding new functionalities, writing documentation, and meetings. Then, a question about frequency of performing these tasks (Q8) was asked based on a 5-point scale (1: Never – 5: Very often). Finally, an optional open-ended question (Q9) allowed the participants to share any other undesirable software development tasks not included in the list.

Third part: Factors that contribute to a task being undesirable (RQ2). This part started with the following question (Q10): "Please indicate to what extent you agree with the factors that may influence your dislike of a task (1: Strongly disagree – 5: Strongly agree)". This question was included to identify the factors contributing to a task being undesirable to students, i.e., what makes a task undesirable to them.

Masood et al. (2022) categorized the emerged factors into five groups: personal, social, organizational, technical, and operational. Based on those, a list of sixteen factors was identified, e.g., Don't bring me the opportunity to learn; Don't bring me the opportunity to improve my skills and/or abilities; Don't allow me to take initiative and have responsibility for the task; Lack of knowledge to execute the task. Then, a question (Q11) about frequency was asked: "Please indicate how often you have experienced these factors in these tasks (1: Never – 5: Very often)". An optional open-ended question (Q12) provided participants with the opportunity to share any other additional factors contributing to their perception of a task as undesirable beyond those listed above.

Fourth part: The perceived impact of performing undesirable software development tasks (RQ3). This part included a question (Q13) about "How do you feel when performing undesirable tasks? Please indicate the intensity of your feelings when performing undesirable tasks" based on a 5-point scale (1: Not at all – 5: Very much). Then, ten feelings were listed, e.g., Unhappy, Unmotivated, Stressed, Frustrated, and Bored. Moreover, another question (Q14) was included to understand how often they experience these feelings when performing undesirable tasks: "Please indicate how often you experience this sensation", based on a 5-point scale (1: Never – 5: Very often). Similar to the previous parts, an optional open-ended question (Q15) was included to allow participants to share any other feelings or thoughts beyond those listed above.

Fifth part: Strategies used to tackle undesirable tasks (RQ4). Participants were asked (Q16) to "Please indicate the frequency with which you use the following strategies to manage undesirable tasks" based on a 5-point scale (1:

Never – 5: Very often). The strategies include tools, techniques, or practices they find effective in the management of tasks, e.g., Just do it, whether (un) desirable; Assign these tasks to the most inexperienced; Assign these tasks to each teammate a part so that it becomes a team activity. In addition, question (Q17) asked: “Please indicate to what extent you perceive that the strategies used have worked,” based on a 5-point scale (1: Never – 5: Always). However, a Not Applicable option (6: NA/Not used) was added to allow participants who did not use the strategy to choose accordingly. Finally, an optional open-ended question (Q18) was included to enable participants to share any other strategies they have used that were not found on the list.

Pilot test. A pilot test with two fellow students was conducted to evaluate the quality and clarity of the survey. They were asked to complete the survey and provide feedback on the clarity of the instructions, the comprehensibility of the questions, and the overall survey experience. It took longer than expected, as one of them had to be reminded three times in the span of ten days. Based on their feedback, minor adjustments were made to improve the survey’s clarity and ensure that the questions captured the intended information.

One of the students found the Likert scales used in Q7 and Q13 to be confusing. An earlier version of the scale of Q7 had the following formulation (1: Love it – 5: Hate it), which did not properly represent the intended meaning. Then, the scale was changed to (1: Not at all – 5: Very much). Likewise, Q13 was earlier formulated as (1: None – 5: High). In this case, the student claimed that the words did not appropriately represent feelings. Thereby, the description was also changed to (1: Not at all – 5: Very much). It is worth noting that both changes were revised after discussion with a senior researcher. The second student mentioned that it took a long time to complete the questionnaire (24 minutes).

Further research was needed to find ways to shorten the time it takes to complete the questionnaire without removing any questions. Décieux et al. (2015) suggested that participants should not be required to answer every question to proceed through the questionnaire. The study argued that forcing participants to answer every question increases dropout rates and reduces the quality of answers. Thus, all open-ended questions except for Q5 were made optional to allow participants to conclude the survey even if they did not have any additional insights.

