

The Role of Generative AI in Software Development Productivity: A Pilot Case Study

Anonymous Author(s)

ABSTRACT

With software development increasingly reliant on innovative technologies, there is a growing interest in exploring the potential of generative AI tools to streamline processes and enhance productivity. In this scenario, this paper investigates the integration of generative AI tools within software development, addressing the need to understand their impact on software productivity in the real world. Through a pilot case study involving software practitioners working in different roles, we gathered valuable experiences on the integration of generative AI tools into their daily work routines. Our findings reveal a generally positive perception of these tools in individual productivity while also highlighting the need for addressing identified limitations. Overall, our research sets the stage for further exploration into the evolving landscape of software development practices with the integration of generative AI tools.

CCS CONCEPTS

• Software and its engineering;

KEYWORDS

software engineering, generative AI, LLMs, productivity

ACM Reference Format:

Anonymous Author(s). 2018. The Role of Generative AI in Software Development Productivity: A Pilot Case Study. In *Proceedings of the 32nd ACM Symposium on the Foundations of Software Engineering (FSE '24)*, November 15–19, 2024, Porto de Galinhas, Brazil. ACM, New York, NY, USA, 8 pages. <https://doi.org/XXXXXXX.XXXXXXX>

1 INTRODUCTION

Over the last decade, software engineering research has explored various aspects of teamwork in software development. Such investigations involved understanding how software engineers work together, how they collaborate to tackle problems, how they share information to complete tasks, their choices in adopting tools, and the obstacles they encounter in their activities [12, 25, 32]. Understanding these aspects is crucial for improving individual and team performance and ultimately achieving success in software processes. Despite these research efforts, understanding productivity in software development remains challenging due to its multifaceted nature [12, 17]. Productivity is influenced by technical, social, and psychological factors, making it complex to fully grasp.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

© 2018 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM

<https://doi.org/XXXXXXX.XXXXXXX>

Additionally, subjective metrics, diverse tasks, and evolving team dynamics add further layers of complexity [11, 26]. Therefore, despite progress in understanding teamwork in software engineering, unlocking the essence of productivity in software teams remains difficult [9, 11, 29].

Currently, the introduction of generative AI has elevated investigations into productivity in software development to a new level as discussions shift towards the incorporation of generative AI-based tools to enhance productivity, increase work efficiency, reduce errors in software tasks, and accelerate software production [8, 21, 22]. Notably, tools like GitHub Copilot have emerged as promising coding aids to improve code writing time. However, despite the general belief in the potential benefits that generative AI brings to software development, particularly regarding productivity gains, empirical evidence remains scarce, especially when considering the context of complex real-world projects, such as industrial settings [20, 24].

The lack of empirical evidence regarding the effectiveness of generative AI in complex real-world software development projects has motivated the present research. Hence, our goal was to explore how generative AI tools might impact the productivity of software professionals working on different roles and activities, including those focused on the delivery of software (e.g., coding and testing), supporting activities (e.g., management and IT infrastructure) and related tasks (e.g., data science). To this end, we investigated this phenomenon in a large software company. Specifically, the following research question guided this study: **How does the integration of generative AI tools influence the productivity of software professionals across different roles and activities?**

The remainder of this paper is organized as follows. In Section 3, we present a literature review on productivity and generative AI in software development. In Section 3, we present the case under study, our data collection, and our data analysis strategy. In Section 4, we introduce the key insights derived from the case, which are discussed in Section 5. Finally, Section 7 outlines our conclusions and plans for future work.

2 BACKGROUND

In this section, we explore previous research on productivity and generative AI in software development, specifically addressing the challenges associated with measuring productivity and integrating generative AI tools into software development tasks.

2.1 Challenges in Productivity Analysis

Understanding and measuring productivity among software developers has long been a challenge due to the absence of universally accepted metrics. The complex nature of development tasks, coupled with multifaceted dimensions of productivity, poses significant hurdles in devising accurate metrics. Factors such as individual

work styles, team dynamics, task complexities, and subjective perceptions contribute to the intricate web of challenges inherent in measuring developer productivity [9, 11].

Diverse industry discussions commonly refer to productivity as the relationship between output and input, but diverse fields adopt varying notions and measurement units [29]. However, looking specifically at software engineering, these conventional metrics, such as considering lines of code as the relationship between input and output, inadequately capture the essence of software development since this work thrives on collaborative efforts stemming from a diverse group of individuals, each contributing unique expertise [9].

