UC Irvine

UC Irvine Previously Published Works

Title

Exploring GenAI in Software Development: Insights from a Case Study in a Large Brazilian Company

Permalink

https://escholarship.org/uc/item/59h2f5hs

Authors

Vaz Pereira, Guilherme
Jackson, Victoria
Prikladnicki, Rafael
et al.

Publication Date

2025-01-22

Copyright Information

This work is made available under the terms of a Creative Commons Attribution License, available at https://creativecommons.org/licenses/by/4.0/

Peer reviewed

eScholarship.org Powered by the California Digital Library
University of California

Exploring GenAI in Software Development: Insights from a Case Study in a Large Brazilian Company

Guilherme Vaz Pereira
PUCRS University
Porto Alegre, Brazil
guilherme.v003@edu.pucrs.br

Victoria Jackson
University of California, Irvine
Irvine, USA
vfjackso@uci.edu

Rafael Prikladnicki
PUCRS University
Porto Alegre, Brazil
rafaelp@pucrs.br

André van der Hoek
University of California, Irvine
Irvine, USA
andre@ics.uci.edu

Luciane Fortes
PUCRS University & Globo
Porto Alegre, Brazil
luciane.fortes@g.globo

Carolina Araújo
Globo
Porto Alegre, Brazil
carolina.araujo@g.globo

André Coelho
Globo
Rio de Janeiro, Brazil
andre.ribeiro.coelho@g.globo

Ligia Chelli
Globo
Rio de Janeiro, Brazil
ligia.chelli@g.globo

Diego Ramos
Globo
Rio de Janeiro, Brazil
diego.ramos@g.globo

***Abstract*— Recent progress in Generative AI (GenAI) impacts different software engineering (ES) tasks in software development cycle, e.g., from code generation to program repair, and presents a promising avenue for enhancing the productivity of development teams. GenAI based tools have the potential to change the way we develop software and have received attention from industry and academia. However, although some studies have been addressing the adoption of these tools in the software industry, little is known about what are developers' real experiences in a professional software development context, aside the hype. In this paper, we explore the use of GenAI tools by a large Brazilian media company that has teams developing software in-house. We observed practitioners for six weeks and used online surveys at different time points to understand their expectations, perceptions, and concerns about these tools in their daily work. In addition, we automatically collected quantitative data from the company's development systems, aiming at getting insights about how GenAI impacts the development process during the period. Our results provide insights into how practitioners perceive and utilize GenAI in their daily work in software development.**

***Index Terms*—Generative AI, Software Development, AI for SE, Industry Case Study.**

I. INTRODUCTION

Over the decades, Software Engineering (SE) has evolved its methods and practices to address both technical and non-technical challenges, resulting in the development and refinement of methodologies and techniques [1]. These advancements have been driven by continuous improvements in tools designed to overcome barriers in software development, enabling engineers to produce higher quality products with greater efficiency, lower costs, and faster delivery times [2].

Generative Artificial Intelligence (GenAI) tools have gained increasing attention within the software engineering community. These tools can assist in all phases of the software development lifecycle, from system analysis and design to maintenance [3]. According to Stack Overflow's 2024 annual developer survey, 76% of 60,907 respondents reported that they are using or plan to use AI during the year [4]. Additionally, several recent studies have explored the use of GenAI in various software development activities, e.g., generating architecture [5], source code generation [6], and program repair [7].

There are various GenAI-based tools to support software development. ChatGPT [8] Google Gemini [9] and GitHub Copilot [10] are some examples. They have the potential to improve software engineering productivity and help developers and companies increase their skills [11]. However, adoption in organizational environments can be challenging for individuals and teams, as it involves new approaches to working with and using evolving tools, sometimes without good practice guidelines on how to use them [12]. Furthermore, the expected improvements depend not only on the tools, but also on how people will use them to remove the current barriers in software development and take advantage of them [2].

The main goal of this study is to as well as to understand the expectations, perceptions, and concerns of them when adopting these tools in their daily work. We also intend to investigate for which tasks practitioners use the GenAI tools. In this context, this paper reports the findings from a study about the experience of the largest Brazilian media group. The company is adopting GenAI tools in its digital hub, including the use for software development activities. We aim to contribute to the Software Engineering community by sharing our observations on the use of GenAI within a real-world software development setting.

During the study, we collected data from practitioners over a two-month period. Over the time, we conducted online surveys at multiple time points to capture their expectations, perceptions, and concerns regarding the adoption of GenAI in their work. Additionally, we automatically collected data from their internal software development systems to better triangulate the findings.

Our findings provide insights on how GenAI tools are perceived and adopted in professional software development settings. While developers appreciate the potential applicability, particularly in code-intensive tasks, our results indicate limitations in creative and collaborative activities. Concerns about impacts on people and work, reliability of suggestions, security and privacy, and code quality and maintenance remain.

The reminder of this paper is organized as follows: Section II presents the related work on the use of generative AI tools within the software industry. Section III describes the study settings. Section IV presents the study results. The section V discusses threats to validation. Finally, Section VII brings some final considerations and futures directions.

II. RELATED WORK

Generative artificial intelligence (GenAI) tools have gained widespread adoption across various industries, including software engineering. For instance, ChatGPT [8] reached 100 million users within two months of its release, demonstrating the growing demand for AI-driven technologies. In software engineering, the LLM-based technology has the potential to cause a transformation [11] [3], prompting significant interest from both academia and industry in understanding their impact on software development.

A recent systematic literature review [13] identifies code generation as the most extensively researched topic in the context of GenAI tools. Studies exploring various dimensions such as correctness [6], performance [14], quality [15], robustness [16], and security [17] [18]. Beyond code generation, GenAI tools have been explored in various other software engineering domains, including program repair [7], code comprehension [19], non-coding activities [20], prompt engineering [21], productivity [22], SE education [23] [24], and the software development process (SDP) [2].

The practical aspects of generative AI for ES activities have also been explored. Barke [25] provides a grounded theory analysis of how programmers engage with AI-powered programming assistants like GitHub Copilot [10]. The study identifies two main interaction modes: acceleration and exploration. In acceleration mode, developers know what they want to achieve and use Copilot to speed up coding by filling in routine tasks. In exploration mode, programmers are unsure of how to proceed and use Copilot to explore different code options or APIs, relying on its suggestions for unfamiliar tasks. It highlights key benefits of Copilot, such as improving workflow speed in acceleration mode, while also pointing out challenges, including disruptions caused by overly long suggestions or irrelevant recommendations.

The application of GenAI in software industry settings has been a topic of interest for researchers and practitioners. Davila et al. [26] investigate the adoption of AI-based tools like GitHub Copilot and ChatGPT in a Brazilian agroindustry's software development team, focusing on their experiences, motivations, and challenges when using these tools. The findings reveal that AI-based assistants are primarily used to reduce programming effort, such as by speeding up searches for code snippets and recalling syntax. However, one of the main challenges identified is the lack of contextual understanding in the AI-generated code suggestions, requiring users to provide more detailed descriptions to improve outcomes.

