Information and Software Technology 186 (2025) 107805

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

Still just personal assistants? – A multiple case study of generative AI adoption in software organizations

Kai-Kristian Kemell [a,*], Matti Saarikallio [b], Anh Nguyen-Duc [c], Pekka Abrahamsson [a]

[a] Tampere University, Department of Information Technology and Computer Science, Tampere, 33014, Finland
[b] Independent researcher, Finland
[c] University of South-Eastern Norway, Bø I Telemark, 3800, Norway

ARTICLE INFO

Keywords:
Large language model
Software engineering
Case study
Generative AI
Artificial intelligence
Technology adoption
Empirical research

ABSTRACT

Context: Generative AI (GenAI) is argued to transform software engineering (SE) in various ways, and GenAI tools show promise for various SE tasks. Software organizations across the globe are currently exploring the use of GenAI for SE.

Objective: While numerous studies have recently been published on GenAI, few studies have looked at the adoption of these tools and their usage from an organizational point of view, focusing instead on individual users. Our objective is to understand how organizations adopt these tools and what their impacts are in industrial contexts, with a focus on the European perspective.

Method: We conducted a multiple case study of seven European companies. We collected data through semi-structured interviews (n=15), as well as through longitudinal observation in one case company. All data were analyzed using thematic analysis.

Results: We analyzed 28 transcripts, resulting in 456 quotations and 557 code occurrences split between 66 individual codes that were categorized under 6 high-level themes. We identified 25 types of tasks GenAI was currently being used for in our case organizations. We identified 12 benefits for GenAI in SE and 10 adoption and use challenges. Key adoption challenges for organizations include data privacy and legislative concerns, the emerging and fast-moving market of GenAI tools, difficulty of measuring the positive impact of the tools, and potential change resistance. For individuals, the key challenges are related to prompting, such as understanding what a good prompt is, and how to write prompts for specific tasks.

Conclusion: GenAI adoption is becoming widespread in SE, but good practices and use cases are still emerging. While GenAI can potentially produce various benefits in SE, companies and individual users are facing various challenges in making the most of GenAI in SE. Overall, GenAI is still primarily used as a personal assistant.

1. Introduction

Software organizations globally are currently looking to leverage Large Language Models (LLMs), and Generative AI (GenAI) more generally, for Software Engineering (SE). According to Gartner surveys, GenAI was the most deployed AI solution in organizations in May 2024 [1], and 79% of corporate strategists saw “AI and analytics as critical to their success over the next two years” [2]. In terms of software development, a 2024 GitHub survey comprising primarily developer respondents reports that “almost everyone (upwards of 97%) reported that they have at some point used these [AI] tools both in and outside of work” [3], although not specifically in relation to GenAI.

AI in SE overall is hardly a novel research area, with various AI tools developed and studied for various tasks [4], including code generation [5,6], software testing [7,8], requirements engineering [9], and debugging-related tasks [10,11]. The recent rise of GenAI [12], however, has resulted in a surge of interest in AI for SE [13].

LLMs in particular are highly versatile tools [12–14] and can potentially be used for a wide variety of tasks in SE [13,15]. Moreover, recent advances in GenAI have resulted in levels of performance that in particular have sparked interest in GenAI. For example, ChatGPT-4 is argued to reach human-level performance in programming [16], and many studies point towards various benefits produced by LLMs in SE, including increased developer productivity. Furthermore, the high availability [17] of high-performing LLMs as software-as-a-service makes them even more attractive.

Currently in SE research, much emphasis has been placed on LLM performance in specific tasks [15], and on the experiences and productivity of individual developers, often in relation to a specific LLM [18,

* Corresponding author.
E-mail address: kai-kristian.kemell@tuni.fi (K.-K. Kemell).

https://doi.org/10.1016/j.infsof.2025.107805
Received 14 September 2024; Received in revised form 6 May 2025; Accepted 28 May 2025
Available online 11 June 2025

19]. An organization-focused perspective is comparatively lacking, and little is known about the adoption process of LLMs, especially from an organizational point of view. Despite widespread developer interest, companies are largely still adjusting to LLMs and exploring their options.

This gap is highlighted in existing literature. In a survey, Fan et al. [15] call for more studies on understanding how LLMs can be integrated into SE workflows and not just used in isolation, pointing to a need for studies more focused on industrial practice. Ani et al. [20] summarize that most research on GitHub Copilot has been “focusing on four main areas: developer productivity, code quality, code security and education”. Nguyen-Duc et al. [21] call for more studies evaluating LLMs in industrial contexts instead of experimental or classroom settings.

In this paper, we conduct an industrial multiple case study. This study starts addressing a number of gaps identified in existing literature on GenAI in SE. First, by utilizing industrial, empirical data, we begin addressing the call for more industrial studies [14,21]. Second, we study the perspective of both organizations and individual users, to better understand how GenAI and LLMs are or can be integrated into SE workflows [15]. Third, we explore the adoption of GenAI in industrial settings, from the point of view of both organizations and individual users, tackling the call for more studies on “the potential barriers and challenges to adoption of GenAI in software-intensive businesses” [21]. Finally, we do not focus on any specific tool, whereas many existing studies focus on a specific GenAI tool, such as GitHub Copilot [20]. We consider this a gap as these tools are often used in conjunction out in the field, as we observe in this study.

Specifically, we conduct an exploratory, multiple case study of seven case companies. We collect data by means of semi-structured interviews, interviewing both management/executive-level respondents and those working in developer roles. In addition, we collect longitudinal observation data from one case by observing the regular meetings of a group of developers piloting GitHub Copilot over the course of eight months. We utilize thematic analysis to analyze the resulting data. Through this study, we seek to tackle the following Research Questions (RQs):

- **(RQ1:)** What challenges do software organizations face when adopting GenAI, and particularly LLMs, for SE?
- **(RQ2:)** What are the benefits of GenAI, and particularly LLMs, in SE, for both organizations and individual users within the organizations, in industry settings?
- **(RQ3:)** How can GenAI, and particularly LLMs, be utilized in industrial SE settings, and how do they change SE in these contexts?

The novel contributions of this study include: (a) the use of industrial data, (b) our longitudinal observation data on adoption, (c) our tool-agnostic approach, and (d) our focus on organizational adoption perspective in addition to individual users' adoption. This study thus addresses key gaps noted in existing literature, namely the lack of empirical, real-world insights on integrating GenAI into SE workflows [15, 21], as well as the lack of studies examining GenAI adoption from the point of view of organizations [21].

The rest of this paper is structured as follows. In Section 2, we discuss existing literature. In Section 3, we discuss the research methodology of the study. In Section 4, we present our results. In Section 5, we provide a summary of our results. In Section 6, we discuss the implications of our results. In Section 7, we discuss threats to the validity of this study. Section 8 concludes the paper.

2. Background

In this section, we discuss existing literature relevant to this study. In Section 2.1, we discuss GenAI and LLMs. In Section 2.2, we discuss their utilization in SE specifically.

2.1. Generative AI and large language models

GenAI as briefly summarized by Ebert & Louridas [13] “can synthesize—or generate—the answers to the questions you pose”. GenAI as a concept broadly refers to different types of machine learning models that generate different types of content, such as text, audio, images, and video. Common approaches include transformer-based models for text [13], but the concept of GenAI also includes approaches such as Generative Adversarial Networks (GANs) and diffusion models for image or video generation [22].

LLMs are a subset of GenAI and focus on the processing of natural language to generate a wide variety of content [13,23]. LLMs are neural network models [13,23], most of which currently rely on the transformer architecture [24], particularly Generative Pretrained Transformers (GPT) or Bidirectional Encoder Representations from Transformers (BERT) [13]. LLMs have recently been the focus of the discussion in Generative AI in the context of SE, though other types of GenAI are potentially of interest as well [13,25]. For example, image generation can support user interface design [13,25].

Recent advances have resulted in levels of LLM performance [16] that have sparked much discussion in both academia and media. Services such as ChatGPT,[1] Copilot,[2] Gemini,[3] etc. offer LLMs as SaaS, requiring little more than a user account to start utilizing. LLMs are also highly versatile [12,13,23], making them attractive to companies.

When used in practice, LLMs are utilized via user inputs referred to as prompts [26]. Prompts are written in natural language, and while at its core an LLM is merely intended to continue a user input (prompt) with a “plausible output”, more recent LLMs are able to perform various tasks based on prompts [26]. In popular media and in recent literature, the act of honing prompts is often referred to as *prompt engineering* [27]. How to write effective prompts for certain purposes has recently received much attention.

2.2. Large language models in SE

In a survey on SE research on LLMs, Fan et al. [15] identify the following areas of SE where LLMs have been studied to varying degrees: requirements engineering, design & planning, code implementation, testing, maintenance, and deployment, listing also various more specific tasks within these areas. Nguyen-Duc et al. [21] provide an overview of research on Generative AI in SE, with a focus on providing future research suggestions. We also identify various surveys with more specific foci. These include a survey on LLM usage for software testing [28], a survey on LLM evaluation [29], and a survey on research related to GitHub Copilot specifically [20].

Many existing SE studies have focused on a specific LLM (e.g., GitHub Copilot [20,30]) or on comparing the performance of different LLMs for a certain task (e.g., [31]). In addition to producing content such as code, LLMs can be helpful in understanding existing code [32]. Many studies have also explored LLMs in SE education contexts [20,33, 34].