Data Collection and Analysis

The survey took place online using a web-based tool (<http://nettskjema.no/>). It was distributed through direct contact, university mailing lists, and social media groups related to software development with students as members, e.g., Webapplikasjoner 2022 and Innføring i Datasikkerhet V2023 discord servers. Participants were informed of voluntary participation, the study’s purpose, data collected from their responses, and how it will be used. Consent was acquired from participants before they started the survey. After the pilot test, data were collected over a four-week period (April to May 2024).

A single reminder was sent to maximize response rates while minimizing inconvenience and encouraging survey sharing.

Quantitative data from closed-ended questions were analyzed using Excel. The descriptive statistics were used to identify which management strategies used for undesirable tasks were popular and to identify the possible effects on students' mental health. Thematic analysis was planned to analyze the open-ended questions, mainly for the flexibility and simplicity of this approach (Lester et al., 2020) to find patterns and themes related to task management strategies and their effectiveness in tackling undesirable tasks. While this approach enables the understanding of richer information provided by participants, only a limited number of responses (6) were collected.

Demographic Information. The participants comprised 30 students (20 undergraduates, seven master's students, two self-taught, and 1 PhD student), predominantly from informatics and digital media programs. The majority (15, 50%) were involved in 3-5 software development projects, with a median level of knowledge rating of 3, indicating a moderately experienced group. The distribution was significantly skewed towards male participants (20, 66.7%), with one preferring not to answer and the rest (9, 30%) being female. Moreover, most participants (9) used Visual Studio Code as an integrated development environment (IDE) to write code. Java and Python, back-end programming languages, appeared in 9 answers. Moreover, JavaScript, React, and a JavaScript library, appeared six times each. Based on the responses, students have used a range of technologies for both frontend and backend development.

RESULTS

This section presents findings related to our research questions.

Task (Un)desirability (RQ1)

The survey provides insights into what software development tasks students consider undesirable (Table 1) and how frequently they encounter them (Table 2).

Table 1: Overview of perception of tasks undesirability (1: Not at all – 5: Very much).

Task (Un)desirability	1	2	3	4	5	AVG
Front-end related-tasks	5	9	8	6	2	2.7
Back-end related-tasks	8	3	4	11	4	3.0
Developing/Adding new functionalities	9	5	7	5	4	2.7
Enhance existing functionalities	4	5	10	9	2	3.0
Reviewing and refactoring low-quality code	3	7	9	7	4	3.1
Fixing bugs or automation failures	4	3	10	8	5	3.2
Testing, for example unit tests	5	4	12	6	3	2.9

(Continued)

Table 1: Continued.

Task (Un)desirability	1	2	3	4	5	AVG
Writing documentation, e.g. requirements, use cases, manuals	4	9	5	7	5	3.0
Working with others to either coordinate or support other teams	7	7	7	7	2	2.7
Meetings, e.g., to plan and report the status of the project	4	8	10	6	2	2.8

As depicted in Table 1, the most disliked tasks are back-end related tasks (15 participants, 50%), followed by bug fixing (13, 44%), and writing documentation (12, 40%). On the other hand, the least disliked tasks are front-end related tasks (disliked by 8, 27%), meetings (disliked by 8, 27%), developing/adding new functionalities (disliked by 9, 30%), and working with others to coordinate or support each other (disliked by 9, 30%). Code testing received the most neutral rating, with 12 (40%) participants selecting the middle value.

Table 2: Overview of frequency of engagement (1: Never – 5: Very often).