Mendes [27] investigates the daily experiences of software developers using AI code assistants in real-world development environments. Based on interviews with 14 software engineers, the study reveals both the advantages and limitations of these. Some benefits identified are faster development through automatic completion and improved code quality. By contrast, challenges such as low accuracy in AI-generated code and frequent interruptions caused by persistent suggestions are pointed out.

Finally, Khojah [28] investigates the real-world usage of ChatGPT by professional software engineers. The study involved 24 participants across 10 organizations and analyzed their interactions with ChatGPT over a week, supplemented by an exit survey. The study identifies three main purposes for using ChatGPT: artifact manipulation (e.g., generating or modifying code), expert consultation (seeking advice or information), and training (learning new concepts). It offers recommendations for improving their usability in professional settings.

This research focus is relevant due to the constant advances in tools and the effect they can have on teams. Our case study in a large Brazilian company contributes to the body of knowledge on the use of GenAI tools by practitioners in their daily software development activities. It allows us to corroborate the findings reported in the literature, as well as contribute new insights by exploring the expectations, perceptions, and concerns of practitioners in a software development setting.

III. STUDY SETTINGS

This study was conducted by a multidisciplinary team, all authors of this paper, involving researchers and an internal team from the partner company, the largest Brazilian media company and one of the most important in Latin America. Its operations include a wide range of platforms, including free and pay television, radio, press, streaming and digital content.

For this study, we invited professionals from the company's Digital Hub. We collected data from June to August 2024. The study was based on two premises defined by the company: anonymity and non-interference in the software development process. All participants volunteered to take part in the study through an open call on the company's communication platform (Slack). They were asked to adopt GenAI into their workflow, including reporting on the tasks carried out with the support of GenAI on JIRA, their project management and issue tracking tool. They were also asked to answer the surveys designed for the study.

The company has over 1000 people who provide in-house software solutions to support all its digital products, such as publishing audio and video content, and a wide range of different applications. The technology stack is diverse and varies according to the business needs of each platform.

The GenAI-based assistant tool predominantly employed within the company, by its own decision based on business partnerships, was Google's Gemini Code Assist [9]. It has AI code assistance and a natural language chat interface where you can ask questions or receive programming guidance.

The data collection and data analysis are described in sections III-B and III-D, respectively.

A. Goal and Research Questions

The main goal of this study is to investigate for which tasks practitioners use the GenAI tools, as well as to understand the expectations, perceptions, and concerns of them when adopting these tools in their daily work. The following research questions were formulated to guide the study.

- **RQ1:** What are the expectations and perceptions of professionals about the impact of using GenAI in software development?
- **RQ2:** In which tasks do practitioners use GenAI?
- **RQ3:** How was the practitioners' experience using AI tools in the development of their daily activities
- **RQ4:** What concerns do professionals have about using GenAI in software development?

The first RQ aims to discover the practitioners' expectations regarding the possible gains of using GenAI in the professional software development setting, as well as their perceptions after using GenAI for a specific time. The aim of the second RQ is to understand the specific ways in which professionals are incorporating GenAI tools into their daily development workflow. The third RQ seeks to know about the overall experience of practitioners using GenAI. Finally, the fourth RQ explores the concerns raised regarding the use of GenAI in this context.

B. Data Collection

We applied a mixed methods approach to collecting data. Surveys at different time points were used to obtain qualitative data. Furthermore, quantitative data was collected automatically with records from the Gemini tool and from the JIRA tool [29].

At the beginning of the study, we applied a pre-survey. Likewise, at the end of the study, we applied a post-survey. In addition, participants were invited to complete a weekly (for six weeks) short survey with questions about their perceptions about the use of GenAI. Figure 1 illustrates this process. Table I shows the main focus, number of questions, and number of responses for each survey.

We used online questionnaires for all surveys. The data was collected anonymously, following the company's internal policy. The opening and closing questionnaires were designed by the researchers and reviewed by the company's internal team. The weekly surveys were designed by the company's internal team and reviewed by the researchers. Some questions were inspired by existing surveys [19] [30] [31].

TABLE I
APPLIED SURVEYS

Questionnaire	Aspects addressed	# Questions	# Responses
Pre-survey	Previous GenAI experience Expectations Concerns	20	66
Weekly-survey	Practitioner perceptions over the weeks	8	98*
Post-survey	Practitioners' perceptions Use of the tool Concerns after the period of use	20	23

*total number of answers.

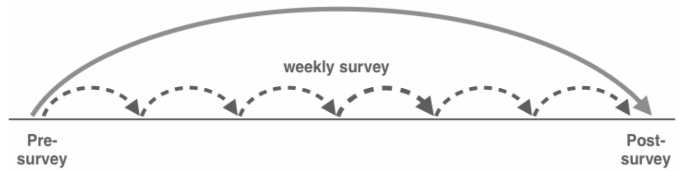


Fig. 1. Surveys application process.

1) Pre-survey: The pre-survey questionnaire aims to collect data on familiarity with the technology, as well as respondents' expectations and concerns about it. It begins by assessing the level of understanding and hands-on experience with LLMs and GenAI tools supporting software development. It then explores the use of GenAI in programming tasks as well as in non-coding software development tasks. Participants are also asked about their expectations of how GenAI will affect the speed, quality, collaboration, and creativity of their work in the near future. They are also asked if they have any concerns about using GenAI in their work. Finally, there are demographic questions to capture the profile of the participants.

2) Weekly-survey: Is a short survey with questions about how the experience was going over the preceding week. This questionnaire was inspired and adapted from a GitHub survey [31] about its AI developer tool, GitHub Copilot.

3) Post-survey: The post-survey was designed to get feedback about practitioners' experiences and perceptions of using GenAI assistance, how they used it (which tasks), and their concerns after this period. Participants were asked to rate their overall satisfaction with the tools, their suitability for workflow, the time it took them to feel comfortable using them and the quality and reliability of the suggestions provided by the AI. In addition, the questions explored the specific tasks in which GenAI was used. Participants were again asked about their concerns with GenAI in professional software development after using it for a period. Finally, the survey collected demographic information to contextualize.

The quantitative data was collected automatically and categorized into two main groups: (1) log data generated by the Gemini Code Assist tool and (2) data extracted from the JIRA Software tool. For JIRA, the development team applied custom tags to the tasks in which GenAI was used, allowing the tasks to be identified and tracked to obtain the Development Cycle Time [32] for each one.

The data collected from Gemini Code Assist includes the following metrics:

- Completion Method: This metric categorizes the type of operation performed by the generative AI tool. It is divided into three subtypes: Complete Code, Generate Code, and Transform Code.
- Full Acceptance Rate: This metric tracks the rate at which the generated code was fully accepted by the developer. In other words, it measures how often the developer adopted the AI's suggestion or code block without significant modifications.

C. Questionnaire Design

The questionnaires used in this study were planned to meet with the research objectives and answer the research questions (RQs) guiding our investigation, detailed in section III-A. The full set of questionnaires used in the study are available in the supplementary materials[1].

The weekly surveys, as stated in Section III-B uses a GitHub survey [31] with some extra questions. It has closed questions assessing specific aspects of tool usage and open-ended questions providing space for additional comments. Furthermore, it includes a closed question to capture the job title of the respondent.