Software developers operate within the domain of knowledge workers, where productivity lies in the exchange of ideas, knowledge, and skills, where collaboration becomes the cornerstone, fostering innovation and comprehensive solutions. [9, 28]. These encompass individual values, professional objectives, and the standards established by the organization. Moreover, their work revolves around activities that demand creativity as part of their responsibilities [14].

Historically, the evolution of productivity measures and discussions in software development reflects a quest for a comprehensive definition that considers multifaceted aspects influencing project outcomes [1]. Over the years, diverse methodologies emerged, from quantifying delivered code to embracing Agile principles, and the rapid evolution in technology continues to shape this quest, as changes in software practices and tools directly impact the productivity of software professionals [1, 3, 15, 33]. For instance, generative AI has recently emerged as a prominent topic, influencing the debate on productivity in software development.

Recently, expanding on several research insights, important dimensions of developers' productivity were highlighted, including [9]:

- **Satisfaction and well-being:** Focuses on happiness, job satisfaction, work-life balance, feeling valued, and a positive work environment. High levels of satisfaction boost productivity, creativity, and retention.
- **Performance:** Encompasses delivering high-quality work efficiently and accurately within set timelines. Metrics like lead time for changes, deployment frequency, and change failure rate gauge performance, impacting customer satisfaction and business outcomes.
- **Activity:** Evaluates the volume and nature of teamwork, such as completed tasks, lines of code, and commits. However, activity metrics shouldn't be the sole focus as they might not directly correlate with productivity or value delivered.
- **Communication and collaboration:** Assesses how well team members communicate, share knowledge, and work together towards common goals, vital for successful software development teamwork.
- **Efficiency and flow:** Efficiency involves optimizing workflows to maximize output by eliminating bottlenecks and unnecessary steps. Flow describes a state of full immersion, fostering creativity, focus, and efficiency among team members.

These dimensions challenge myths and misconceptions surrounding developers' productivity, demonstrating that only the combination of several metrics allows for a nuanced understanding of productivity in software engineering.

2.2 Productivity and Generative AI

Recent studies have shown that generative AI can enhance productivity in software development primarily through automated code generation [16]. In this context, large language models are trained on several code datasets to produce usable code in response to specific task prompts. While common chat interfaces can generate both code and natural language responses, tools designed specifically for code generation, such as GitHub Copilot, CodeWhisperer, and ChatGPT, have gained popularity among developers [4, 31].

GitHub Copilot, developed collaboratively by GitHub, OpenAI, and Microsoft, is described as an AI pair programmer, providing developers with real-time code suggestions based on the context of comments and existing code, supporting their productivity by minimizing disruptions and increasing focus on several aspects of programming [10]. Additionally, CodeWhisperer is an AI code generator by AWS that offers real-time code suggestions as developers write, anticipate the completion of lines of code, comments, or generate entire functions and code blocks, allowing faster task completion [2]. ChatGPT, developed by OpenAI, serves as a conversational AI and virtual assistant, helping developers in coding, debugging, and learning, ultimately enhancing productivity in software development [23].

Preliminary studies focused on the use of these tools and discussed their potential effects on the productivity of developers. In [24], the authors highlighted the positive correlation between the use of GitHub Copilot and the productivity of developers, with a focus on the automation of repetitive tasks. Following this, [13] expands this scenario and confirms the positive correlations while also recognizing the need for broader productivity metrics beyond coding time to obtain insights on developer satisfaction. Further, [7] also explores the speed gains offered by generative AI-based tools, acknowledging that these gains vary with task complexity and developer experience. Lastly, [36] offers a comprehensive review of AI integration trends in software development, including the increased role of AI in task automation and decision support.

The reports primarily focus on task completion time as a productivity metric, but productivity in software development entails complex factors beyond this [29]. In [20], a more comprehensive investigation details an application of ChatGPT in software development, covering the entire software creation process from requirements to deployment. Though productivity measurement is not the primary focus, the study offers insights into the nuanced challenges of measuring the effects of generative AI on productivity.

3 METHOD

Recognizing the limited empirical evidence from real-world settings regarding the use of generative AI to enhance productivity in software development, we chose a methodology centered on a case study [35]. Software engineering case studies analyze real-life settings, like companies or teams, using a systematic approach to collect and analyze data that can inform industrial practice [27]. In

this research, prior to conducting a comprehensive investigation considering the nuanced nature of productivity, we began with a simplified pilot case study [19, 34]. This pilot study aimed to gather insights from developers who were using these tools for the first time as an official working tool. By focusing on this initial experience from developers, we plan to gain a preliminary understanding of how we can explore the nuanced characteristics of productivity in this context. Further details regarding our methodology are outlined in the sections below.