Task (Un)desirability	1	2	3	4	5	AVG
Front-end related-tasks	1	8	9	5	7	3.3
Back-end related-tasks	2	8	9	7	4	3.1
Developing/Adding new functionalities	5	5	6	10	4	3.1
Enhance existing functionalities	4	8	10	7	1	2.8
Reviewing and refactoring low-quality code	6	10	8	2	4	2.6
Fixing bugs or automation failures	6	7	4	8	5	3.0
Testing, for example unit tests	4	9	11	5	1	2.7
Writing documentation, e.g. requirements, use cases, manuals	2	8	7	11	2	3.1
Work with others to either coordinate or support other teams	2	5	5	9	9	3.6
Meetings, e.g., to plan and report the status of the project	2	3	9	10	6	3.5

Table 2 shows that working with others (18, 60%) and meeting them (16, 53%) were the most recurring tasks students had engaged in. Undesirable tasks, such as writing documentation (13, 44%) and fixing bugs (13, 44%) were also frequently encountered, suggesting a significant source of pain for students in their projects. One participant mentioned IDE setup as an undesirable task, which suggests students had some sort of issue installing/setting up an IDE program. Another participant claimed, “*Required preplanning is not fun, I usually do the work then the preplanning,*” which pointed out that preplanning might be an undesirable task that is not named on our list. However, it is unclear whether the preplanning is an individual or a group-related task.

Factors Contributing to Task Undesirability (RQ2)

The results suggested the most common factors that make tasks undesirable, as follows:

- *Unrealistic deadlines* made tasks more undesirable, with a 70% rate of agreement (21 participants);
- *Lack of communication* about the importance of tasks and their future impact, with a 67% agreement rate (20 participants); and
- *Thinking about the impact of the task on their well-being*, including loss of sleep, with a 66% agreement rate (20 participants).

Emotional and Psychological Impact (RQ3)

The emotional response to performing undesirable tasks was overwhelmingly negative, with a high number of students (21, 70%) feeling unmotivated and losing interest in these tasks. Table 3 depicts the perceived impact of performing undesirable tasks, while Table 4 shows the frequency of these feelings.

Table 3: Overview of impact when performing undesirable tasks (1: Not at all – 5: Very much).

Emotional and Psychological Impact	1	2	3	4	5	AVG
Unhappy	0	10	6	8	6	3.3
Unmotivated	1	1	7	12	9	3.9
Stressed	4	9	8	5	4	2.9
Frustrated	2	3	9	12	4	3.4
Bored	2	3	8	7	10	3.7
I lose interest	0	2	7	11	10	4.0
I put little effort	4	9	10	4	3	2.8
I put off the task (procrastinate)	4	6	6	8	6	3.2
I think about leaving the group and/or course	15	6	2	2	5	2.2

Moreover, 16 (53%) of the participants felt frustrated, and 17 (56%) felt bored while performing undesirable tasks. Indeed, a significant number of students reported procrastination (14, 47%) and thoughts of leaving the group or course (7, 24%) in response to these tasks. However, leaving the group or course is not a recurring thought, as only 7% (2) of the participants had thought about it frequently. On the other hand, the most frequent feeling was unmotivated, with a 53% (16) rating.

Table 4: Overview of the frequency of feeling when performing undesirable tasks.

Emotional and Psychological Impact	1	2	3	4	5	AVG
Unhappy	4	11	6	5	4	2.8
Unmotivated	2	6	6	9	7	3.4
Stressed	5	9	7	7	2	2.7
Frustrated	4	9	5	8	4	3.0

(Continued)

Table 4: Continued.

Emotional and Psychological Impact	1	2	3	4	5	AVG
Bored	3	5	10	7	5	3.2
I lose interest	3	5	8	11	3	3.2
I put little effort	7	7	8	7	1	2.6
I put off the task (procrastinate)	6	6	7	5	6	3.0
I think about leaving the group and/or course	16	7	5	2	0	1.8

STRATEGIES USED FOR MANAGING UNDESIRABLE TASKS (RQ4)

Students employed various strategies to cope with undesirable tasks, with varying degrees of perceived effectiveness. The most reported strategies for managing undesirable tasks included:

Just Do It: The most common strategy was to perform the tasks directly, with 73% (22 participants) using this approach often or very often. This strategy was also among the most effective, with 60% (18 participants) reporting it as effective or very effective.