The creation of the pre- and post-survey followed an iterative and incremental process. The research team refined the questions through multiple iterations of development, review, and adjustments. This allowed us to ensure that the questions were clear, relevant, and capable of capturing the necessary data for both qualitative and quantitative analysis.

Some questions in the post-survey mirrored those in the pre-survey to allow for a comparison between the participants' expectations and their perceptions. For example, both surveys included questions about how participants anticipated GenAI would impact software development speed, quality, and team collaboration.

Both pre-survey and post-survey instruments emphasize the anonymity of responses and state that the data will be used exclusively for research purposes. They then proceed with questions aligned to their specific objectives, concluding with demographic questions.

D. Data Analysis

To meet our objectives, we use both qualitative and quantitative analyses of data derived from surveys, as well as data collected from the tools, as detailed in Section III-B. Data was analyzed by the research team together with the company's internal team, in accordance with the organization's internal policies.

We developed Python scripts to support data manipulation and interpretation, generating graphs and tables for the analysis. The open-ended question answers, about practitioners' concerns, were collaboratively analyzed, discussed, and categorized into high-level topics by authors consensus. The scripts were used to streamline data processing, while the collaborative categorization of open-ended responses helped reduce the risk of analysis bias.

[1]Suplementary Materials - https://bit.ly/4dEuDnx

E. Participants and Demographics

Initially, 211 people signed up to be part of the study. Participation was entirely voluntary, with anonymity guaranteed in accordance with the company's confidentiality policies. The pre-survey and post-survey gathered demographic and professional data, including gender, years of experience, professional roles, and frequently used programming languages. The weekly surveys were shorter than others, thus focusing only on participants' professional roles.

1) Pre-survey: Sixty-six people answered this survey. Of these, 51 respondents (77.2%) identified as male, while 14 (21.2%) participants identified as female, and 1 participant preferred not to disclose their gender identification. No other gender identification was mentioned. Participants demonstrated considerable experience in software development, with 26 respondents (39.4%) reporting between 4 and 7 years of experience and 24 respondents (36.4%) having more than 8 years of experience. The job titles represented were diverse, spanning 12 distinct positions; however, the predominant roles were back-end (37.9%) and front-end developers (33.3%). The most frequently selected programming languages were JavaScript (41 responses) and TypeScript (32 responses), followed by Go (25 responses) and Python (23 responses), indicating a prevalence of web and back-end technologies among the respondents.

2) Post-survey: the participant demographics remained similar. Of the 23 respondents, 19 (82.6%) identified as male and 4 (17.4%) identified as female. No other gender identification was cited. The majority of participants again had significant experience, with 8 respondents (34.8%) reporting 4 to 7 years of experience and 9 respondents (39.1%) having over 8 years of experience. Similar to the pre-survey, the primary roles were back-end (39.1%) and front-end developers (26%), with 8 different job titles represented overall. JavaScript was the most frequently chosen programming language, with 14 occurrences, followed by Go and TypeScript, each with 10 occurrences. Python and Ruby were also selected, with 5 and 4 responses, respectively.

Participants had different levels of familiarity with LLMs, as summarized in Table II. In addition they reported that, in their previous experiences, they most frequently used GenAI tools for programming tasks such as modifying existing code, writing test cases, and learning new concepts. On the other hand, activities such as writing documentation and debugging code had less frequent usage. For non-coding tasks, GenAI tools were mainly applied to scenario creation and use case specification. However, the majority of the respondents indicated that they did not use GenAI tools for this type of activity.

ChatGPT was the most widely used tool among practitioners (58 responses), followed by Google Gemini Code Assist (46 responses) and GitHub Copilot (33 responses). Other tools,

TABLE II
PRACTITIONERS FAMILIARITY WITH LARGE LANGUAGE MODELS (LLMS)

Familiarity with LLMs	# Responses	Percentage (%)
I don't know what an LLM is	7	10.6
I've heard of the term LLM	6	9.1
I vaguely know how an LLM works	11	16.7
I can clearly explain what an LLM is, and name several	6	9.1
I have played with/experimented with one or more LLMs on my own	15	22.7
I've started testing the use of an LLM in my work	9	13.6
I routinely use an LLM as part of my work	12	18.2

such as Bard (12 responses) and Tabnine (9 responses), were also mentioned, though their usage was lower.

The variation in response rates across the surveys can be attributed to several factors. The duration of the study likely affected participant engagement over time, particularly in a voluntary setting where availability and interest can be variable. Additionally, the repetitive nature of the weekly surveys, despite their simplicity, may have contributed to reduced participation, as well as the sense of relevance of the survey content to the participants' specific roles.

IV. FINDINGS

In this section, we present the findings of our study, addressing the defined research questions. The findings are based on data collected from a combination of pre-surveys, post-surveys and weekly surveys, as well as quantitative data from Gemini and JIRA described in section III-B . Despite the varying number of responses, the data provides valuable insights on developers' relations with GenAI tools in real-world software development contexts.

A. RQ1: On Expectations and Perceptions On The Impact of GenAI

To answer the first research question (RQ1), we present our findings on software practitioners' expectations and perceptions of GenAI. We explore their perspectives (see Figure 2) regarding speed in the development process, software quality, and collaboration with team members.

Our participants were positive about the potential benefits the adoption of GenAI in software development would bring, with very high expectations about its potential to enhance SDP speed and software quality.

1) Software development process speed: Before GenAI, 98.5% (65/66) of respondents expected faster SDP speed, with 10.6% (7/66) of them expecting transformational changes. These expectations were broadly matched, as following the use of GenAI, the majority of participants (95.7%, 22/23) reported an increase in development speed. Of these, 2 people described the change as life-changing, while only 1 person indicated that there was no noticeable impact.

Quantitative data on cycle time automatically extracted from JIRA supports the practitioners' perceptions of SDP speed. The data indicates a reduction in development cycle time for tasks where GenAI was applied. The average cycle time for these tasks reduced by 23% based on the company's historical data.

2) Software quality: Around 75% (50/66) of participants expected software quality to improve with the introduction of GenAI, while only 9% (6/66) expected quality to worsen. Afterward, around 74% (17/23) of participants perceived quality had indeed improved, withonly 8.69% (2/23) thought it was transformative. 8.69% (2/23) of participants thought software quality had gotten worse.

3) Collaboration with team members: Expectations regarding team collaboration were modest, with most respondents not anticipating any significant impact from GenAI on collaborative dynamics. This perception remained consistent post-usage. Before and after using GenAI, the most common response was "nothing really changes". This perhaps reflects a viewpoint that GenAI is a tool designed to aid individual work rather than a tool for aiding collaborative activities.

Expectations and Perceptions of GenAI

High expectations for accelerating software development and enhancing quality were met with the majority of respondents reporting improvements in both areas after using GenAI, although these improvements were not considered transformative. Particularly, task data from JIRA shows a 23% reduction in cycle time. Contrastingly, there were low-expectations in terms of impact of GenAI on collaboration with most participants reporting no significant change in collaborative dynamics.