Overall, existing research points towards the following. LLMs are useful in a wide variety of SE tasks [13,15,21,23]. They are particularly useful in code generation, testing, and debugging [15,21]. They are argued to increase developer productivity notably [13,14,19,35], and to produce various other benefits for developers, such as aiding with knowledge acquisition [14].

More profound impacts are also envisioned. Ebert & Louridas argue that GenAI “has the potential to change the software profession more than any other recent technology” [13]. Similar sentiments are put

[1] https://openai.com/.
[2] https://copilot.microsoft.com/.
[3] https://gemini.google.com/.

Table 1
Overview of case companies.

ID	Size	Country	Company focus
A	Large, 1500+	Multinational, Nordic	Software and consultancy for various industries
B	SME, 25+	Finland	Development for Microsoft platforms
C	Micro, 5-	Finland	Management consultancy *(respondent focused on experiences with current client company software project)*
D	SME, 200+	Norway	Software for energy industry
E	Large, 500+	Multinational, Nordic	Software and consultancy for various industries
F	Micro, 5-	Norway	Education technology
G	SME, 25+	Finland	Digital games

forward in other papers. For example, it is argued that LLMs may shift development work from writing code to reviewing it [36].

In terms of related work, we identify a handful of studies with similar points of view that are based on practitioner data. Simaremare & Edison [37] also explore GenAI adoption by software practitioners, though focusing only on individual adoption experiences. Russo [38] similarly studies GenAI adoption through individual users' adoption experiences, but highlights implications for organizations. Ulfsnes et al. [39] study how GenAI has changed SE workflows, focusing on individual users as well. Coutinho et al. [14] conduct a case study on the productivity impacts of GenAI in industrial SE contexts. Finally, Dolata et al. [40] specifically focus on freelancer dvelopers. Our study contributes to this nascent body of knowledge by focusing on organizational adoption in addition to individual user adoption, by also including the viewpoints of management and executives.

3. Study design

This study is an exploratory case study. More specifically, this is a qualitative case study that takes on a more interpretive research approach [41]. Case studies were originally used primarily for exploratory purposes and are still considered particularly well-suited for it [41]. Due to the novelty of the topic and viewpoint regarding it, we chose such an exploratory approach in favor of a more positivist one testing hypotheses.

We discuss our research methodology in more detail in the rest of this section. Section 3.1 covers our data collection approach and our cases, while in Section 3.2 we discuss the data analysis.

3.1. Data collection

This study is a multiple case study comprising seven cases (Table 1). In selecting our cases, we wanted to (a) include companies of varying sizes (from micro to large multinational), and (b) include companies from different countries. This was done to make our results more representative. However, all our case companies were European and primarily based in Finland or Norway. In terms of size, they represent a varied set of cases.

We utilized semi-structured interviews (Table 2) to collect data from all seven cases. These interviews are discussed in more detail in Section 3.1.1. In addition to the interview data, we collected longitudinal observation data from Company A, which is discussed in more detail in Section 3.1.2. In brief, this data was collected via regular meetings and can be likened to group interviews.

Case A was an in-depth case study. In addition to a higher number of interviews covering multiple business units of the company, we collected longitudinal observation data from Case A. This approach allowed us to explore the GenAI adoption process within Company A in great detail. The data from the six other cases has provided us with additional perspectives on GenAI adoption, which have either served to validate our findings from Company A, or provided contrasting experiences to enrich our data.

3.1.1. Interview data collection (All cases)

For the semi-structured interviews (n = 15; Table 2), we involved a wide range of respondents. As we utilized semi-structured interviews to collect data, we devised an interview instrument for the interviews. This instrument is included in its entirety in Appendix. For Company A, we involved respondents working in a number of different roles in order to gain a comprehensive view of the organization. For the other cases, we prioritized management and executive-level respondents with a good understanding of the current situation of the company in relation to GenAI adoption and usage.

For Case B, we conducted a group interview with the CEO and CTO. For Case C, we interviewed the program manager of a client's project, focusing on their experiences within that project. For Case E, we interviewed the head of a branch office, who was aware of company-level decisions and had a good understanding of the situation in their branch. For Case F, we interviewed the CEO of a startup. For Case G, we interviewed a co-founder who currently works as a developer but is still closely involved in organizational decision-making. These were all respondents with a good view of their respective organizations due to their positions. For Case D, we intended to conduct additional interviews with management, but were unable to secure further interviews. While our respondent could discuss some company-level decisions, they did not participate in organizational decision-making. We nonetheless included this case to add the viewpoint of one more organization.

The interviews were conducted between September 2023 and December 2024. All data collection was conducted remotely via video conferencing tools, including the meetings observed in Company A. Only the audio was recorded for analysis. We utilized an AI speech recognition software (OpenAI Whisper [42]) to produce preliminary transcripts of all the recordings, which we then manually cleaned up where necessary. We scheduled 1-h interviews with the respondents. The durations varied in practice, as some respondents had more experiences or knowledge to discuss, while some were simply less talkative. The only exception was Case E, where the respondent only agreed to a 30 min interview.

For the remainder of this subsection, we will summarize the interview instrument and justify the question choices. All the questions in the instrument were generally asked of all respondents. The only exception was that we did not ask category D questions of all respondents in Company A. For Company A, these questions were asked of executives, and supplementary data was acquired through observation. Depending on the respondent, further questions were focused on personal tool experiences or on the situation within their company.

The background questions (Category A) were intended to provide further information about the respondents. Questions related to the usage of GenAI (Category B) were intended to provide an overall picture of the current situation of GenAI adoption within the case organization. We also asked whether the company used other AI tools to understand their experiences with AI overall.

Question Category C further explored the current situation and experiences with GenAI in the company. These questions were partially informed by existing research. First, how to best utilize GenAI in SE is still an open question, especially in industry contexts [21], so we asked about current use cases. Second, some studies indicate that the

Table 2
Interviews by respondent. *C* refers to case company ID. *Exp (years)* refers to relevant job experience in years *(categories: 0–2; 3–5; 6–10; 11–20; 21+ years)*.

C	Respondent	Exp. (years)	Dura-tion
A	Operations executive	11–20	49:28
A	Operations executive	11–20	53:25
A	Software developer	6–10	44:07
A	Senior software developer	6–10	26:10
A	Senior software developer	6–10	1:00:36
A	Senior software developer/scrum master	11–20	53:30
A	Senior software developer	11–20	41:25
A	Systems architect	21+	59:39
A	Senior software developer	11–20	37:50
B	CEO and CTO *(2 participant group interview)*	21+ & 11–20	49:18
C	Founder *(of own company)*/program manager *(of client's project)*	21+	47:15
D	Senior data scientist	6–10	36:25
E	Head of a branch office	11–20	30:25
F	CEO	11–20	35:23
G	Co-founder/game developer	11–20	38:32

Table 3
Pilot group participants at Company A.

#	Job title
1	Junior software developer
2	Scrum master/senior software developer
3	Senior system architect
4	Senior software developer
5	Scrum master/software developer
6	Software developer
7	System architect

performance of LLMs may vary depending on project context (e.g., programming language [43]), so we asked about the project contexts in the organization. Third, existing research (e.g., [39,44]) has studied the sentiments of individual developers towards LLMs and their impact on their work, so we also asked questions about this to supplement these existing findings with more industrial data.

Question Category D was focused on company-level perspectives. These questions focused on the reasons behind GenAI adoption, possible long-term plans for AI, and the impacts of GenAI. As GenAI is argued to have transformational potential in SE [13], we wanted to explore what kind of changes may have already taken place.

3.1.2. Observation data collection (Company A only)

In addition to interviews, we collected observation data from Company A through a pilot group. In August 2023, Company A set up a pilot group of seven developers who began to use GitHub Copilot. The participants of this pilot group are outlined in Table 3. In addition to the seven pilot users, these meetings were attended by the first and second authors.

This pilot group met regularly, initially bi-weekly (Aug–Dec 2023), and later monthly (Jan–Mar 2024). Thus, a total of 13 pilot group meetings were held. Each meeting lasted 30 min.

Each meeting started with a summary of any updates regarding GenAI usage and adoption plans in the company by management. Afterwards, each participant was asked to share their own experiences and updates since the previous meeting, in order to facilitate conversation among the participants. For Company A, the motivation behind this pilot group was to gather internal data for wider tool adoption later, by leveraging any lessons learned by the participants. For this study, this pilot group allowed us to observe LLM adoption in a longitudinal manner.

3.2. Data analysis

We utilized thematic analysis to analyze all our transcripts. We utilized an inductive coding approach due to the novelty of the phenomenon, as suitable theoretical frameworks for the context are still emerging. The process was carried out in an iterative fashion, with codes, themes, and sub-themes emerging and being refined throughout the process. The analysis was carried out by the first author, using ATLAS.ti.[4]

The purpose of this approach was to discover commonalities and discrepancies between the seven cases, while providing an overview of the data through the codes and themes. As a summary, our analysis yielded six highest-level themes: (1) challenges with LLMs, (2) LLM adoption (organization), (3) LLM adoption (user), (4) LLM benefits (user), (5) LLM use cases (tasks), and (6) potential future plans (organization).