Collaboration and Communication: Working in pairs was another relatively effective strategy, used often or very often by 56% (17) of participants and reported as effective by 63% (19 participants). Keeping everyone informed was used frequently by 63% (19 participants) and found effective by 53% (16 participants).

Task Distribution: Less favored strategies included assigning tasks to the most inexperienced or on a first-come, first-served basis, which were used less frequently and reported as less effective.

Ultimately, asking others for help was perceived as one of the most effective strategies for handling undesirable tasks, with a 70% (21 participants) effectiveness rating.

DISCUSSION

This study provides preliminary insights into how students perceive undesirable tasks in their academic software development projects, how these undesirable tasks affect them, and the strategies they employ to tackle such undesirable tasks, along with their perceived effectiveness.

Masood et al. (2022) suggests that novice software developers and graduates are willing to work on all types of tasks, regardless their desirability, as they are interested in learning and gaining experience. However, it seems that learning and gaining experience alone does not appear to be enough, as our findings revealed that computer science students also dislike certain tasks, such as *bug fixing* and *writing documentation*, similarly to software developers. Both our findings and those reported in (Masood et al., 2022) suggest that undesirable tasks lead to negative emotional response, such as lack of motivation, boredom, and stress in both students and professional software developers. Interestingly, although software professionals in (Masood et al., 2022) view *working with others* as an undesirable task, students in our study did not share this perception. Our findings suggest

that, unlike software professionals, students in our study tend to prefer collaborative work, reflecting (human) dispositions involved in working with others to achieve a common solution (CC2020 Task Force, 2020). This is noteworthy because Computing Curricula 2020 (CC2020 Task Force, 2020) identifies knowledge, skills and dispositions as the three core dimensions of competency, with dispositions often considered the most challenging, as they are frequently overlooked in educational settings. Thus, a shift towards group-oriented and peer-supported learning environments in software development education could be beneficial.

Furthermore, the strategies employed to cope with undesirable tasks in both our study and (Masood et al., 2022) suggest a preference for tackling them directly, as avoidance is not seen as an option. Uniquely to our study, however, is the emphasis on collaboration, which was also considered more effective. Strategies, such as *passing tasks to the most inexperienced* or on a *first-come, first-served basis* were more commonly accepted and practiced in industrial contexts as reported in (Masood et al., 2022) whereas our findings suggest that these practices are uncommon among students. The strategies explored in this study should be incorporated into the software development curricula design to mitigate the negative impact of undesirable software development tasks, however, ways to make these tasks more desirable for students should also be identified.

AI technologies have the potential to contribute to making software development tasks more desirable, for instance, the use of AI-powered tools for documentation generation. Students, however, must remain aware of AI's limitations and challenges such as security and privacy risks, context-awareness issues, and over-reliance on automation. For back-end related tasks and bug fixing, we advocate for an AI-augmented approach where course design also incorporates new tools that boost hard skills (Sánchez-Gordón et al., 2023) while ensuring ethical, secure, and sustainable integration (Guimaraes & Nascimento, 2025). Education should align with real-world scenarios and prepare students for emerging technologies that meet industry demands (Sánchez-Gordón et al., 2023).

Our study has some limitations, primarily due to the small sample size comprising students from only one Norwegian higher education institution. Therefore, we do not claim generalization. Understanding of the survey responses is limited, as most open-ended questions received few or no answers. The only exception was the mandatory open-ended question on technologies/tools used in students' software projects. Despite these limitations, to the best of our knowledge, this is the first study in providing insight into software tasks (un)desirability in an academic context.