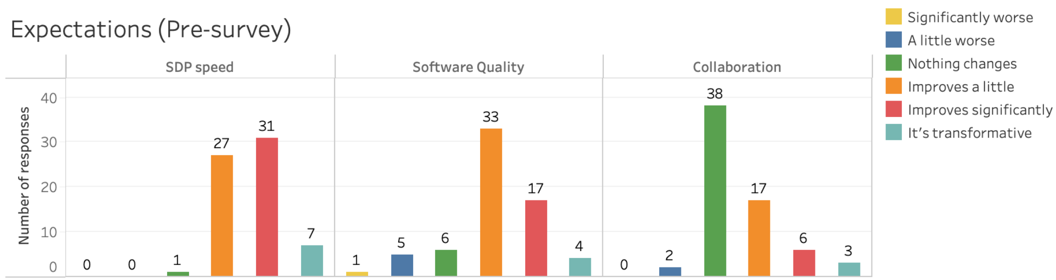
B. RQ2: In Which Tasks Is GenAI Used

We discuss how practitioners utilized GenAI throughout the study, focusing on specific development tasks and the frequency of use. The Figure 5 illustrates the distribution of responses indicating the frequency of GenAI use for each of the listed activities throughout the study.

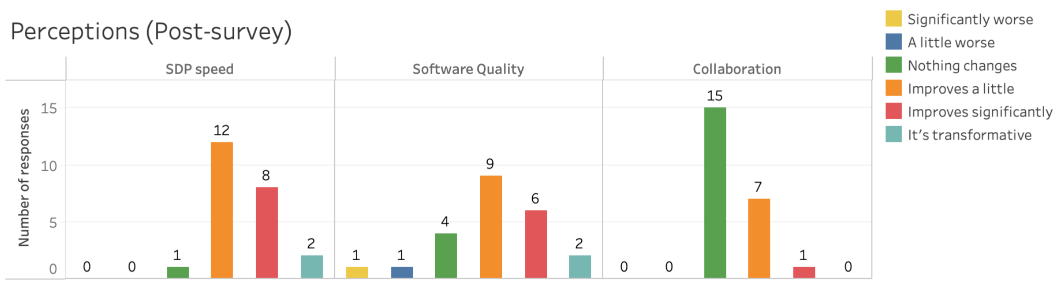
1) Coding related tasks: In considering the different types of tasks (Figure 5), coding related tasks were reported as the most frequently supported by GenAI. Specifically, "Writing new code", "Modifying existing code", and "Writing test cases" were the most common use cases, with a significant number of respondents reporting frequent or consistent use of GenAI in these areas. "Debugging code" was the coding task with the least AI support.

In terms of actual usage of the Gemini Code Assist tool, Figure 3 illustrates the number of suggestions generated for active users each week, as well as the developers' acceptance rate of these suggestions. The average acceptance rate for the period was 42.88% for 14,945 suggestions. Over the course of six weeks, there was a decline in the number of suggestions, while the acceptance rate increased slightly.

2) Non-coding tasks: In contrast, tasks involving collaboration with team members, such as "Whiteboard meetings". "Stand-up meetings" and "Teamwork in general" saw minimal



(a)



(b)

Fig. 2. (a) Respondents expectations on the impact of GenAI use on SDP speed, software quality, and collaboration between team members. (b) Respondents perceptions on the impact of GenAI use on SDP speed, software quality, and collaboration between team members.

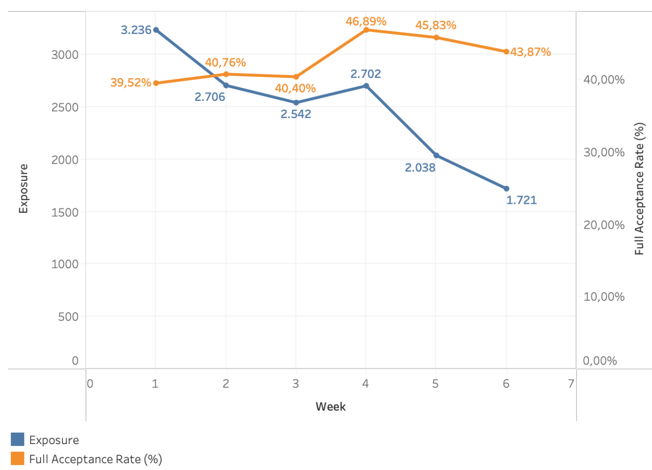


Fig. 3. Number of Gemini Code Assist suggestions and acceptance rate.

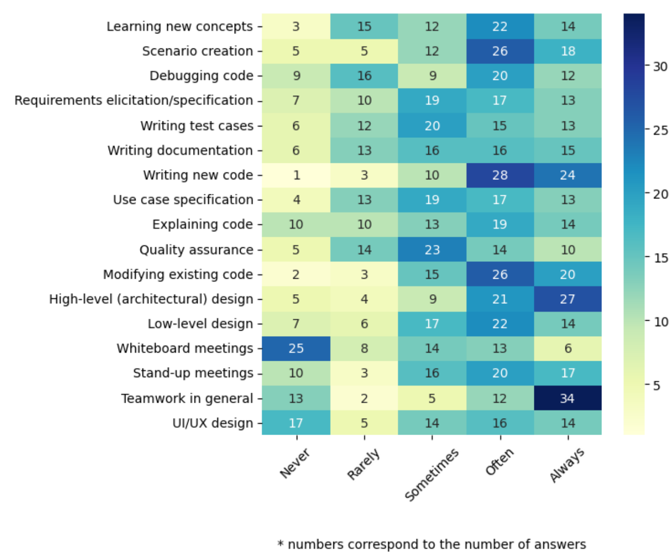


Fig. 4. Respondents perceptions on what tasks are considered creative.

or no GenAI usage. This aligns with the expectations discussed in Section IV-A3.

3) Creative tasks: When comparing tasks viewed as being creative (Figure 4) and the GenAI frequency of use per task type (Figure 5), we can see that tasks often considered as requiring creativity, such as "High-level (architectural) design", "Low-level design" and "UI/UX design" had limited GenAI influence reported. A similar attitude prevails for tasks related to analysis and design, such as "Scenario creation", "Requirements elicitation/specification", and Use case specification. The only tasks that are considered creative and are supported with frequent use of GenAI are "Writing new code" and "Learning new concept".

4) GenAI and Learning: Many respondents indicated they used GenAI to "Learn new concepts" suggesting that these tools are viewed not only as coding assistants but also as valuable resources for continuous learning and knowledge acquisition.

Note the profile of the respondents likely influences these findings, since the majority (78.2%, 18/22) of respondents work as programmers.

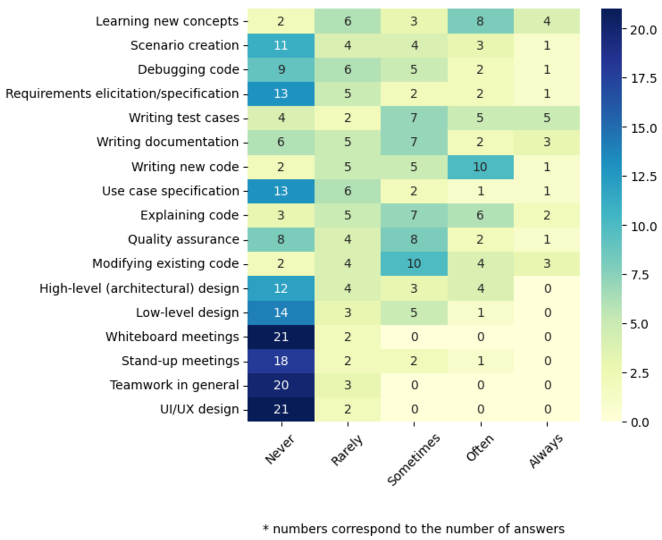


Fig. 5. Types of activities and GenAI usage frequency.