There were 456 quotations across 28 transcripts. These quotations were allocated a total of 557 code occurrences. These occurrences were split between 66 individual codes, thus with an average of approximately 8 occurrences per individual code. The highest number of occurrences for a single code was 36, while the lowest was 1. The full results of the thematic analysis (codes, subthemes, themes, and their occurrences) are available as an external resource on figshare: https://doi.org/10.6084/m9.figshare.28593032

4. Results

In this section, we present the results of our study as follows. In Section 4.1 we provide an overview of our cases in terms of GenAI adoption situation. In Section 4.2 we discuss factors related to GenAI adoption from the point of view of *individual users* in SE. In Section 4.3 we discuss factors related to GenAI adoption from the point of view of *organizations* in SE.

4.1. Case overview

We provide an overview of the situations of the companies in relation to GenAI adoption and usage to provide context to our findings. The situations of the companies differed in various ways, including which tools were being used, how they were being used, and how they were being adopted in practice.

Company A started a top-down adoption process of GitHub Copilot in 2023, and after the piloting concluded in March 2024, Company A began to increasingly push their developers to adopt GitHub Copilot across business units, while exploring other LLM tools as well. In addition to LLM use in software development, Company A also explored LLM use for other purposes, such as sales and customer support.

Company B. At the time of the interview, Company B was starting to adopt LLM tools. GitHub copilot was being piloted with a number of developers, and many developers outside the ones selected to pilot it

[4] https://atlasti.com.

expressed interest towards getting a license as well. Enterprise license options for services were starting to appear at the time, and Company B was weighing the benefits with the cost of enterprise licenses, leaning towards purchasing in the near future.

Company C was working with a client company who were working on a large software project for a public Finnish institution. Together with their client company, Company C was exploring the use of LLMs within that development context in June 2024 when the interview was conducted. Development in the project was being done fully offline due to security concerns, rendering LLM cloud services unusable. The project involved hundreds of documents that occasionally had to be changed to match changes made to the code, and developers were having trouble grasping these connections. The LLM was intended to help with this.

Company D was providing its developers with a license for GitHub Copilot when the interview was conducted. ChatGPT use was permitted, and developers could be reimbursed for any subscription fees if they purchased a personal one for work use. Guidelines were provided in relation to data privacy, but not in terms of how to use the tools effectively.

Company E was giving their developers nearly full freedom in terms of which LLM tools to adopt. At the time of the interview, no enterprise licenses were provided and the company was still exploring its options and how to best leverage the tools. Employees were reimbursed for any work-related LLM service subscriptions.

Company F was an early-stage startup working on its first product. The company was actively seeking to leverage AI in different ways to help with the initial product development with limited (personnel) resources. AI was seen as something that could help the company *"punch above its weight class"* [CEO (Case F)] in terms of productivity. The company was using GenAI internally, with ChatGPT primarily used for summaries of meetings, GitHub Copilot being used for programming, and Wondershare Virbo being used for marketing, in addition to the company incorporating ChatGPT-based AI features into its own product as well.

Company G was a former startup, now established digital game company working on its second game. The company was actively leveraging ChatGPT for various tasks, including both programming-related tasks, product-related content creation, and community management. ChatGPT licenses had been provided within the company since early 2024. Additionally, the company had been leveraging Midjourney since 2022 to aid with the production of art assets for its products.

To summarize, the companies exhibited different approaches to adopting LLM tools:

- Company A: Top-down, company provides licenses and decides what tools are to be used.
- Company B: Top-down, company provides licenses and decides what tools are to be used.
- Company C: Top-down, project-specific use case for local tool.
- Company D: Mixed, company provides some licenses but employees use other tools at their own discretion as well.
- Company E: Bottom-up, company reimburses subscriptions, employees can choose tools they want.
- Company F: Mixed, company provides some licenses but employees use other tools at their own discretion as well.
- Company G: Mixed, company provides some licenses but employees use other tools at their own discretion as well.

To provide further context, we also list all the GenAI tools utilized by the companies. This is relevant because, on many occasions, our respondents reflected on how different tools could complement each other in their work. The tools used by the case companies were almost exclusively LLMs for most case companies, though Companies F and G also utilized GenAI for image or video generation. These tools, by company, were (in alphabetical order):

- Company A: ChatGPT, Copilot (formerly Bing Chat), GitHub Copilot, Microsoft 365 Copilot
- Company B: ChatGPT, Copilot (formerly Bing Chat), GitHub Copilot
- Company C: LLaMA 3
- Company D: ChatGPT, GitHub Copilot
- Company E: ChatGPT, GitHub Copilot, Microsoft 365 Copilot
- Company F: ChatGPT (also via API), GitHub Copilot, Wondershare VIBO
- Company G: ChatGPT, Midjourney

4.2. User adoption of GenAI in SE

In addition to exploring GenAI adoption from the point of view of organizations, we have also explored it from the perspective of individual employees, focusing primarily on software development roles. This is relevant because LLMs are currently primarily being used as personal assistants for individual employees in SE, in addition to any potential process-focused applications. This makes individual and organizational adoption intertwined in practice.

4.2.1. User adoption process of GenAI in SE

We will start this section by summarizing the adoption experiences among the pilot group participants in Company A. Overall, initial adoption was slow in the pilot group. For the first months, the respondents largely treated GitHub Copilot as something separate from "real work", and interest in experimenting with the tool varied between participants. Aside from the occasional useful auto-complete, the respondents considered GitHub Copilot to be something they had to go out of their way to use. Even though GitHub Copilot was integrated into their IDEs, the respondents felt that it was initially not a natural part of their work. Over the course of the pilot group, the participants started to increasingly consider GitHub Copilot a part of their work. Moreover, some of the developers began to adapt the way they worked to make better use of GitHub Copilot.

We highlight the following observations based on data from both the Company A pilot group and all of our interviews. First, underwhelming initial experiences seem common. Especially with GitHub Copilot and its auto-complete, many of the respondents reported underwhelming initial impressions. They could not get the tool to understand what they wanted to do next and they felt that many of its suggestions were poor. The respondents reported problems trying to get the AI to fix its own outputs even when they pointed out what was wrong with them.

Second, prompts feel difficult to evaluate, and it is challenging to discernibly improve at prompting. Despite some generic lessons learned among the respondents, most respondents found it difficult to gauge what was a good prompt and what was not, making this a recurring challenge even past initial adoption. While some respondents discussed some lessons learned, such as being specific in prompts or only asking one thing per prompt, they felt that it was difficult to evaluate whether they were getting better at prompting or if the models were getting better with updates.

"Yeah, you do notice it sometimes, like when you've been doing this stuff, you feel like you're asking basically the same type of question, but then you don't get anything even close to the answer you were expecting. You'd think there's obviously room for improvement. But it's usually really hard to see that unless you totally mess it up". [Co-founder/game developer (Case G)]

"Maybe I just don't have the words to really perfectly pin down the exact problem I want the AI to solve". [System architect (Case A)]

"But, uh, I'm also not sure whether I've actually gotten better at asking questions—especially about new stuff—or if the AI model is just getting better". [Senior software developer (Case A)]

Third, multiple respondents highlighted the importance of learning to ignore bad outputs. Occasional bad outputs did not mean that all outputs were bad. Fourth, many respondents felt that they eventually developed an intuitive understanding of what a tool was (not) good at, making it more useful. However, this took effort, as they had to experiment with them. One respondent remarked the following in this regard:

"My impression is that, using this kind of tool, when it works it works beautifully, or at least gives you some like good result very quick, otherwise it's gonna be... it's not going to work". [Senior data scientist (Case D)]

In the pilot group of Company A, respondents began to adapt the way they worked to make better use of LLMs. Some began to always write comments explaining what they were about to do before writing any code, simply to inform the AI. Additionally, one respondent remarked that they had started doing test-driven development thanks to AI:

"For example I used to try, like, test-driven development before but... whenever I needed to do it I needed to read the document again and then it was boring and then I just gave up after some time... but now I can easily ask it [AI tool] and then it just, like... helps me to move forward very quick and in a persistent manner, so now I am in love with test-driven deployment. So that is something that's kind of changing the way that I work". [Senior data scientist (Case D)]

Some respondents also discussed having to get used to the idea of using LLMs as a part of the adoption process:

"It's really automated to start to google something or, or ask a teammate, so it'll take some mental adjusting to think that ah, I have this new tool, maybe I can figure out how to solve this problem or something like that". [Pilot group participant (Case A)]

At the end of the observation period in Company A, we asked the participants of the pilot group to reflect on their overall sentiments regarding GitHub Copilot. Although they acknowledged its usefulness, their concluding remarks were relatively reserved, suggesting they viewed it as a helpful tool rather than something transformative or revolutionary at this stage.

"I feel like I haven't heard people saying much anything about this [GitHub] Copilot, so it's not that kind of... in the center of their work, even, even if it's there in Visual Studio helping them all the time. But is it just playing just a small side role in daily work?" [Pilot group participant (Case A)]

"Yeah, but it is still just a tool that is not... in my opinion, is not that huge of a thing. It's a tool that you use. And it helps, in my opinion". [Pilot group participant (Case A)]

4.2.2. User benefits of GenAI in SE

The primary benefit for individual developers was the perceived time-saving, which translated into a higher productivity by being able to get more work done. This was achieved primarily through the elimination of various menial tasks. Beyond this big picture, however, we explored in more detail how the respondents felt they were benefiting from GenAI and how the different benefits manifested in practice in industrial settings.