3.1 The Case

The company selected for our case study was established in 1996 and specialized in on-demand software solutions across various sectors like finance, telecommunications, government, manufacturing, services, and utilities. With a workforce exceeding 1,200 professionals, over 70% are directly engaged in software development across 50 distinct teams. These teams comprise individuals from diverse technical backgrounds, including programmers, quality assurance (QA) specialists, and designers, who are proficient in popular software development methodologies such as Scrum, Kanban, and Waterfall and work to develop systems for global clients spanning North America, Latin America, Europe, and Asia. Additionally, the company employs several professionals in related fields like data science and supporting functions such as IT and human resources.

This company provides an excellent setting for our case study as it offers diverse tasks ranging from common coding activities to other crucial stages of the software development process. By exploring the impact of generative AI tools across these tasks, we can gain comprehensive insights. Moreover, given that various contextual factors and backgrounds influence productivity, the company's involvement in multiple industrial sectors allows us to assess productivity within varied contexts, enhancing the study's robustness and applicability. Additionally, the company's interest in experimenting with the use of generative AI tools provides an opportunity to leverage the insights gained towards devising processes and policies for their integration into projects.

3.2 Defining the Pilot Study Scope

Given the diverse array of projects, contexts, backgrounds, and professionals within the company, we chose to refine our research approach by initiating a preliminary investigation within a more focused subset. This entailed selecting a specific group of individuals to engage with AI generative tools and document their experiences. By focusing on this targeted group, we aimed to gather valuable insights that would guide us in the formulation of a broader, more extensive investigation within the case, exploring various productivity facets in software development.

During this initial phase, we obtained 17 licenses for various tools, including ChatGPT Plus, OpenAI API, Midjourney, and GitHub Copilot, for professionals to utilize in their work and provide feedback on these tools. In this pilot study, any interested professional could participate, provided they met three specific criteria: a) they hadn't regularly used generative AI tools in their work (priority given to those who hadn't used them at all); b) they received approval from their team manager to integrate the tool into their

tasks; c) they committed to reporting their experiences, including both positive and negative impacts on their perceived productivity.

After extending invitations to participate via the company's communication channels, we selected 14 volunteers who were either actively engaged in software development or working in supporting roles to join the pilot case study. These participants were chosen based on the aforementioned criteria, with diversity across backgrounds, experience levels, and project types considered to ensure broad representation. This approach aimed to capture a spectrum of perspectives and experiences, thereby enhancing the insights derived from the pilot study.

3.3 Data Collection

In line with the case study methodology, we employed various data collection methods to explore the case, including questionnaires with open-ended questions [18] and observations [30]. The primary data collection technique utilized was the questionnaire, which directly gathered insights from the professionals who volunteered to use the AI tools. Unlike traditional case studies that often rely on interviews, we opted for a questionnaire-based approach in this pilot study. This decision was made because the use of the tools was voluntary and not consistently integrated into the volunteers' daily work life or projects. We aimed to minimize interference with their work dynamics or team activities. Additional data were obtained from observing the company's communication channels, such as Slack, both to identify potential study volunteers and to explore what other professionals in the company were discussing about generative AI. This data collection process spanned four weeks during the final months of 2023.

The questionnaire (Table 1) was crafted to elicit descriptive answers, focusing on qualitative data from the study participants. Questions were designed to capture various aspects of their experiences with the provided AI tools, shedding light on the benefits, challenges, and overall impact on productivity and workload. For instance, participants were prompted to summarize their activities and experiences with the AI tool, identify the main benefits and difficulties encountered, and reflect on the tool's influence on productivity and value creation in their tasks. Moreover, the questions explored the participants' satisfaction with the results and their willingness to continue using the AI tool. Through these questions, we aimed to gather insights into the participants' experiences and perceptions regarding the use of generative AI tools in software development that could support us in the establishment of a comprehensive longitudinal case study.

3.4 Data Analysis

Our data analysis process involved examining responses to open-ended questions in the questionnaire and notes from observing interactions in communication channels. We employed coding strategies, including line-by-line open [5] supplemented by thematic analysis [6]. Through this approach, we identified recurring themes and patterns related to participants' experiences with AI tools, including benefits, challenges, productivity impact, satisfaction levels, and willingness to continue using generative AI tools in software development.