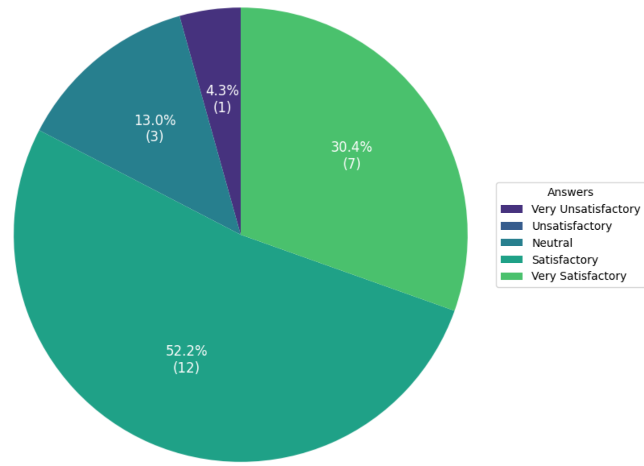


Fig. 6. Practitioners' overall experience with GenAI in software development.

Use of GenAI for software development tasks

GenAI was widely adopted for technical tasks, particularly those that are code-intensive. In contrast, its use in collaborative and creative work is limited. Although GenAI is proving valuable in specific areas of software development, there are still notable gaps in its applicability to human-centric practices. GenAI was helpful for learning purposes, highlighting the role of GenAI in enhancing developer skills and knowledge.

C. *RQ3: On the Practitioners' Experience Of GenAI*

In this subsection, we explore the overall experience of practitioners using GenAI tools. The findings offer insights into how these tools affected everyday development work.

1) Overall Experience: Figure 6 show that the overall experience about using GenAI was predominantly positive. According to the post-survey, 82.6% (19/23) of respondents rated their experience as either "Satisfactory" or "Very satisfactory" Additionally, 95.6% (22/23) reported that GenAI tools made their tasks easier, with responses ranging from "a little easier" to "transformative". In terms of integration, most respondents (86.9%) (20/23) indicated that GenAI tools were "Well" or "Very well" incorporated into their existing workflows. Further, 34.8% of practitioners reported feeling comfortable using GenAI tools within the two first weeks, while 47.8% took two to four weeks to feel comfortable.

2) Experience on quality and reliability of GenAI: However, there was neutral feedback on the quality of the suggestions and the reliability of GenAI tools for software development. Neutral responses were predominant for both questions, with "Medium" being the most common response to the question "How do you rate the quality of the suggestions provided by generative AI tools?" and "Moderately reliable" for "Do you consider generative AI tools reliable for supporting your development activities?". This neutral response perhaps indicates why the developers' average acceptance rate of Gemini suggestions was no more than 42.88% (see Figure 3): they did not find the quality or the reliability of the response acceptable.

3) Experience on using GenAI in other aspects of software development: As shown in Figure 7, our survey respondents had varying views on using GenAI in different aspects of software development

I complete tasks faster: A significant proportion (43.9%, 43/98) of respondents agreed that GenAI tools help them complete tasks more quickly, with 25.5% (25/98) strongly agreeing. However, 30.6% (30/98) disagreed or remained neutral. This suggests that while GenAI is perceived to offer productivity benefits for many, this experience is not universal. This finding supports the improvement to overall development speed calculated in JIRA tickets and noted in IV-A1.

I enjoy programming more: The responses show that 20.4% (20/98) of respondents agreed that their enjoyment of programming increased with GenAI, and 14.3% (14/98) strongly agreed. Nevertheless, 60.2% (59/98) remained neutral. This reasonable percentage of positive responses indicates that GenAI positively impacts the programming experience for some users, likely by automating tedious or repetitive tasks.

I learn from suggestions: Approximately 61.2% (60/98) of respondents agreed that they learn from the suggestions provided by GenAI tools. However, 34.7% (34/98) remained neutral, suggesting that while many find the suggestions educational, for others, the impact on learning is more limited.

I spend less mental effort on repetitive tasks: A total of 78.6% (77/98) of participants agreed or strongly agreed that GenAI reduces the mental effort required for repetitive tasks, with 39.8% (39/98) strongly agreeing. This finding reinforces the role of GenAI in automating time-consuming tasks, allowing developers to focus on more complex work.

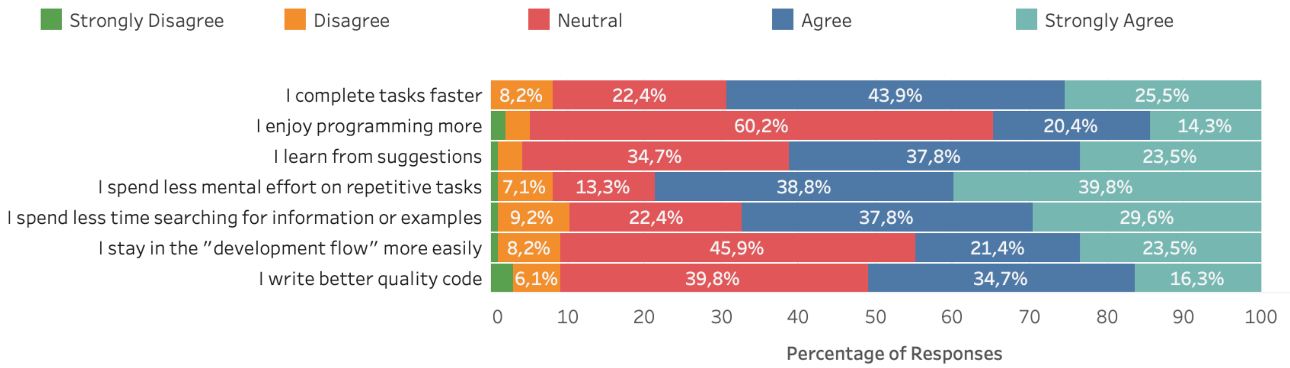


Fig. 7. Survey responses on impact of GenAI on various aspects of software development

I spend less time searching for information or examples: While 67.3% (66/98) of respondents agreed or strongly agreed that GenAI reduces the time spent searching for information or examples, a notable portion (22.4%, 22/98) remained neutral, and 9.2% (9/98) disagreed. This suggests that while GenAI is considered useful for retrieving information, its effectiveness may vary depending on specific use cases.

I stay in the "development flow" more easily: Many developers (44.9%, 44/98) reported that they stay in the "development flow" more easily when using GenAI. However, 45.9% (45/98) were neutral, indicating that GenAI's ability to help developers maintain focus varies significantly.