Elimination of menial tasks was beneficial in various ways for our respondents. This was observed across a wide variety of tasks, such as the generation of large blocks of repetitive code:

"I had to map, for example, some enum values from one really long enum to another. And I knew that... oh damn, this is gonna be boring. This kind of select case... dudududududu, and there'll be twenty of them. So I could just add a TO DO there [and have the AI do it]." [Pilot group participant (Case A)]

However, some of the respondents discussed how eliminating menial tasks was not just about saving time. Some mentioned increased job satisfaction due to having to do less work they considered unfulfilling and getting to focus on the more demanding tasks that they felt were rewarding. Furthermore, one respondent reflected on the way this helped them retain flow when it was achieved (in direct response to the above quotation):

"My experience in situations like that, to add, is that it protects flow... the work flow, or feeling of flow. Like, you easily get this frustration like... UGH... I'd love to think about that next thing but I have to do this boring mapping first. So if you can remove that from there, on top of saving five minutes, it'll save that [flow] feeling". [Pilot group participant (Case A)]

This was also associated with the way many respondents felt that LLMs were a replacement for Google search and Stack Overflow for them. In the case of some tools, this also meant being able to stay in the IDE instead of having to leave it to search for information, which helped with staying focused. This could also eliminate delays if it meant not having to ask a co-worker and wait for a response while being stuck.

In relation to information retrieval, many of the respondents reflected on the usefulness of LLMs in learning new things and providing explanations of something. In a similar vein, LLMs were considered useful for re-familiarizing oneself with something, or not having to remember as much syntax:

"You can ask all sorts of things there — like, have we made any offers to Company X or Company Y — and it'll even pull up the old offers with proper references. But right now, the answers aren't really all that consistent". [Operations executive (Case A)]

"Like, with Unreal Engine, there's so much stuff in the documentation that's buried so deep you just don't feel like digging it up yourself, so it finds it way faster for you. And of course, if it's not in there, then it'll go through the whole internet—up to a certain point". [Co-founder/game developer (Case G)]

"It probably depends on your job context, like if you work on the same Jira ticket, same code, everything is familiar to you... you don't need to ask so much. But if you're in less familiar waters, you might look to the tools more. Like tell me in PowerShell, which I don't use actively every day, how to handle some parameter". [CTO (Case B)]

Multiple respondents working with bigger code bases also considered LLMs to be useful in navigating a large code base and understanding new parts of it faster. There was also discussion about this in relation to junior programmers:

"Well, I can't speak for others, but for me, when I started, I had some difficulty in making some code contributions because the code base was so huge. And it took... a lot, a lot of time to get used to it. So if I, if I had [GitHub] Copilot back then, I think I would have made contributions much earlier [...]" [Pilot group participant (Case A)]

"But then there's also those on the junior end of things who have found the, like, help... who, like, feel that it is a, a good help when they have to look up a lot of information". [CEO (Case B)]

In this fashion, LLMs were considered helpful in development contexts with large, existing code bases as well. Yet in some ways such contexts also made them less useful:

"But sometimes when these tasks are further development [of an already operational system] and it's that kind of a bigger project, then maybe its value is a bit lessened... if you have to like, make small changes to code here and there. If it's something where you start the project from scratch then there it's pretty good at generating all the boilerplate code". [Senior software developer (Case A)]

Indeed, multiple respondents thought that LLMs particularly excelled at getting started with a blank slate in an entirely new project, where lots of boilerplate code had to be written. While this ties directly with the overall notion of them being useful for eliminating menial tasks, it further illustrates the kind of real-world situations where LLMs are particularly useful.

Many of the respondents also considered the conversational nature of LLMs to be useful in and of itself. LLMs were considered useful as a second opinion, either to reaffirm your own thinking or to help you consider alternative viewpoints. LLMs could propose alternative ways to tackle programming problems, which could result in learning something new.

"It doesn't necessarily increase performance or make anything faster, but it's like an integral part of this development work and programming that you iterate through options and sometimes toss around ideas when you have to

find a solution to some problem, as this is problem-solving at its core, so in a way it's someone else to talk to". [Senior software developer (Case A)]

"For me it's like a tool for structuring my own thinking, which gives me like a starting point, so I can validate it myself". [Program manager (Case C)]

Finally, our respondents made some direct remarks about the interplay of different LLM tools. All of our case organizations except Company C were making use of multiple GenAI tools, and in some cases respondents were using tools not officially sanctioned by the company as well. This highlights that some of the benefits associated with GenAI may be a result of utilizing multiple tools in one's work.

"To be honest, yes, I use ChatGPT. I think that, you can ask more, like ChatGPT can, can sometimes be more useful than [GitHub] Copilot. You can have like a direct conversation, why this is not working". [Senior software developer (Case A)]

"I have used, side-by-side, ChatGPT and then that Bing Chat and I feel like in a way Bing works better and sometimes ChatGPT works better". [Senior software developer (Case A)]

"I'd suggest to start small, don't try to give it [GitHub Copilot] as a prompt to create an entire website at once. Instead, like, handle it with some other generative chat, like the thinking work first like what you want to do. And then when you know what to do then [GitHub] Copilot can jump in for the coding". [Senior software developer (Case A)]

4.2.3. User challenges of GenAI in SE

In Section 4.2.1, we discussed challenges faced by end-users within organizations when *adopting* LLMs. Here we outline additional challenges discussed by respondents in relation to continued use of LLMs past initial adoption.

During the observation in Company A, our respondents discussed various tool-specific sources of frustration. These included issues such as suggestions appearing and disappearing at random and GitHub Copilot writing a new function despite being asked to refactor an existing one. However, rather than focusing on such tool-specific findings, we look past individual tools to report more general challenges faced by the respondents.

Issues related to output reliability were widely discussed by the respondents. Erroneous outputs and so-called hallucinations were seen as a problem that necessitated being constantly critical of AI outputs. In programming-related tasks, these erroneous outputs included issues such as deprecated APIs, code versions, and links to online resources.

The difficulty of writing effective prompts remained after the initial adoption and learning process. Many respondents considered it difficult to gauge whether their prompt was bad or whether the tool was simply not suited for the task at hand. None of the respondents mentioned having found particularly helpful resources for prompting, as they were struggling with specific prompts for specific tasks. One respondent remarked that a part of the problem was also that they seldom got to re-use any prompt, as they rarely had to face the exact same problem twice.

4.3. Organizational adoption of GenAI in SE

In this subsection, we focus on the organizational point of view. To some extent, these intertwine with our findings related to user adoption and benefits and challenges, as LLMs were largely used as personal assistants for individual employees in our case companies.

4.3.1. Organizational benefits of GenAI in SE and reasons behind GenAI adoption

Our case organizations were unable to quantify any benefits gained from GenAI adoption, leading them to instead speculate. While the general sentiment was that GenAI was beneficial, they were struggling to quantify the impacts.

The general sentiment among the management respondents was that the tools were being positively received by employees. This was also in part evident by the widespread bottom-up interest in the tools. Respondents from Company A and B mentioned that developers were actively asking for licenses, while in Company E, GenAI tools were widely used by developers even without organizational licenses being provided. However, none of the case companies had actively collected any data related to GenAI.

Six of the seven case companies saw GenAI as simply being the current state of the art. Based on studies, word-of-mouth, and marketing of LLM service providers, GenAI was seen as near-mandatory. While the primary motivator behind adoption was the perceived productivity increase, especially the management of Company A was interested in other benefits as well, such as reducing turnover by increasing employee satisfaction. Company C had a clear, project-specific use case that motivated adoption. Two case companies also wanted to leverage GenAI use as a part of their marketing by being early adopters of a new technology.

4.3.2. Organizational challenges of GenAI in SE

By far the most common challenges faced on an organizational level were the data privacy and regulatory issues associated with GenAI. These were something nearly all our case companies struggled with in some way. First, some were concerned with potential cybersecurity issues stemming from their code being used as future training data. Second, the customers of some companies did not want their code or data being fed to GenAI. Explicit client consent was needed for GenAI tool use. In some cases, clients declined. Third, regulations limited what data could be included in prompts. This was also something that hindered the work of individual developers, past wider organizational impacts:

"I have to make an example myself, I cannot ask... I cannot just input the document into the question... into the prompt, so yeah so for that I need to be very careful. Sometimes I'm very tempted to do it because it's going to be like, very fast, easy solution". [Senior data scientist (Case D)]

During this study, enterprise licenses for LLM services started to emerge, offering better data privacy options. While these licenses alleviated some of the concerns for organizations that did purchase them, they did not help if the problem was data legislation. Moreover, some developers were also using additional LLM tools with free or personal licenses.

The emerging nature of the field or market was considered a challenge in multiple ways. First, new tools were continuously emerging. This contributed to indecisiveness about which tools to adopt. Second, existing tools were continuously updated, and while this typically resulted in perceived performance improvements, some were concerned about potential performance drops. Devoting resources towards utilizing a service provided by a third party and then having its performance degrade as a result of an update was a risk perceived by Company A. Third, another perceived risk was the continuity of LLM services in an emerging market. Together, these concerns made it harder to justify expensive license purchases. Locally hosted LLMs were seen as one way of tackling these issues going forward. However, only Company C was actively taking steps to utilize locally hosted LLMs at the time.