Table 1: Survey Questionnaire

1. This form is designed to gather feedback from team members who have been granted licenses to utilize Generative AI tools. The data collected will contribute to an Experience Report paper focusing on how Generative AI can enhance organizational productivity. Rest assured, the information provided will be handled with utmost confidentiality and security. Any materials derived from this study will utilize anonymized data, ensuring the privacy of respondents. No personal or sensitive information will be disclosed to third parties without prior consent. Do you agree to participate? () Yes
2. Which tool did you receive a license for? () ChatGPT () Open AI API () Midjourney () GitHub Copilot
3. What is your role in your team?
4. Please give a brief summary of the activities you performed with the provided AI tool.
5. Share details of your experience of working with the provided tools.
6. Explain the main benefits you encountered while using the provided AI tool.
7. Describe any impact you observe on your productivity while using any of the tools.
8. Did the tool contribute to enhancing the value of your activities? For instance, did it help you make progress in tasks or provide other valuable insights? If yes, how did it contribute?
9. You were satisfied with the results of the tasks performed with the assistance of generative AI?
10. Would you like to continue using AI in your work?
11. Describe any other important or interesting aspects of your experience.

4 FINDINGS

Initially, we had 14 individuals participating in our pilot study. However, one individual withdrew from the study for not being able to incorporate the tool usage in their daily activities considering the current phase of their project, resulting in a final group of 13 software professionals with various roles within software teams. It includes four software engineers, four software designers, three data scientists engaged in software projects, one software QA specialist, and one agile coach. All members actively participate in software development activities across different projects. Furthermore, our participants comprise an experienced cohort of professionals, with seven at a mid-level, two senior professionals, two principal professionals, and one technical manager, demonstrating a breadth of expertise and leadership within the field. Below, we present the findings derived from their experiences with the AI tools integrated into their regular activities within their projects. This encompasses insights into their utilization patterns, identified benefits, encountered challenges, and perceived impacts on productivity. In Table 2, we present evidence extracted from the participants' narratives that support our findings.

4.1 Generative AI Tools: Uses

The software professionals participating in our pilot study have utilized generative AI tools with a common objective: to acquire new knowledge across various aspects of their work. Among the cited uses, we identified:

- **Generating and Reviewing Artifacts:** Involves using AI tools to review, refine, and produce various project documents, such as requirements specifications, design documents, or project plans, ensuring accuracy and completeness in project deliverables.
- **Supporting Ideation Processes:** Entails harnessing AI to support the development of novel ideas, concepts, or solutions, aiding, for instance, in brainstorming sessions and design thinking.
- **Resolving Doubts in Code Construction:** Involves leveraging AI to assist in resolving technical issues encountered during programming activities.
- **Conducting Formal Writing:** Refers to utilizing AI-powered natural language generation tools to aid in the creation of formal project documentation, such as project reports, technical documentation, or client presentations.

Considering these findings, we can observe that the overarching goal in using generative AI in software activities remained consistent—using acquired knowledge to optimize work processes and facilitate problem-solving. Additionally, AI tools have played an interesting role in improving communication among stakeholders. These activities underscore a common thread of utilizing AI to streamline workflows and improve team capabilities across diverse tasks and domains.

4.2 Generative AI Tools: Benefits

In our analysis, we identified two main benefits highlighted by the participants, namely, time optimization and versatility. Participants in the study consistently highlighted the tool's support for time optimization as a primary benefit. By harnessing generative AI, software professionals were able to save time while completing their tasks. In particular, this time saving was apparent through the support for various activities that involve writing artifacts, such as reports. The tool's ability to generate coherent and relevant content and provide valuable insights and suggestions significantly supported participants' writing activities, enabling them to produce these types of artifacts with greater ease. Furthermore, participants highlighted the AI tools' versatility in supporting a wide range of software tasks as a visible benefit. The effectiveness of providing timely and relevant support across these diverse tasks was demonstrated to be a valuable advantage of the tools.