I write better quality code: On code quality, 51% (50/98) of participants agreed that GenAI helps them write better quality code, while 39.8% (39/98) remained neutral and 9.2% (9/98) disagreed. This reflects a more mixed perception of the impact on code quality, possibly due to the need for manual review and the varying reliability of suggestions.

Practitioners' Experience Of GenAI

The practitioners' overall experience with GenAI was positive, especially concerning task efficiency and workflow integration. However, feedback on the quality and reliability of GenAI suggestions was more varied. Most participants found the tools helpful in reducing mental effort in repetitive tasks and improving the speed of performing tasks. There are mixed opinions about the accuracy of the suggestions and their possible impact on code quality.

D. RQ4: On Concerns About Using GenAI

By exploring the key concerns raised by participants, we provide insights into the perceived risks and challenges that could impact the adoption of GenAI in the software industry.

Through an open-ended question in both the pre-survey and post-survey, participants expressed their concerns regarding the use of GenAI. 28% (25/89) of practitioners indicated they had no concerns regarding the use of GenAI. After analyzing the remaining responses (we excluded three as they were either invalid or the respondent had no opinion), we placed each into one of five identified categories: Code quality and maintenance, Impact on People and Work, Responsible Use, Reliability of suggestions, Security and Privacy:

- **Impact on People and Work:** Refers to concerns about how GenAI will affect developers' roles and their work.
- **Responsible Use:** Describes concerns around responsible use of GenAI in the workplace.
- **Security and Privacy:** Relates to concerns around data security and privacy, including the risk of exposing sensitive information when using GenAI tools, as well as the potential for unintentional copyright violations in AI-generated code.
- **Reliability of suggestions:** Regards the accuracy and consistency of GenAI suggestions, with concerns about the need for frequent human intervention, and on leading people to mistakes.
- **Code quality and maintenance:** Relates to the quality of the AI-generated code, e.g., the risk of generating low-quality or error-prone code that may increase technical debt and require frequent and costly maintenance.

Table III presents the percentage of responses per category. The largest concern related to "Impact on people and work" with "Code quality and maintenance" being the least concern.

1) Impact on People and Work: Concerns about the implications for practitioners' jobs were the most frequently cited. Several respondents expressed worries about an excessive dependence on AI tools, "*If developers start relying too much on AI, they may lose essential problem-solving skills*". In a similar way, another respondent remarked, "*The biggest concern is that people don't become dependent on the tool*".

Additionally, concerns about job security were raised, "*Management sees this as a potential justification for cutting costs by reducing the number of developers*".

Furthermore, the impact on junior developers was a recurrent theme, with many fearing that GenAI could hinder their learning process, "*GenAI is great for speeding up work,*

TABLE III
GenAI Concerns Raised By Survey Participants

Category	Responses (%)
Impact on people and work	29.5
Responsible use	24.5
Security and privacy	19.7
Reliability of suggestions	14.7
Code quality and maintenance	11.5

but I'm concerned that junior developers won't learn to code properly if they rely on it too much".

2) Responsible Use: Participants raised concerns that people should view GenAI as a complement to human expertise rather than become beholden to it. Thus, it is important people are responsible in using GenAI and consider responses critically, "*I worry that they think that GenAI can solve everything easily, forgetting that there are several complexities regarding business rules and architectural issues...the process remains human/machine.*"

3) Security and Privacy: There were significant concerns raised around security and privacy issues, particularly with the risk of exposing proprietary or sensitive data when using GenAI tools. One respondent stated, "*I am worried about the potential for data leakage when using GenAI in cloud environments. We need to ensure that sensitive information isn't inadvertently shared or exposed*". Another respondent pointed out the possibility of AI unintentionally generating code that infringes on copyright, especially when trained on public repositories ,"*How can we be sure that AI-generated code doesn't violate licensing or copyright terms?*".

4) Reliability: The reliability of GenAI's suggestions was also questioned, with people expressing doubt about the consistency and accuracy of AI-generated solutions. One practitioner noted, "*The suggestions provided by AI are helpful, but often require significant modifications, which reduces the time-saving benefits*". This highlights a core challenge: while GenAI can assist in streamlining some tasks, the necessity for manual review and modification can reduce potential productivity gains. Another response stated, "*I still worry about the reliability of the answers in relation to code implementation*".

5) Code quality and maintenance: Eight answers expressed concerns about the quality of code generated by GenAI tools. One respondent highlighted the risk of poor-quality output, "*AI-generated code often lacks the finesse and optimization that experienced developers provide, leading to a higher maintenance burden in the long term*". This sentiment was echoed by others, particularly regarding the potential for introducing bugs or low-quality solutions that may increase technical debt, especially for less experienced developers who may not have the expertise to critically assess the generated code. Another respondent noted: "*I believe that without a certain level of knowledge, the quality of the code tends to decline*".

Concerns About GenAI

The leading concern involves the potential negative impact on developers' roles and skills and how developers use GenAI in their daily work. For example, overreliance on GenAI could affect problem-solving skills, particularly for junior developers. Security and privacy risks, such as data exposure and potential copyright violations, are a concern. Code quality and impact on long-term maintenance were not mentioned often but one that required frequent human reviews to mitigate.

V. Limitations and Threats to Validity

Our study is subject to some threats to validity, which are discussed in this section to provide a clear understanding of the limitations.

External Validity: The results of this study are based on a single-company study. While this offers valuable insights into a real-world software development setting, the findings may not be generalized to other companies with different scenarios, such as other business domains (e.g., finance), distinct team structures, development processes, or the use of specific technologies (e.g., proprietary languages). Furthermore, our findings may not be transferable to other business domains (e.g., finance).

Conclusion Validity: The relatively small sample size, particularly in the post-survey, may affect the robustness. The variation between the number of responses to the different surveys indicates that some participants did not remain active throughout the study. At the same time, some quantitative data and findings are across many more participants and, as such, are more robust. Future studies with larger and more diverse samples are necessary to confirm the trends observed.

Construct Validity: Our study relied on questionnaires as the primary instrument for data collection, which may lead to varying interpretations of the questions. To mitigate this limitation, the questionnaires were passed through an iterative review process by the authors to ensure clarity and consistency.

While these threats to validity are acknowledged, we have taken actions to mitigate their potential impact on the study's findings. By clearly defining the study's context, refining the data collection instruments, and maintaining transparency in the analysis, we believe that the insights presented here provide a valuable contribution. Future research will be important for validating and extending the results.

VI. Discussion

Generative AI tools are a reality and are of widespread interest to the software industry and academia. This case study investigated the early-stage adoption of Generative AI (GenAI) tools within the largest Brazilian media company that develops in-house software systems. Drawing upon survey data on participants' perceptions and experiences of GenAI, task data from JIRA, and code completion data from Google Gemini Code Assist, we contribute findings on the impact of GenAI on software professionals.

We observe that among our survey respondents, **GenAI was predominantly used for coding tasks with little use in other areas of software development**, such as architecture, design, and requirements. The reasons for this skewed usage are unclear. We note that most of our participants were developers, which could bias our data. However, surveys on research on GenAI use in software engineering [13] have noted little research beyond coding and testing. This lack of adoption may reflect a perception that GenAI is less suited for more human-centered activities such as feature definition.