Additionally, measuring the impacts of GenAI was considered challenging. Four out of seven case organizations were interested in doing so, but struggled to find relevant metrics.

"Qualitatively this is easy when you just ask people and sense the overall sentiment that how is this changing the company. But the quantitative side, so that you could somehow, these investments... to measure their impact, or return-on-investment. That could be interesting". [Operations executive (Case A)]

"I can only rely on data you can find on the internet regarding that yeah... yeah, I think that's the common sentiment. [...] I think overall it's really hard to measure productivity when it comes to this". [Head of a branch office (Case E)]

This ties to challenges related to organizational bureaucracy. A problem in this regard was simply taking a while to arrange the enterprise licenses and distribution of the tool inside the organization, even when top-down adoption was the goal. In this regard, cost was also a relevant issue, with respondents from both Company A and B remarking that being able to better quantify the positive impacts of these tools would aid in justifying license purchases internally.

Finally, we observed some change resistance in the piloting group in Company A. One respondent expressed being very skeptical of LLMs at the start of the pilot group, and at the end of the piloting period they were still unconvinced of their usefulness. This ties to our other findings in that it seems that it takes conscious effort to make the most of LLMs, such as by trying to improve at prompting or actively writing comments in code to help an LLM along. Conversely, LLMs may appear less useful when not taking such steps. While change resistance towards LLMs did not appear common in our case companies, this may have been a result of these companies only offering the tools to those interested, as opposed to actively asking everyone to use them, for the time being.

4.3.3. GenAI use cases in industrial SE settings

Across all cases, LLMs were being utilized for the following tasks or categories of tasks: (1) acceptance criteria generation, (2) code generation (general), (3) code migration, (4) code refactoring, (5) code review, (6) comment and documentation generation, (7) content generation for own product, (8) creating meeting proceedings, (9) customer support, (10) cv creation, (11) data processing and analysis, (12) debugging, (13) detecting and updating deprecated code, (14) document management, (15) explaining or summarizing existing code, (16) idea generation, (17) information retrieval, (18) marketing material generation for software offerings, (19) project plan generation, (20) requirements engineering, (21) setting up error handling, (22) template generation, (23) test generation, (24) transcribing meetings, and (25) translation (natural language). This is merely a list of all the tasks LLMs were being used for. We did not ask respondents to systematically evaluate the usefulness of the tools for specific tasks.

Primarily, LLMs were being utilized as personal assistants for individual employees in our case companies. In Company C, an LLM was being used for a specific, process-oriented use case. In Companies F and G, GenAI was being leveraged directly for products, with Company F actively incorporating LLM-based features into their product, and Company G using GenAI to help produce content for their products. Company A was also exploring the use of LLMs in customer support and document management towards the end of the data collection for this study. Thus, while companies are actively exploring various use cases for GenAI, process- and product-related use cases seem to still be emerging in the industry.

Overall, our case companies were mostly exploring GenAI in programming and related tasks. Nonetheless, other tasks were also discussed by our respondents: *"And then community management also uses it, for this, like... what is this? Text and text template... template creation, yeah".* [Co-founder/game developer (Case G)] *"We use Teams AI features for transcribing, ChatGPT for making meeting summaries. We also use a tool called Virbo to create multimedia used in marketing and content for our product".* [CEO (Case F)]

For the time being, none of our case companies considered AI to have the potential to replace anyone in SE. None of our management and executive respondents discussed any prospects of fully automating tasks or processes, perceiving GenAI as a way of enhancing existing processes for now. Similarly, the respondents focusing on user experiences also widely felt that current LLMs were not suited for complex problems. If a problem was challenging for them personally, LLMs could not help. This was also seen as a prompting issue, in terms of it being difficult to explain more complex problems to the AI and provide it with all the necessary contextual information.

"In a way, once you reach a certain level in this field, you start running into problems that nobody else can really solve for you anymore—or that nobody else has ever run into—because they're so specific to your context, and so complicated, or just some weird combo of things. And at that point, you can't really rely on it anymore". [Senior software developer (Case A)]

"If you do try something more complex, then first off, just figuring out how to phrase the question already takes quite a while. And second, the end result usually ends up being kind of off, and probably not very usable". [Co-founder/game developer (Case G)]

4.4. Impact of company size, focus, and other background factors

As our seven cases comprise companies ranging from micro to large multinational, as well as multiple types of software businesses, we are able to provide some discussion on the impact of such factors. However, these are mainly intended as considerations for future research.

We observed B2B(/B2G) companies facing problems that arose from customer code and customer data. Companies A, B, C, and E all faced problems with GenAI usage as a result of limitations imposed by clients. Explicit client approval for GenAI use was required, and in some cases clients disallowed certain tools and use cases. On the other hand, Companies G and F owned all the code they worked on. These two companies were utilizing GenAI more freely and were not as concerned with what could and could not be done with it. Though Company G only provided subscriptions for two tools, use of other tools was allowed.

"That [ChatGPT] and Grok, X's AI, is what I've been alternating between lately [...] And we don't have any bans, so it's totally okay... but of course you should use common sense." [Co-founder/game developer (Case G)]

Our larger case companies (A and E) seemed to struggle with bureaucracy more as well in relation to GenAI adoption. Company A took over six months to transition from piloting GitHub Copilot with a select group of people to starting to distribute licenses to its employees. Company E was taking its time weighing its options on what licenses to purchase, letting its employees use tools at their own discretion in the meantime. However, we did not observe any drastic differences between how these tools were used or how their usefulness was perceived between companies of different sizes. Only Company G posited that they were benefiting from the tools to such an extent that they felt like they notably improved their competitiveness as a micro-sized, early-stage startup.

5. Summary of results

In this subsection, we provide a summary of our results, while directly answering the RQs outlined in the introduction.

RQ1: What challenges do software organizations face when adopting GenAI, and particularly LLMs, for SE?

We have discussed challenges faced by organizations in detail in Section 4.3.2. Additionally, when having individual employees within organizations adopt these tools, user adoption challenges are also relevant from the organization's point of view. We have outlined these user adoption challenges in detail in Sections 4.2.1 and 4.2.3. Our results related to both types of adoption challenges are summarized in Table 4.

RQ2: What are the benefits of GenAI, and particularly LLMs, in SE, for both organizations and individual users within the organizations, in industry settings?

We summarize our results related to RQ2 in Table 5. Overall, measuring the positive impact (benefits) of LLMs was something the case companies considered a current challenge. None of the case companies had been able to quantify any of the perceived benefits, instead relying on external data (e.g., surveys, research) and qualitative sentiments within the company to evaluate them. However, some organizational benefits should also follow from the benefits reported by individual users, which we have also explored in this study. For example, increased developer productivity should also increase organizational productivity.

Table 4
Challenges in GenAI Adoption from different the point of view of organizations and the end-users of the tools.

Organization	User
1. Data privacy and legislative concerns	1. Prompting is challenging to learn; prompts are difficult to evaluate
2. Constant emergence of new tools and technological progress	2. Tool-specific usability challenges
3. Black box models	3. Erroneous outputs and hallucinations
4. Measuring GenAI impact	4. Challenges with using these tools for complex programming problems
5. Organizational bureaucracy	
6. Change resistance	

Table 5
Potential benefits of GenAI in SE in industrial settings.

#	Potential benefit
1	Time-saving
2	Increased productivity
3	Elimination of menial tasks
4	Helps retain flow
5	Increased job satisfaction
6	Less need for Google and Stack Overflow
7	Eliminating delays, feeling stuck less often
8	Helps with learning new things
9	Ease of re-familiarizing oneself with tools, platforms, and languages
10	Understanding code and code bases
11	Ease of setting up code for a new project
12	Having a second opinion available on demand

We have discussed the benefits reported by individual end-users in detail in Section 4.2.2. While many of these overlap and could be categorized for brevity, we consider going into more detail one of the bigger benefits of a qualitative, explorative case study. For example, eliminating menial tasks surely saves time, and in turn increases productivity. However, understanding in more detail *why* LLMs save time and what types of menial tasks they eliminate was one of the goals of this study.

We also adopted a tool-independent point of view. Consequently, these benefits were a result of the utilization of different GenAI tools. The different GenAI tools utilized by our case companies are outlined in Section 4.1.

RQ3: How can GenAI, and particularly LLMs, be utilized in industrial SE settings, and how do they change SE in these contexts?

We have identified 25 primarily SE tasks (or categories of tasks) that LLMs were being utilized for in our case organizations. These were: (1) acceptance criteria generation, (2) code generation (general), (3) code migration, (4) code refactoring, (5) code review, (6) comment and documentation generation, (7) content generation for own product, (8) creating meeting proceedings, (9) customer support, (10) CV creation, (11) data processing and analysis, (12) debugging, (13) detecting and updating deprecated code, (14) document management, (15) explaining or summarizing existing code, (16) idea generation, (17) information retrieval, (18) marketing material generation for software offerings, (19) project plan generation, (20) requirements engineering, (21) setting up error handling, (22) template generation, (23) test generation, (24) transcribing meetings, and (25) translation (natural language).