4.3 Generative AI Tools: Challenges

The participants reported challenges associated with utilizing generative AI in their software development, with reliability and refinement emerging as the most recurring issues. Software professionals described encountering difficulties in ensuring the reliability and functionality of generated responses, especially when attempting to use multiple questions simultaneously. They also faced challenges in crafting precise prompts to obtain objective and accurate responses, along with concerns about the absence of sources to reinforce the reliability of results. Additionally, participants highlighted the need for refinement and fine-tuning of the generated results to achieve optimal usage. Despite the AI's capability to produce responses, participants found that the outputs often required manual adjustments and polishing before they could be effectively

incorporated into their work. Finally, security measures emerged as a potential challenge, with at least two participants noting their inability to use the support of the tools with sensitive project data due to security constraints.

4.4 Generative AI Tools: Effects of Perceived Productivity

Aligned with the previous findings, participants reported a positive effect of generative AI tools on their perceived productivity. They highlighted how these tools facilitated efficiency gains in various aspects of their software development activities, thus primarily relating the optimization of time with productivity gains. One notable characteristic of this optimization was the consolidation of several individual tools into a single tool to perform various activities. Therefore, despite encountering challenges such as reliability concerns or limited outcomes, with the exception of one individual, software professionals mostly reported a positive impact on their perceived productivity.

On an additional note, software professionals related their increased productivity with the value that the AI tools incorporated into their work, especially in facilitating the creation of relevant and insightful content, whether reports, code, or design models. More specifically, the tools supported their productivity by contributing to learning and knowledge acquisition by providing quick access to information because, despite needing external verification, using these tools is much more productive than seeking information through queries in search engines.

5 DISCUSSIONS

We focused our discussions on the nature of software developers' productivity presented in the literature, particularly emphasizing that software developers operate within the domain of knowledge workers, where productivity lies in the exchange of ideas, knowledge, and skills. In this context, we compared how the positive impact of using generative AI tools reported by participants aligns with several dimensions of productivity, namely, satisfaction and well-being, performance, activity, communication and collaboration, and efficiency and flow.

Primarily, participants emphasized how these tools improve *efficiency and flow*, with gains being observed in various aspects of their software development activities. By optimizing time and consolidating multiple tools into a streamlined workflow, participants maximized the efficiency of their outputs with less effort. Moreover, the reported positive impact on productivity suggests that participants experienced an improvement in their *performance* as they were able to create relevant and insightful content, such as reports, code, or design models.

Additionally, while not explicitly mentioned by the participants, we understand that the use of generative AI tools can indirectly impact *communication and collaboration* within software development teams. By providing quick access to information and facilitating knowledge acquisition, these tools can enhance communication and collaboration by enabling team members to share insights and align their understanding toward common goals.

5.1 Implications

Our study has implications for research, as our pilot case study takes on a real-world perspective to explore the evolving landscape of software development practices with the integration of generative AI tools. As participants expressed positive experiences despite encountering challenges, they suggested that the benefits of these tools outweighed the drawbacks. Even though our findings are preliminary, they underscore the need for further investigation into the efficacy and impact of these tools across various dimensions of productivity. In particular, we highlight the need for research focused on refining these tools to address reliability concerns and expand their capabilities, particularly around the dimensions of productivity not identified in this study, e.g., activity and satisfaction.

Additionally, our study has implications for industrial practice, as it sheds light on the practical benefits of integrating generative AI tools into software development workflows, considering different professional roles, including programming, testing, and design. The positive experiences reported by participants indicate the potential of these tools to enhance productivity. Therefore, by addressing challenges and leveraging the advantages offered by generative AI, software companies can potentially optimize their development processes. Moreover, considering the challenges reported by practitioners regarding reliability concerns and usage difficulties, our findings suggest the importance of providing adequate training and support to facilitate the effective adoption of these tools within development teams, ensuring that they maximize their potential benefits while minimizing any associated risks.

5.2 Future Work

Following our pilot case study, our immediate future work involves conducting a comprehensive case study, capitalizing on the availability of the company that participated in this study. Our focus will be twofold. Firstly, we aim to explore the particularities arising from the utilization of generative AI tools across various software development roles, ranging from developers to QAs and designers, thereby gaining insights into how different professionals perceive and utilize these tools within their specific tasks. Secondly, we aim to further explore the relationship between generative AI tools and the dimensions of productivity by expanding our participant cohort within the case study to encompass a diverse range of project configurations and software development methodologies. By doing so, we aim to offer a more detailed analysis of the impact of generative AI tools on productivity across different software development contexts, thereby facilitating a more nuanced discussion on this subject.