We note that **development speed improved quickly following adoption with cycle time reduced, on average, by 23% during the six-week study** compared to historical cycle time data. This actual reduction matched the perceptions of many participants who thought GenAI sped up development tasks. Our survey data shows this is likely due to spending less time searching for information and less mental effort on repetitive tasks. It could also be influenced by the speed at which our participants felt comfortable using GenAI, with nearly 75% comfortable within four weeks of adopting the tools. This early speed boost contrasts with Microsoft studies that note 11 weeks are required before benefits are experienced [33].

Similar to other studies (e.g., [34]), **GenAI was beneficial for learning activities**, with more than half of the survey participants highlighting that they learn from suggestions. Continual learning is important to develop expertise [35], so being able to learn while delivering faster helps both the individual and their employer.

Interestingly, **enjoyment of coding increased with the introduction of GenAI**. We did not delve into the reasons why, but we speculate it could be due to the automation of mundane work, such as GenAI's ability to quickly autocomplete partially written code [25], leaving time for the more enjoyable and creative parts of coding. Enjoyment of coding could help improve job satisfaction, potentially leading to improved individual productivity [36].

Finally, our participants noted concerns about GenAI, with **many concerns raised about the negative impact of GenAI on their work and role, with some expressing worries about job security**. These concerns align with more general studies of the impact of GenAI on the workforce, with some noting that the software profession is at high risk of substantial change due to GenAI [37]. Similar to other studies [26], data security and privacy concerns were also raised. Although unlike [26] that noted absence of ethical and legal concerns from study participants, legal concerns such as copyright concerns were mentioned by our participants, perhaps showing that awareness is increasing in the software profession of some ethical and legal issues with GenAI [38].

A. Implications for Practitioners

Our findings show that the deployment of GenAI is largely positive, with organizational benefits (shorter cycle times) and individual benefits (increased enjoyment of coding). Yet, we observe concerns from participants, especially about the impact on their work and role. We encourage organizations to foster an open and transparent dialog with their staff about such concerns and to find ways to address their concerns.

B. Implications for Researchers

Some findings warrant further investigation. Firstly, most participants were developers with little GenAI use reported outside coding. We suggest that researchers should explore how non-developer roles across the software development process can benefit from GenAI tools. Secondly, as mentioned by some of our participants, GenAI use may impact novice developers more than more senior developers such that novices become dependent on GenAI [39]. It would be beneficial to understand the relationship between developers of varying levels of seniority and the use of these tools. Moreover, this study was conducted in one setting, so analyzing teams in diverse contexts (legacy code, different industries) is essential, as findings may well be context-dependent. Finally, the reasons why GenAI use improves enjoyment of coding should be explored further to see if this can lead to increased job satisfaction. Previous research identifies a relationship between perceived enjoyment and intention to use GenAI [40].

VII. CONCLUSION

This study provides an overview of the use of GenAI tools in the software industry. Our results suggest that practitioners generally had a positive experience integrating GenAI into their daily work. However, some important concerns persist, mainly regarding the long-term impact on developers and their work processes, and about the responsible use of GenAI tools.

There are also concerns about how GenAI might affect skill development, particularly for junior developers, who may miss out on learning opportunities if they become too dependent on these tools. Additionally, the implications for collaborative and creative tasks remain unclear, as GenAI tools are still primarily used for technical and code-intensive activities.

Future work is needed to increase our understanding of the use of GenAI in the software development process. We intend to explore how non-developer roles across the software development process can benefit from GenAI tools. Understanding the relationship between developers of varying levels of seniority and the use of these tools, as well as analyzing teams in diverse contexts, will also be important areas of focus. In addition, we believe that future research is needed to further explore the reduction in development cycle time suggested.

ACKNOWLEDGEMENTS

We thank the partner company in this study. This paper was supported by the Ministry of Science, Technology, and Innovation of Brazil (Law 8.248 from Oct 23, 1991), within the scope of PPI-SOFTEX, coordinated by Softex, and published in the Residência em TIC 02 - Aditivo, Official Gazette 01245.012095/2020-56. Guilherme Pereira is supported by the Federal Institute of Education, Science and Technology of Rio Grande do Sul (IFRS). Rafael Prikladnicki is partially supported by CNPq in Brazil. This material is based upon work supported by the US National Science Foundation under grant CCF-2210812.

REFERENCES

[1] G. Booch, "The history of software engineering," *IEEE Software*, vol. 35, no. 5, pp. 108–114, 2018.

[2] I. Ozkaya, "The next frontier in software development: Ai-augmented software development processes," *IEEE Software*, vol. 40, no. 04, pp. 4–9, jul 2023.

[3] D. Russo, "Navigating the complexity of generative ai adoption in software engineering," *ACM Trans. Softw. Eng. Methodol.*, vol. 33, no. 5, jun 2024. [Online]. Available: https://doi.org/10.1145/3652154

[4] S. Overflow, "Stack overflow 2024 developer survey," 2024. [Online]. Available: https://survey.stackoverflow.co/2024/ai/

[5] A. Ahmad, M. Waseem, P. Liang, M. Fahmideh, M. S. Aktar, and T. Mikkonen, "Towards Human-Bot Collaborative Software Architecting with ChatGPT," in *Proceedings of the 27th International Conference on Evaluation and Assessment in Software Engineering*, ser. EASE '23. Association for Computing Machinery, 2023, pp. 279–285. [Online]. Available: https://dl.acm.org/doi/10.1145/3593434.3593468

[6] N. Nguyen and S. Nadi, "An empirical evaluation of github copilot's code suggestions," in *Proceedings of the 19th International Conference on Mining Software Repositories*, 2022, pp. 1–5.

[7] D. Sobania, M. Briesch, C. Hanna, and J. Petke, " An Analysis of the Automatic Bug Fixing Performance of ChatGPT ," in *2023 IEEE/ACM International Workshop on Automated Program Repair (APR)*. Los Alamitos, CA, USA: IEEE Computer Society, May 2023, pp. 23–30. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/APR59189.2023.00012

[8] O. AI, "Chatgpt," 2024. [Online]. Available: https://chatgpt.com

[9] Google, "Google gemini," 2024. [Online]. Available: https://gemini.google.com/app

[10] Github, "Github copilot," 2024. [Online]. Available: https://github.com/features/copilot

[11] C. Ebert and P. Louridas, "Generative ai for software practitioners," *IEEE Software*, vol. 40, no. 4, pp. 30–38, 2023.

[12] K. P. Agrawal, "Towards adoption of generative ai in organizational settings," *Journal of Computer Information Systems*, vol. 0, no. 0, pp. 1–16, 2023. [Online]. Available: https://doi.org/10.1080/08874417.2023.2240744

[13] X. Hou, Y. Zhao, Y. Liu, Z. Yang, K. Wang, L. Li, X. Luo, D. Lo, J. Grundy, and H. Wang, "Large language models for software engineering: A systematic literature review," 2024. [Online]. Available: https://arxiv.org/abs/2308.10620

[14] N. Nascimento, P. Alencar, and D. Cowan, "Artificial intelligence vs. software engineers: An empirical study on performance and efficiency using chatgpt," in *Proceedings of the 33rd Annual International Conference on Computer Science and Software Engineering*, ser. CASCON '23. USA: IBM Corp., 2023, p. 24–33.