GenAI tools are currently primarily being used as personal assistants for employees. We found few use cases where GenAI was being used as a part of organizational processes. Company C was utilizing an LLM as a part of a project-specific process, Company F was using GenAI as a part of its product, and Company G was generating some content for its products with GenAI. Thus, at present, it seems that process and product-related use cases for GenAI are still emerging in the industry, alongside any hypothetical larger changes to SE resulting from GenAI.

On the other hand, some individual respondents felt that GenAI had already changed the way they worked. Multiple respondents remarked that they had started writing comments before writing code to help an IDE-integrated LLM along. One respondent felt that LLMs had enabled them to do test-driven development by reducing the effort involved. Yet our respondents largely considered LLMs to be tools among others, with varying perceptions of their usefulness. Some felt that LLMs notably positively impacted their work while others seldom found use for them. Thus, on the level of individual users as well, any major changes to SE in industrial settings as a result of GenAI seem to still be underway.

6. Discussion

Our results, which we discuss here, are summarized in Section 5. Many of the benefits of LLMs for individual users that we have identified in this study (Table 5) validate the results of existing studies, while also providing new insights or providing further details into how they manifest in industrial settings. While comparatively fewer industrial studies exist, these benefits also align with the results of the case study of Coutinho et al. [14] and the practitioner interviews of Ulfsnes et al. [39]. Overlapping key benefits include: (a) better knowledge acquisition [14,39], (b) time-saving and increased productivity [14,39], and (c) job satisfaction [39]. Additionally, each study lists more minor benefits such as rubber ducking [14] (related to our 'having a second opinion available on demand') that are conceptualized differently. Nonetheless, we also discuss more atomic benefits not detailed in these studies, such as 'understanding code and code bases'.

Furthermore on the note of benefits, existing SE research posits that job satisfaction and productivity are "intricately connected" for software developers [45]. We (alongside Ulfsnes [39]) also found this link in relation to GenAI usage. Our respondents discussed increased job satisfaction as a result of LLMs eliminating menial tasks while letting them focus on rewarding ones. Elimination of menial tasks is also associated with productivity increases in other existing studies on GenAI [13,14].

Existing research argues that LLMs will transform SE [13], for example, by shifting developers' work from writing code to reviewing it [36]. However, none of our case organizations used LLMs in ways that notably transformed SE on an organizational level. We observed no higher levels automation found on existing taxonomies for degrees of SE task automation using AI [46] and GenAI [47]. Indeed, our findings in this regard align with Russo [38] who summarizes that "the compatibility of AI tools within existing development workflows predominantly drives their adoption [for individual users]". Our case organizations were utilizing LLMs almost exclusively as personal assistants for individual employees, who in turn were predominantly using them to support their existing workflows. While existing literature provides some practical examples of how more transformative changes could be achieved by integrating GenAI into SE processes and workflows, such use cases seem to still be emerging in the industry.

We nonetheless observed some minor changes, as some of our respondents discussed ways in which GenAI had changed the way they work individually. (a) Multiple respondents mentioned writing comments before beginning to write code just to give the AI an understanding of what they were about to do. This is reminiscent of the concept of literate programming proposed by Knuth [48] in 1984. (b) One respondent felt that GenAI had enabled them to do TDD, which had previously felt too time-consuming for them. (c) Some responses indicated changes in communication, as LLMs replaced coworkers for answering questions. While we have conceptualized this as a benefit in terms of reducing time spent waiting for help, it also signals decreased communication, as highlighted by Ulfsnes et al. [39]. Such changes highlight how GenAI may be changing how developers work aside from simply acting as a replacement for Google or Stack Overflow.

Various use cases for GenAI have already been explored in SE literature [13,15,23] and many are proposed for future research [21]. In this regard, the 25 types of use cases we have found serve to confirm the industrial relevance of some of these use cases. Our 25 use cases also supplement those 22 Simaremare & Edison [37] identified from existing studies and their own industry data with some of our use cases overlapping with theirs and some being novel.

Aside from benefits and use cases, our findings also provide insights into what kinds of challenges organizations and individual users are facing while adopting GenAI tools in industrial contexts (Table 4), helping to start tackling a gap highlighted in existing research [14,21]. Data privacy and regulatory issues were the most notable challenges, which is inconsistent with the existing practitioner survey of Gartner [1], but consistent with existing industrial studies [37,38]. The Gartner [1] survey reports "estimating and demonstrating AI value" as the biggest barrier for AI implementation. While six out of our seven case companies considered this challenging, they were all nonetheless still leveraging GenAI, convinced of its benefits. Though some existing studies provide quantitative estimates of the benefits of GenAI in SE (e.g., [19]), these studies are seldom conducted in industry settings where quantifying impacts is more challenging.

Our findings also complement the industrial survey by GitHub [3], which reports that 97% of their respondents had utilized AI tools for programming, while only 30 to 40% of companies were "actively encouraging and promoting their adoption". Our qualitative findings paint a similar picture. Individual employees seemed to often use tools outside the ones their organization officially provided, and often used them in their spare time, pointing to organizations lagging behind in GenAI adoption. This points to shadow IT [49] being a potential problem in relation to GenAI as well.

Finally, we studied GenAI adoption and usage without focusing on a specific tool. Many existing studies have focused on specific tools (e.g., GitHub Copilot [20]), justifiably so in order to better evaluate the impacts of specific tools for specific tasks. However, out in the field, GenAI tools are often used in conjunction, as our results show. The perceived benefits of GenAI may be a result of using multiple different GenAI tools in one's workflow, which is something seldom discussed in existing SE literature. In this regard, our findings complement those of a few existing studies based on practitioner data that also examined the use of GenAI without focusing on a specific tool [14,37,38], and vice versa.

6.1. Practical implications

Data privacy and legislative concerns are currently actively hindering GenAI adoption and usage, especially for organizations operating in regions with stricter regulations. Customer code and company documents and code involving personal data can all be problematic when using cloud-based GenAI services. Invest in enterprise licenses that promise better data privacy or host GenAI locally to tackle these issues.

Manage expectations and communicate goals. Employees may have underwhelming initial experiences. With how much hype is currently surrounding GenAI, this may result in inflated expectations. Similarly, some employees may be concerned about being replaced by AI.

Support employee adoption of GenAI. Provide guidelines or tips on how to use GenAI in your organization, as employees may not always be proactive in exploring these tools. Use external learning resources or produce your own (e.g., internal wiki). Developers generally want to use LLMs [3] and our results also indicate they may increase job satisfaction in developers.

LLMs can produce various benefits and are highly versatile, but good use cases are still emerging. We highlight various potential benefits of GenAI in SE and list 25 use cases our case companies used GenAI for. While GenAI is still currently largely used as a personal assistant for individual employees, process-level applications should be explored.

Emerging market and field pose challenges. New tools are constantly emerging, alongside new ways to leverage GenAI, while some existing services may be discontinued. Consider focusing on modular solutions where the underlying model can be easily changed to stay up-to-date with new developments.

7. Threats to validity

There are a number of threats to the validity of our findings, despite our attempts to mitigate them. In reporting these, we utilize the classification scheme proposed by Runeson & Höst [41] (who build on Yin's work [50]) for case studies in SE. This scheme utilizes four aspects of validity: construct validity, internal validity, external validity, and reliability. While this scheme is better suited for positivist approaches, Runeson & Höst also "prefer to operationalize this scheme for flexible design studies" [41] as well, and it is well-known in SE research. Thus, we also utilize it here.

Construct validity. This validity aspect focuses on whether our respondents would have shared our understanding of the central concepts in our research (i.e., GenAI, LLMs, prompting). We did notice that not all respondents necessarily understood right away what GenAI referred to, while LLMs were a more familiar concept. We made sure to use practical examples (*"such as ChatGPT"*), which seemed to clarify the concept. There are also more minor concepts such as 'challenges', but to gather the most data, we made sure to ask similar questions from different angles. For example, in relation to challenges, we would ask further questions using different wording such as "has there been anything that hasn't worked well?" We thus do not consider this a notable threat for this study.

However, somewhat related to construct validity, we do want to note that, with LLM functionalities increasingly being incorporated into various existing software, there may have been cases in which respondents were using other LLMs unwittingly. AI features incorporated into word processing software are some examples of more subtle AI features that may go unnoticed.

Internal validity. Given our exploratory, interpretive approach, we have studied a wide range of phenomena related to GenAI adoption and usage. For example, we have identified a number of benefits that individual LLM users have experienced, based on self-reporting. We did not measure any of these benefits in any way, and they are entirely based on the self-reported sentiments of the respondents. Thus, overall, we urge future research to build on the results of this study and study individual challenges and benefits in more detail using other research approaches.

External validity. The generalizability of findings from qualitative case studies is a long-standing debate. To improve the validity of our findings in this regard, we have studied multiple case organizations. Having a larger number of cases and respondents can mitigate external validity threats in case studies. Eisenhardt posits that "while there is no ideal number of cases, a number between 4 and 10 cases usually works well" [51].

Additionally, we raise the following points: (1) one of our cases (Company A) is a more in-depth case, the results of which data from the other six cases serves to further validate, and vice versa, and (2) our findings correlate with much existing research and some industry surveys, which supports their validity. The second point also supports our findings' relevance to "other people outside the investigated case[s]" [41].

Having only one respondent per case for Cases C to G poses a threat to validity, which we have tried to mitigate by focusing on management and executive-level respondents for Cases C, E, and F. For Case D, we intended to conduct additional interviews but were unable to secure any, resulting in only one developer interview. We still kept Company D as one case to provide additional data.