6 THREATS TO VALIDITY

While our pilot case study provides valuable insights into integrating generative AI tools into software development workflows, some limitations inherent in the method must be acknowledged. Firstly, as a pilot study, our investigation involved a small number of participants from a single company, and our findings are not statistically generalizable to a broader population. Instead, we anticipate that researchers and practitioners can draw insights from our discussions,

Table 2: Evidence Obtained from Participants

Finding	Theme	Papers
Usage	Generating and Reviewing Artifacts	“The primary activity I engaged in with the tool was revisiting the questions from the TDD form.” (P01)
		“I used the tool to support desk research activities, to create questions for interviews.” (P06)
		“I used it to conduct a proof of concept for audio transcription.”(P12)
	Supporting Ideation Processes	“to experiment with ideation processes. I also used it to support the creation of presentation material.” (P06)
		“I used midjourney as part of the creative process, combining prompts and images we already use to generate new texture ideas.” (P07)
	Resolving Doubts in Code Construction	“I used the tool to solve some small doubts about Python code construction.” (P01)
		“I was able to evaluate how the tool can support data exploration processes, code generation, and others.” (P03)
		“The tool was used for code syntax research, automatic generation of simple algorithms, writing unit tests.” (P11)
	Conducting Formal Writing	“The use of ChatGPT has been very useful at the beginning of text production activities.” (P04)
		“Aiding in composing texts from topics, suggesting ideas for product and process names.” (P09)
Benefits	Time Optimization	“I have been using the OpenAI GPT-4 model for a variety of tasks, including text generation.” (P10)
		“It allowed me to be faster in text writing situations and debugging.” (P09)
		“The tool has proven to be useful in saving time and effort, allowing me to focus on more critical tasks.” (P10)
	Knowledge Acquisition	“The main gain was in development and research time.” (P12)
		“I believe the main benefit was what I learned.” (P03)
		“Initially, I sought information I was already familiar with to validate my knowledge.” (P05)
	Versatility	“It is a very powerful ally in consuming information, verifying and summarizing theories and approaches.” (P13)
		“I believe that using our own materials as data and making the AI generate new combinations.” (P07)
		“The flexibility and adaptability of the tool have been crucial aspects that enhance its value and applicability in various contexts.” (P10)
		“It is a much more productive way of seeking knowledge when compared to the model of using search engines (e.g., Google) up to that point.” (P13)
Challenges	Reliability	“Difficulty in verifying the reliability of some information.” (P06)
		“I faced challenges, mainly in ensuring that the generated responses are accurate and reliable, which sometimes requires manual review and adjustment.” (P10)
		“The absence of sources is one of the main barriers to the reliability of the results.” (P13)
	Precision	“Continuous use, makes the results less interesting.” (P04)
		“The main difficulties were in achieving more ‘usable’ results; all items generated still need refinement.” (P07)
		“Writing quality prompts for objective and correct answers.” (P09)
	Security	“The adoption of a model that I can use sensitive data from the company.” (P04)
		“I would like to add a crucial observation about the ethical and privacy challenges associated with the use of AI tools like GPT.” (P10)
		“The main difficulty was not exposing code used in clients.” (P11)
		“Absolutely. The value in the productivity and speed of generating results from my prompts is undeniable.” (P05)
Productivity	“Yes, the AI tool provided significant value in various areas of my activities. Firstly, it significantly improved my efficiency.” (P10)	

learn about our findings, and transfer the knowledge acquired from our pilot case study to their unique situations and contexts.

Additionally, the study focused primarily on participants' perceptions and experiences without comprehensive quantitative metrics to assess the impact of generative AI tools on productivity. Therefore, as with any qualitative research, there is a potential for researcher bias in data interpretation and analysis. To mitigate this threat to validity, we heavily relied on the raw reports provided by the participants, consistently comparing our interpretations with their views throughout our analysis.

Finally, the pilot nature of the study constrained the depth of data collection and analysis, preventing a thorough exploration of the topic. These limitations underscore the need for future research endeavors with broader and more varied samples, integrating both qualitative and quantitative approaches to offer a comprehensive understanding of the topic.

7 CONCLUSIONS

In this paper, we have explored the integration of generative AI tools into software development workflows, aiming to understand their impact on productivity from the perspective of software professionals. Our goal was to provide an understanding of how these tools can be utilized within real-world projects. Through a pilot case study involving software professionals, we collected insights into their experiences while integrating these tools into their daily work routines.