[15] A. Moradi Dakhel, V. Majdinasab, A. Nikanjam, F. Khomh, M. C. Desmarais, and Z. M. J. Jiang, "Github copilot ai pair programmer: Asset or liability?" *Journal of Systems and Software*, vol. 203, p. 111734, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0164121223001292

[16] A. Mastropaolo, L. Pascarella, E. Guglielmi, M. Ciniselli, S. Scalabrino, R. Oliveto, and G. Bavota, "On the robustness of code generation techniques: An empirical study on github copilot," *arXiv preprint arXiv:2302.00438*, 2023.

[17] N. Perry, M. Srivastava, D. Kumar, and D. Boneh, "Do users write more insecure code with ai assistants?" in *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 2785–2799. [Online]. Available: https://doi.org/10.1145/3576915.3623157

[18] H. Pearce, B. Ahmad, B. Tan, B. Dolan-Gavitt, and R. Karri, "Asleep at the keyboard? assessing the security of github copilot's code contributions," 2021.

[19] D. Nam, A. Macvean, V. Hellendoorn, B. Vasilescu, and B. Myers, "Using an llm to help with code understanding," in *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, ser. ICSE '24. New York, NY, USA: Association for Computing Machinery, 2024. [Online]. Available: https://doi.org/10.1145/3597503.3639187

[20] A. Schmidt, "Speeding up the engineering of interactive systems with generative ai," in *Companion Proceedings of the 2023 ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, ser. EICS '23 Companion. New York, NY, USA: Association for Computing Machinery, 2023, p. 7–8. [Online]. Available: https://doi.org/10.1145/3596454.3597176

[21] J. White, S. Hays, Q. Fu, J. Spencer-Smith, and D. C. Schmidt, *ChatGPT Prompt Patterns for Improving Code Quality, Refactoring, Requirements Elicitation, and Software Design*. Cham: Springer Nature Switzerland, 2024, pp. 71–108.

[22] S. Imai, "Is github copilot a substitute for human pair-programming? an empirical study," in *2022 IEEE/ACM 44th International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, May 2022, pp. 319–321.

[23] E. Shein, "The impact of ai on computer science education," *Commun. ACM*, vol. 67, no. 9, p. 13–15, aug 2024. [Online]. Available: https://doi.org/10.1145/3673428

[24] S. Laato, B. Morschheuser, J. Hamari, and J. Björne, "Ai-assisted learning with chatgpt and large language models: Implications for higher education," in *2023 IEEE International Conference on Advanced Learning Technologies (ICALT)*, 2023, pp. 226–230.

[25] S. Barke, M. B. James, and N. Polikarpova, "Grounded copilot: How programmers interact with code-generating models," *Proc. ACM Program. Lang.*, vol. 7, no. OOPSLA1, apr 2023. [Online]. Available: https://doi.org/10.1145/3586030

[26] N. Davila, I. Wiese, I. Steinmacher, L. D. Silva, A. Kawamoto, G. P. Favaro, and I. Nunes, "An industry case study on adoption of ai-based programming assistants," in *2024 IEEE/ACM 46th International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*. Los Alamitos, CA, USA: IEEE Computer Society, apr 2024, pp. 92–102. [Online]. Available: https://doi.ieeecomputersociety.org/10.1145/3639477.3643648

[27] W. Mendes, S. Souza, and C. R. B. De Souza, ""you're on a bicycle with a little motor": Benefits and challenges of using ai code assistants," in *2024 IEEE/ACM 17th International Conference on Cooperative and Human Aspects of Software Engineering (CHASE)*, 2024, pp. 144–152.

[28] R. Khojah, M. Mohamad, P. Leitner, and F. G. De Oliveira Neto, "Beyond Code Generation: An Observational Study of ChatGPT Usage in Software Engineering Practice," *Proc. ACM Softw. Eng.*, vol. 1, no. FSE, pp. 1819–1840, Jul. 2024. [Online]. Available: https://dl.acm.org/doi/10.1145/3660788

[29] Atlassian, "Jira," 2024. [Online]. Available: https://www.atlassian.com/br/software/jira

[30] I. Shani and G. Staff, "The developer wishlist," 2023. [Online]. Available: https://github.blog/news-insights/research/survey-reveals-ais-impact-on-the-developer-experience/

[31] R. Salva, "Measuring the impact of github copilot," 2023. [Online]. Available: https://resources.github.com/learn/pathways/copilot/essentials/measuring-the-impact-of-github-copilot/

[32] M. Poppendieck and T. Poppendieck, *Lean software development: An agile toolkit: An agile toolkit*. Addison-Wesley, 2003.

[33] Microsoft, "Ai data drop: The 11 by 11 tipping point," 2024. [Online]. Available: https://www.microsoft.com/en-us/worklab/ai-data-drop-the-11-by-11-tipping-point

[34] J. T. Liang, C. Yang, and B. A. Myers, "A large-scale survey on the usability of ai programming assistants: Successes and challenges," in *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, ser. ICSE '24. New York, NY, USA: Association for Computing Machinery, 2024.

[35] S. Baltes and S. Diehl, "Towards a theory of software development expertise," in *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ser. ESEC/FSE 2018. New York, NY, USA: Association for Computing Machinery, 2018, p. 187–200. [Online]. Available: https://doi.org/10.1145/3236024.3236061

[36] M.-A. Storey, T. Zimmermann, C. Bird, J. Czerwonka, B. Murphy, and E. Kalliamvakou, "Towards a theory of software developer job satisfaction and perceived productivity," *IEEE Transactions on Software Engineering*, vol. 47, no. 10, pp. 2125–2142, 2021.

[37] R. Kochhar, "Which us workers are exposed to ai in their jobs — pewresearch.org," Pew Research, 2024. [Online]. Available: https://www.pewresearch.org/social-trends/2023/07/26/which-u-s-workers-are-more-exposed-to-ai-on-their-jobs/

[38] A. Poth, A. Wildegger, and D.-A. Levien, "Considerations about integration of genai into products and services from an ethical and legal perspective," in *Systems, Software and Services Process Improvement*,

M. Yilmaz, P. Clarke, A. Riel, R. Messnarz, C. Greiner, and T. Peisl, Eds. Cham: Springer Nature Switzerland, 2024, pp. 155–171.

[39] J. Prather, B. N. Reeves, P. Denny, B. A. Becker, J. Leinonen, A. Luxton-Reilly, G. Powell, J. Finnie-Ansley, and E. A. Santos, ""it's weird that it knows what i want": Usability and interactions with copilot for novice programmers," *ACM Trans. Comput.-Hum. Interact.*, vol. 31, no. 1, Nov. 2023.

[40] S. H. Y. Young Woo Kim, Min Chul Cha and S. C. Lee, "Not merely useful but also amusing: Impact of perceived usefulness and perceived enjoyment on the adoption of ai-powered coding assistant," *International Journal of Human–Computer Interaction*, vol. 0, no. 0, pp. 1–13, 2024.