In terms of external validity, we also highlight three other limitations in our research approach. First, all of our case companies are based in Europe, making this a European perspective. While many of our case companies struggled with regulations related to AI and data, such concerns may not play as large a role for companies operating elsewhere. Second, though we have included companies from two different countries and companies of different sizes, we have ultimately still utilized convenience sampling in practice, relying on personal networks to find cases. We consider these companies to represent a set of average European companies that are not forerunners in innovation but simply software companies looking to keep up with industry trends. Finally, our respondents were more senior developers with 6+ years of experience. The role of job experience in the use and adoption of GenAI in industrial settings is something that should be explored in future SE research, but we were unable to do this with the data we had. However, none of our management respondents discussed having noticed differences in this regard within their organizations.

Reliability. We have used thematic analysis to analyze a large amount of qualitative data in a systematic fashion. This analysis was conducted by the first author, who occasionally discussed the process with the second author. Both the first and second author were also present for nearly all of the interviews and discussed the research process and the data that had been collected thus far often and frequently during the data collection period. We argue that this has served to slightly reduce the subjectivity of our findings. Past this, we have tried to tackle reliability threats by sharing both the interview instrument and the full results of the thematic analysis.

Ethical concerns. As additional ways to mitigate threats to validity, we have anonymized the respondents and case companies, only retaining some key information about them. The goal behind this was to make the respondents feel less hesitant about sharing unflattering details, either about their organization or their own usage of GenAI. Conversely, we also wanted to avoid situations where respondents might be inclined to portray themselves or their organization in a more positive light than realistic. Finally, while AI ethics is a highly topical in research (cf. [52]), none of our case organizations expressed concerns over ethical issues related to GenAI.

8. Conclusions

In this paper, we have studied the adoption of GenAI in industrial SE contexts by means of a multiple case study. We collected semi-structured interview data from seven case companies and longitudinal observation data from one of these case companies. We utilized thematic analysis to analyze all the data.

Our contributions are three-fold. First, our results highlight various challenges that organizations, and individual users within organizations, face when adopting GenAI for SE. Key challenges for adoption for organizations include data privacy and legislative concerns, the emerging and fast-moving market of GenAI tools, difficulty of measuring the positive impact of the tools, and potential change resistance. For individuals, the key challenges are related to prompting, such as understanding what is a good prompt and how to write prompts for specific tasks, as well as inaccurate or otherwise lacking AI outputs.

Second, we identify various benefits gained from using GenAI in industrial SE contexts as well. Many of these serve to validate existing research by providing further insights into the mechanisms behind different benefits, and by exploring how they manifest in industrial settings. Third, our results highlight how GenAI is currently utilized in industrial SE contexts, as well as ways in which GenAI is changing the way developers work. To this end, we have identified 25 types of tasks these tools are currently used for. Section 5 provides a more detailed summary of our results.

Based on this study, we outline the following future research suggestions. Future research should further investigate the role of job experience on the adoption and perceived usefulness of GenAI in industrial SE practice. We also urge future research to explore GenAI in software startup contexts. Based on our findings from one of our case companies as well, startups may find themselves particularly benefiting from GenAI. For example, GenAI tools may struggle with very large code bases but seem to excel at setting up new projects, possibly making them particularly useful in startup contexts. We also urge further research to validate and further investigate the relevance and impact of the various challenges and benefits we have outlined in this study.

CRediT authorship contribution statement

Kai-Kristian Kemell: Writing – review & editing, Writing – original draft, Visualization, Validation, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Matti Saarikallio:** Resources, Methodology, Investigation, Data curation, Conceptualization. **Anh Nguyen-Duc:** Writing – original draft, Visualization, Resources, Data curation, Conceptualization. **Pekka Abrahamsson:** Writing – original draft, Methodology, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Matti Saarikallio reports a relationship with “Company A” (as it is referred to in this study) that includes: employment. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was partly funded by the Jorma Ollila grant of the Nokia Foundation, Finland.

Appendix. Interview instrument

This is the interview instrument that was used to collect the interview data for this study through semi-structured interviews. Given the nature of semi-structured interviews, additional questions not included in the instrument were asked based on the responses of each respondent. The purpose of the instrument was to provide a general framework for each interview.

A. Background

- Please briefly describe your role and responsibilities in your company.
- What is your official job title?
- How many years of job experience do you have? And specifically job experience that you consider relevant for your current job.

B. (Gen)AI Usage

- What (Gen)AI tools are you currently using?
- At what stage is their implementation/use on a general level?
- How many people and who are using them?

C. Current Situation

- What tasks are they used for, meaning what do you do with them?
- In what kind of projects are they used? How big is the codebase, what language...
- Do you know if the users have liked them?
- What challenges have you faced with these tools? Risks, ethical issues, or any adjustments with laws or regulations?
- What has worked well with them?

D. Strategy/Financial

- What motivated you to adopt these tools?
- What is your long-term vision for using AI? Do you plan to expand?
- Have you measured the benefits, either financial or productivity? KPIs
- Do you feel this has somehow impacted your organizational culture already? Has this somehow affected processes/broader organization, or is it just “testing”?
- Does this have any marketing value for you?

Data availability

The authors do not have permission to share data.

References

[1] Gartner, Gartner survey finds generative AI is now the most frequently deployed AI solution in organizations, 2024, URL https://www.gartner.com/en/newsroom/press-releases/2024-05-07-gartner-survey-finds-generative-ai-is-now-the-most-frequently-deployed-ai-solution-in-organizations.

[2] Gartner, Gartner survey finds 79% of corporate strategists see AI and analytics as critical to their success over the next two years, 2023, URL https://www.gartner.com/en/newsroom/press-releases/2023-07-05-gartner-survey-finds-79-percent-of-corporate-strategists-see-ai-and-analytics-as-critical-to-their-success-over-the-next-two-years.

[3] GitHub, Survey: The AI wave continues to grow on software development teams, 2024, URL https://github.blog/news-insights/research/survey-ai-wave-grows/.

[4] Z. Kotti, R. Galanopoulou, D. Spinellis, Machine learning for software engineering: A tertiary study, ACM Comput. Surv. 55 (12) (2023) 1–39.

[5] Y. Li, D. Choi, J. Chung, N. Kushman, J. Schrittwieser, R. Leblond, T. Eccles, J. Keeling, F. Gimeno, A. Dal Lago, et al., Competition-level code generation with alphacode, Science 378 (6624) (2022) 1092–1097.

[6] A. Svyatkovskiy, S.K. Deng, S. Fu, N. Sundaresan, Intellicode compose: Code generation using transformer, in: Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, 2020, pp. 1433–1443.

[7] D. Amalfitano, S. Faralli, J.C.R. Hauck, S. Matalonga, D. Distante, Artificial intelligence applied to software testing: A tertiary study, ACM Comput. Surv. 56 (3) (2023) 1–38.

[8] V. Garousi, S. Bauer, M. Felderer, NLP-assisted software testing: A systematic mapping of the literature, Inf. Softw. Technol. 126 (2020) 106321.

[9] F. Nazir, W.H. Butt, M.W. Anwar, M.A. Khan Khattak, The applications of natural language processing (NLP) for software requirement engineering-a systematic literature review, in: Information Science and Applications 2017: ICISA 2017, Vol. 8, Springer, 2017, pp. 485–493.

[10] M.I. Azeem, F. Palomba, L. Shi, Q. Wang, Machine learning techniques for code smell detection: A systematic literature review and meta-analysis, Inf. Softw. Technol. 108 (2019) 115–138.

[11] A. Hammouri, M. Hammad, M. Alnabhan, F. Alsarayrah, Software bug prediction using machine learning approach, Int. J. Adv. Comput. Sci. Appl. 9 (2) (2018).

[12] Z. Xi, W. Chen, X. Guo, W. He, Y. Ding, B. Hong, M. Zhang, J. Wang, S. Jin, E. Zhou, R. Zheng, X. Fan, X. Wang, L. Xiong, Y. Zhou, W. Wang, C. Jiang, Y. Zou, X. Liu, Z. Yin, S. Dou, R. Weng, W. Cheng, Q. Zhang, W. Qin, Y. Zheng, X. Qiu, X. Huang, T. Gui, The Rise and Potential of Large Language Model Based Agents: a Survey, 2023, http://dx.doi.org/10.48550/arXiv.2309.07864, URL http://arxiv.org/abs/2309.07864. Issue: arXiv:2309.07864 arXiv:2309.07864 [cs].

[13] C. Ebert, P. Louridas, Generative AI for software practitioners, IEEE Softw. (2023).

[14] M. Coutinho, L. Marques, A. Santos, M. Dahia, C. França, R. de Souza Santos, The role of generative AI in software development productivity: A pilot case study, in: Proceedings of the 1st ACM International Conference on AI-Powered Software, in: AIware 2024, Association for Computing Machinery, New York, NY, USA, 2024, pp. 131–138, http://dx.doi.org/10.1145/3664646.3664773.

[15] A. Fan, B. Gokkaya, M. Harman, M. Lyubarskiy, S. Sengupta, S. Yoo, J.M. Zhang, Large language models for software engineering: Survey and open problems, in: 2023 IEEE/ACM International Conference on Software Engineering: Future of Software Engineering, ICSE-FoSE, IEEE, 2023, pp. 31–53.