Our findings revealed a generally positive perception of generative AI tools among participants. These tools were particularly valued for their ability to streamline workflows through learning opportunities, optimize time, and facilitate the creation of relevant and insightful content. However, the practitioners reported challenges that mainly revolved around reliability concerns and difficulties in obtaining the desired outcomes, forcing them to manually fix the obtained outcomes for inaccuracies or inconsistencies in generated content.

In conclusion, our pilot case study provides insights into the integration of generative AI tools within software development practices. While our findings suggest promising benefits associated with the utilization of these tools, it is important to address the identified limitations through future research efforts to seamlessly integrate them into the software development process. Overall, our study sets the stage for continued exploration into the evolving landscape of software development practices with the integration of generative AI tools.

REFERENCES

- [1] A. J. Albrecht. 1979. Measuring Application Development Productivity. In *Proceedings of IBM Applications Development Symposium*. Monterey, 83.
- [2] Amazon Web Services. 2023. Amazon CodeWhisperer. <https://aws.amazon.com/codewhisperer/>. Accessed: 2023-12-10.
- [3] Barry Boehm et al. 2000. *Software Cost Estimation with COCOMO II*. Prentice Hall, Upper Saddle River.
- [4] Alexia Cambon, Brent Hecht, Benjamin Edelman, Donald Ngwe, Sonia Jaffe, Amy Heger, Mihaela Vorvoreanu, Sida Peng, Jake Hofman, Alex Farach, et al. 2023. *Early LLM-based Tools for Enterprise Information Workers Likely Provide Meaningful Boosts to Productivity*. Technical Report. MSFT Technical Report. <https://www.microsoft.com/en-us/research>
- [5] Kathy Charmaz. 2014. *Constructing grounded theory*. sage.
- [6] Daniela S Cruzes and Tore Dyba. 2011. Recommended steps for thematic synthesis in software engineering. In *2011 international symposium on empirical software engineering and measurement*. IEEE, 275–284.
- [7] McKinsey Digital. 2023. Unleashing developer productivity with generative AI. <https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/unleashing-developer-productivity-with-generative-ai>. Accessed in Mar 22, 2024.
- [8] Christof Ebert and Panos Louridas. 2023. Generative AI for software practitioners. *IEEE Software* 40, 4 (2023), 30–38.
- [9] Nicole Forsgren, Margaret-Anne Storey, Chandra Maddila, Thomas Zimmermann, Brian Houck, and Jenna Butler. 2021. The SPACE of Developer Productivity: There's more to it than you think. *Queue* 19, 1 (2021), 20–48.
- [10] Github. 2021. GitHub Copilot. <https://copilot.github.com>. Accessed on November 23, 2023.
- [11] Marcela Guerrero-Calvache and Giovanni Hernández. 2022. Team productivity in agile software development: a systematic mapping study. In *International Conference on Applied Informatics*. Springer, 455–471.
- [12] Martin Hoegl, K Praveen Parboteeah, and Hans Georg Gemuenden. 2003. When teamwork really matters: task innovativeness as a moderator of the teamwork–performance relationship in software development projects. *Journal of Engineering and Technology Management* 20, 4 (2003), 281–302.
- [13] Eirini Kalliamvakou. 2022. Research: quantifying GitHub Copilot's impact on developer productivity and happiness. <https://github.blog/2022-09-07-research-quantifying-github-copilots-impact-on-developer-productivity-and-happiness/>. Accessed in Mar 22, 2024.
- [14] Young-Ho Kim, Eun Kyoung Choe, Bongshin Lee, and Jinwook Seo. 2019. Understanding personal productivity: How knowledge workers define, evaluate, and reflect on their productivity. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–12.
- [15] B Lakhanpal. 1993. Understanding the factors influencing the performance of software development groups: An exploratory group-level analysis. *Information and Software Technology* 35, 8 (1993), 468–473. [https://doi.org/10.1016/0950-5849\(93\)90044-4](https://doi.org/10.1016/0950-5849(93)90044-4)
- [16] Hongxin Li, Jingran Su, Yuntao Chen, Qing Li, and ZHAO-XIANG ZHANG. 2024. SheetCopilot: Bringing Software Productivity to the Next Level through Large Language Models. *Advances in Neural Information Processing Systems* 36 (2024).
- [17] Yngve Lindsjörn, Dag IK Sjøberg, Torgeir Dingsøyr, Gunnar R Bergersen, and Tore Dybå. 2016. Teamwork quality and project success in software development: A survey of agile development teams. *Journal of Systems and Software* 122 (2016), 274–286.
- [18] Jefferson Seide Molléri, Kai Petersen, and Emilia Mendes. 2016. Survey guidelines in software engineering: An annotated review. In *Proceedings of the 10th ACM/IEEE international symposium on empirical software engineering and measurement*. 1–6.
- [19] Cleviton VF Monteiro, Fabio QB da Silva, and Luiz Fernando Capretz. 2016. The innovative behaviour of software engineers: Findings from a pilot case study. In *Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*. 1–10.
- [20] Mauricio Monteiro, Bruno Castelo Branco, Samuel Silvestre, Guilherme Avelino, and Marco Tulio Valente. 2023. End-to-End Software Construction using ChatGPT: An Experience Report. *arXiv preprint arXiv:2310.14843* (2023).
- [21] Daye Nam, Andrew Macvean, Vincent Hellendoorn, Bogdan Vasilescu, and Brad Myers. 2024. Using an llm to help with code understanding. In *2024 IEEE/ACM 46th International Conference on Software Engineering (ICSE)*. IEEE Computer Society, 881–881.
- [22] Shakked Noy and Whitney Zhang. 2023. Experimental evidence on the productivity effects of generative artificial intelligence. *Science* 381, 6654 (2023), 187–192.
- [23] OpenAI. 2023. ChatGPT: Optimizing Language Models for Dialogue. <https://openai.com/blog/chatgpt>. Acessado em 10 de dezembro de 2023.
- [24] Sida Peng, Eirini Kalliamvakou, Peter Cihon, and Mert Demirel. 2023. The impact of ai on developer productivity: Evidence from github copilot. *arXiv preprint arXiv:2302.06590* (2023).
- [25] Rafael Prikladnicki, Yvonne Dittrich, Helen Sharp, Cleidson De Souza, Marcelo Cataldo, and Rashina Hoda. 2013. Cooperative and human aspects of software engineering: Chase 2013. *ACM SIGSOFT Software Engineering Notes* 38, 5 (2013), 34–37.
- [26] Daniel Rodriguez, MA Sicilia, E García, and Rachel Harrison. 2012. Empirical findings on team size and productivity in software development. *Journal of Systems and Software* 85, 3 (2012), 562–570.
- [27] Per Runeson and Martin Höst. 2009. Guidelines for conducting and reporting case study research in software engineering. *Empirical software engineering* 14 (2009), 131–164.
- [28] Anastasia Ruvimova, Alexander Lill, Jan Gugler, Lauren Howe, Elaine Huang, Gail Murphy, and Thomas Fritz. 2022. An exploratory study of productivity perceptions in software teams. In *Proceedings of the 44th International Conference on Software Engineering*. 99–111.
- [29] Caitlin Sadowski and Thomas Zimmermann. 2019. *Rethinking productivity in software engineering*. Springer Nature.