CONCLUSION

This study contributes to a better understanding of human factors influencing students enrolled in IT-related degree programs. The preliminary results identified undesirable software development tasks for students, such as

back-end related tasks, bug fixing and writing documentation. Moreover, results revealed that students cope with such tasks by employing strategies such as “Just Do It” and working in groups. These findings reveal an opportunity to design software development curricula that enhance learning experiences by incorporating strategies that not only mitigate the negative impact of undesirable software tasks but also make them more desirable for students. Future research could expand on this study by including a larger and more diverse sample from multiple institutions in a variety of geographical locations. Ultimately, research efforts can be dedicated to performing a longitudinal study design to track changes in software development tasks (un)desirability over time in academic and professional settings.

REFERENCES

- Afanasieva, T. V., Zagaichuk, I. A., & Zhelepov, A. S. (2019). Framework for Assessing Professional Growth of Software Developers. *Proceedings of the 2019 5th International Conference on Computer and Technology Applications*, 46–50.
- Bradley, N. C., Fritz, T., & Holmes, R. (2022). Sources of software development task friction. *Empirical Software Engineering*, 27(7), 175.
- CC2020 Task Force. (2020). *Computing Curricula 2020: Paradigms for Global Computing Education*. ACM. <https://doi.org/10.1145/3467967>
- Décieux, J., Mergener, A., Neufang, K., & Sischka, P. (2015). Implementation of the forced answering option within online surveys: Do higher item response rates come at the expense of participation and answer quality? *Psihologija*, 48(4), 311–326.
- Etemadi, V., Bushehrian, O., & Robles, G. (2022). Task assignment to counter the effect of developer turnover in software maintenance: A knowledge diffusion model. *Information and Software Technology*, 143, 106786.
- Guimaraes, E., & Nascimento, N. (2025). AI in the Software Development Lifecycle: Insights and Open Research Questions. *Proceedings of the 33rd ACM International Conference on the Foundations of Software Engineering*, 1353–1357.
- Kurian, R. M., & Thomas, S. (2023). Importance of positive emotions in software developers’ performance: A narrative review. *Theoretical Issues in Ergonomics Science*, 24(6), 631–645. <https://doi.org/10.1080/1463922X.2022.2134483>
- Lester, J. N., Cho, Y., & Lochmiller, C. R. (2020). Learning to do qualitative data analysis: A starting point. *Human Resource Development Review*, 19(1), 94–106.
- Licorish, S. A., & MacDonell, S. G. (2018). Exploring the links between software development task type, team attitudes and task completion performance: Insights from the Jazz repository. *Information and Software Technology*, 97, 10–25.
- Lima, D. L., De Souza Santos, R., Garcia, G. P., Da Silva, S. S., França, C., & Capretz, L. F. (2023). Software Testing and Code Refactoring: A Survey with Practitioners. *2023 IEEE International Conference on Software Maintenance and Evolution (ICSME)*,
- Masood, Z., Hoda, R., Blincoe, K., & Damian, D. (2022). Like, dislike, or just do it? How developers approach software development tasks. *Information and Software Technology*, 150, 106963. <https://doi.org/10.1016/j.infsof.2022.106963>
- Molléri, J. S., Petersen, K., & Mendes, E. (2020). An empirically evaluated checklist for surveys in software engineering. *Information and Software Technology*, 119, 106240.
- Nundlall, C., & Nagowah, S. D. (2021). Task allocation and coordination in distributed agile software development: A systematic review. *International Journal of Information Technology*, 13(1), 321–330.

-
- Russo, D., Hanel, P. H. P., Altnickel, S., & van Berkel, N. (2023). Satisfaction and performance of software developers during enforced work from home in the COVID-19 pandemic. *Empirical Software Engineering*, 28(2), 53.
- Sánchez-Gordón, M., Tovar, E., Colomo-Palacios, R., Piedra, N., & Castro, M. (2023). Educating Augmented Programmers. *Computer*, 56(12), 100–104.
- Scmidt, M. E. (2000). *Implementing the IEEE Software Engineering Standards*. Sams.
- Software Development Task Management Survey*. (2024, May 6). [Online resource]. Figshare; figshare. <https://doi.org/10.6084/m9.figshare.25757181>