[16] S. Bubeck, V. Chandrasekaran, R. Eldan, J. Gehrke, E. Horvitz, E. Kamar, P. Lee, Y.T. Lee, Y. Li, S. Lundberg, et al., Sparks of artificial general intelligence: Early experiments with gpt-4, 2023, arXiv preprint arXiv:2303.12712.

[17] M. Bernabei, S. Colabianchi, A. Falegnami, F. Costantino, Students' use of large language models in engineering education: A case study on technology acceptance, perceptions, efficacy, and detection chances, Comput. Educ.: Artif. Intell. 5 (2023) 100172, http://dx.doi.org/10.1016/j.caeai.2023.100172.

[18] X. Zhou, P. Liang, B. Zhang, Z. Li, A. Ahmad, M. Shahin, M. Waseem, Exploring the problems, their causes and solutions of AI pair programming: A study on GitHub and stack overflow, J. Syst. Softw. 219 (2025) 112204, http://dx.doi.org/10.1016/j.jss.2024.112204.

[19] T. Weber, M. Brandmaier, A. Schmidt, S. Mayer, Significant productivity gains through programming with large language models, Proc. ACM Hum.- Comput. Interact. 8 (EICS) (2024) 1–29.

[20] Z.C. Ani, Z.A. Hamid, N.N. Zhamri, The recent trends of research on GitHub copilot: A systematic review, in: International Conference on Computing and Informatics, Springer, 2023, pp. 355–366.

[21] A. Nguyen-Duc, B. Cabrero-Daniel, A. Przybylek, C. Arora, D. Khanna, T. Herda, U. Rafiq, J. Melegati, E. Guerra, K.-K. Kemell, M. Saari, Z. Zhang, H. Le, T. Quan, P. Abrahamsson, Generative artificial intelligence for software engineering – a research agenda, 2023, http://dx.doi.org/10.48550/arXiv.2310.18648, Issue: arXiv:2310.18648 arXiv:2310.18648 [cs].

[22] H. Cao, C. Tan, Z. Gao, Y. Xu, G. Chen, P.-A. Heng, S.Z. Li, A survey on generative diffusion models, IEEE Trans. Knowl. Data Eng. (2024).

[23] I. Ozkaya, Application of large language models to software engineering tasks: Opportunities, risks, and implications, IEEE Softw. 40 (3) (2023) 4–8, http://dx.doi.org/10.1109/MS.2023.3248401.

[24] A. Vaswani, Attention is all you need, Adv. Neural Inf. Process. Syst. (2017).

[25] C. Ebert, J.P. Arockiasamy, L. Hettich, M. Weyrich, Hints for generative AI software development, IEEE Softw. 41 (5) (2024) 24–33, http://dx.doi.org/10.1109/MS.2024.3410641.

[26] E. Jiang, K. Olson, E. Toh, A. Molina, A. Donsbach, M. Terry, C.J. Cai, PromptMaker: Prompt-based prototyping with large language models, in: Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems, in: CHI EA '22, Association for Computing Machinery, New York, NY, USA, 2022, http://dx.doi.org/10.1145/3491101.3503564.

[27] L. Giray, Prompt engineering with ChatGPT: a guide for academic writers, Ann. Biomed. Eng. 51 (12) (2023) 2629–2633.

[28] J. Wang, Y. Huang, C. Chen, Z. Liu, S. Wang, Q. Wang, Software testing with large language models: Survey, landscape, and vision, IEEE Trans. Softw. Eng. 50 (4) (2024) 911–936, http://dx.doi.org/10.1109/TSE.2024.3368208.

[29] Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang, W. Ye, Y. Zhang, Y. Chang, P.S. Yu, Q. Yang, X. Xie, A survey on evaluation of large language models, ACM Trans. Intell. Syst. Technol. 15 (3) (2024) http://dx.doi.org/10.1145/3641289.

[30] A. Moradi Dakhel, V. Majdinasab, A. Nikanjam, F. Khomh, M.C. Desmarais, Z.M.J. Jiang, GitHub copilot AI pair programmer: Asset or liability? J. Syst. Softw. 203 (2023) 111734, http://dx.doi.org/10.1016/j.jss.2023.111734.

[31] C.E.A. Coello, M.N. Alimam, R. Kouatly, Effectiveness of ChatGPT in coding: a comparative analysis of popular large language models, Digital 4 (1) (2024) 114–125.

[32] D. Nam, A. Macvean, V. Hellendoorn, B. Vasilescu, B. Myers, Using an LLM to help with code understanding, in: Proceedings of the IEEE/ACM 46th International Conference on Software Engineering, ICSE '24, Association for Computing Machinery, New York, NY, USA, 2024, http://dx.doi.org/10.1145/3597503.3639187.

[33] P. Menon, Exploring GitHub copilot assistance for working with classes in a programming course, Issues Inf. Syst. 24 (4) (2023).

[34] J. Prather, B.N. Reeves, P. Denny, B.A. Becker, J. Leinonen, A. Luxton-Reilly, G. Powell, J. Finnie-Ansley, 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. 31 (1) (2023) http://dx.doi.org/10.1145/3617367.

[35] S. Peng, E. Kalliamvakou, P. Cihon, M. Demirer, The impact of ai on developer productivity: Evidence from github copilot, 2023, arXiv preprint arXiv:2302.06590.

[36] C. Bird, D. Ford, T. Zimmermann, N. Forsgren, E. Kalliamvakou, T. Lowdermilk, I. Gazit, Taking flight with copilot: Early insights and opportunities of AI-powered pair-programming tools, Queue 20 (6) (2023) 35–57, http://dx.doi.org/10.1145/3582083.

[37] M. Simaremare, H. Edison, The state of generative AI adoption from software practitioners' perspective: An empirical study, in: 2024 50th Euromicro Conference on Software Engineering and Advanced Applications, SEAA, 2024, pp. 106–113, http://dx.doi.org/10.1109/SEAA64295.2024.00024.

[38] D. Russo, Navigating the complexity of generative AI adoption in software engineering, ACM Trans. Softw. Eng. Methodol. 33 (5) (2024) http://dx.doi.org/10.1145/3652154.

[39] R. Ulfsnes, N.B. Moe, V. Stray, M. Skarpen, Transforming software development with generative AI: Empirical insights on collaboration and workflow, in: A. Nguyen-Duc, P. Abrahamsson, F. Khomh (Eds.), Generative AI for Effective Software Development, Springer Nature Switzerland, Cham, 2024, pp. 219–234.

[40] M. Dolata, N. Lange, G. Schwabe, Development in times of hype: How freelancers explore generative ai? in: Proceedings of the IEEE/ACM 46th International Conference on Software Engineering, ICSE '24, Association for Computing Machinery, New York, NY, USA, 2024, http://dx.doi.org/10.1145/3597503.3639111.

[41] P. Runeson, M. Höst, Guidelines for conducting and reporting case study research in software engineering, Empir. Softw. Eng. 14 (2009) 131–164.

[42] A. Radford, J.W. Kim, T. Xu, G. Brockman, C. McLeavey, I. Sutskever, Robust speech recognition via large-scale weak supervision, 2022, arXiv preprint arXiv:2212.04356.

[43] A. Buscemi, A comparative study of code generation using chatgpt 3.5 across 10 programming languages, 2023, arXiv preprint arXiv:2308.04477.

[44] K. Jin, C.-Y. Wang, H.V. Pham, H. Hemmati, Can ChatGPT support developers? An empirical evaluation of large language models for code generation, in: 2024 IEEE/ACM 21st International Conference on Mining Software Repositories, MSR, 2024, pp. 167–171.

[45] N. Forsgren, M.-A. Storey, C. Maddila, T. Zimmermann, B. Houck, J. Butler, The SPACE of developer productivity: There's more to it than you think, Queue 19 (1) (2021) 20–48, http://dx.doi.org/10.1145/3454122.3454124.

[46] R. Feldt, F.G. de Oliveira Neto, R. Torkar, Ways of applying artificial intelligence in software engineering, in: Proceedings of the 6th International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering, RAISE '18, Association for Computing Machinery, New York, NY, USA, 2018, pp. 35–41, http://dx.doi.org/10.1145/3194104.3194109.

[47] J. Melegati, E. Guerra, DAnTE: A taxonomy for the automation degree of software engineering tasks, in: A. Nguyen-Duc, P. Abrahamsson, F. Khomh (Eds.), Generative AI for Effective Software Development, Springer Nature Switzerland, Cham, 2024, pp. 53–70.
[48] D.E. Knuth, Literate programming, Comput. J. 27 (2) (1984) 97–111.
[49] S. Haag, A. Eckhardt, Shadow it, Bus. Inf. Syst. Eng. 59 (2017) 469–473.
[50] R.K. Yin, Case Study Research: Design and Methods, Vol. 5, Sage, 2009.
[51] K.M. Eisenhardt, Building theories from case study research, Acad. Manag. Rev. 14 (4) (1989) 532–550.
[52] L. Weidinger, J. Uesato, M. Rauh, C. Griffin, P.-S. Huang, J. Mellor, A. Glaese, M. Cheng, B. Balle, A. Kasirzadeh, et al., Taxonomy of risks posed by language models, in: Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency, 2022, pp. 214–229.