- [30] Carolyn B. Seaman. 1999. Qualitative methods in empirical studies of software engineering. *IEEE Transactions on software engineering* 25, 4 (1999), 557–572.
- [31] Samarth Sikand, Kanchanjot Kaur Phokela, Vibhu Saujanya Sharma, Kapil Singi, Vikrant Kaulgud, Teresa Tung, Pragya Sharma, and Adam P Burden. 2024. How much SPACE do metrics have in GenAI assisted software development?. In *Proceedings of the 17th Innovations in Software Engineering Conference*. 1–5.
- [32] Diane Strode, Torgeir Dingsøyr, and Yngve Lindsjorn. 2022. A teamwork effectiveness model for agile software development. *Empirical Software Engineering* 27, 2 (2022), 56.
- [33] Claes Wohlin and Mattias Ahlgren. 1995. Soft factors and their impact on time to market. *Software Quality Journal* 4, 3 (1995), 189–205. <https://doi.org/10.1007/bf01351923>
- [34] Qing Xie and Atif M Memon. 2008. Using a pilot study to derive a GUI model for automated testing. *ACM Transactions on Software Engineering and Methodology (TOSEM)* 18, 2 (2008), 1–35.
- [35] Robert K Yin. 1994. Discovering the future of the case study. Method in evaluation research. *Evaluation practice* 15, 3 (1994), 283–290.
- [36] Beiqi Zhang, Peng Liang, Xiyu Zhou, Aakash Ahmad, and Muhammad Waseem. 2023. Practices and challenges of using github copilot: An empirical study. *arXiv preprint arXiv:2303.08733* (2023).