Integrating Generative AI Tools into Software Development

A case study overseeing the adoption of generative AI tools in the software development industry

Master's Thesis in Computer Science and Engineering

TIM NIEMEIJER

Department of Computer Science and Engineering,
CHALMERS UNIVERSITY OF TECHNOLOGY AND UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2025
www.chalmers.se

Supervisor: Aris Alissandrakis, Computer Science and Engineering
Examiner: Miroslaw Staron

Master's Thesis 2025
Department of Computer Science and Engineering,
Software Engineering and Technology
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2025

Abstract

Generative AI (GenAI) and large language models (LLMs) have become popular in recent years for their versatility, usefulness, and ease of use. Now, these GenAI tools are being adopted for more professional use cases, such as software development. The case study presented in this thesis took place from March to August 2025 at a larger international IT company, while they tried to test and adopt Google's GenAI tool Gemini into the organization, with a special focus on the software development department. This case study examines different aspects of the integration and adoption of GenAI tools in a larger organization, as well as how developers view and trust these tools in a software development context. The results showed that both the *perceived usefulness* and *perceived ease of use* of generative AI tools for software development are high, and that the vast majority of developers perceive their productivity, development velocity, and problem-solving speed to all increase over a four-month period when using GenAI tools. As a product of this study, a conceptual model was developed based on collected data and adoption theory to show how different factors influence each other in the selection and adoption of GenAI tools. Tool selection from an organizational perspective showed that cost was of secondary importance to compliance, productivity, and integration into existing workflows.

Key words and phrases: Generative AI, Large Language Models, Technology adaptation, Software development industry, Case Study.

Contents

List of Figures

List of Tables

1
Introduction

In the following chapter, the motivation and context of the thesis will be explained. Additionally, the research objectives and research questions will be defined for clarity and understanding of what the goals of the thesis are.

1.1 Motivation and Context

In recent years, the number and popularity of AI tools have increased dramatically. Today, few people remain unfamiliar with tools such as ChatGPT (OpenAI, 2025), Microsoft's CoPilot (GitHub, 2025), or Google's Gemini (Google, 2025). These AI technologies have evolved significantly—from simple web-based chat-bots to sophisticated systems now being tested or directly integrated into various industries, including automotive (Soegoto et al., 2019), medicine (Rajpurkar et al., 2022), agriculture (Liu, 2020), and now also software development (Rajbhoj et al., 2024; Choudhuri et al., 2024a; Li et al., 2024).

The emergence of GenAI tools marks a shift not just in what software can do, but in how software is built. Tools like GitHub Copilot and Gemini are being actively embedded into software development workflows, influencing how developers write, review, and reason about code. Barke et al. (2023) observed that developers use AI coding assistants in two major ways: to accelerate routine tasks and to explore alternative implementation strategies—highlighting both productivity support and creative augmentation. Alternatively, Vaithilingam et al. (2022) and Pearce et al. (2022) found that developers' expectations of GenAI tools often diverge from their actual experiences, citing concerns such as over-reliance, diminished code comprehension, and the need for continuous oversight. These findings reflect a shift toward GenAI as a collaborator in the development process—still earning developers' trust and learning to meet their expectations.

While GenAI tools promise efficiency and creative support, they also introduce new complexities into the development process. Developers must navigate uncertainty around the correctness, explainability, and intent of AI-generated code. Moreover, the collaborative dynamic between developer and AI is not yet well understood: issues of trust, accountability, and appropriate reliance continue to surface as key factors shaping effective adoption. These challenges highlight the need for deeper investigation into how developers integrate, trust, and adapt to GenAI tools within real-world workflows.

Understanding how developers adopt and interact with AI tools also requires insight from established theoretical frameworks. The Technology Acceptance Model (TAM) proposed by Davis (1989) and recent Human-AI Interaction research from HCI literature such as Yang et al. (2020) provide essential foundations for analyzing user perceptions, acceptance, and trust toward emerging AI technologies. Additionally, issues of transparency and bias in AI decision-making highlighted by Binns et al. (2018), Eiband et al. (2019), and Jacovi et al. (2021) underscore the critical importance of trust as a pivotal factor influencing the effectiveness of human-AI collaboration.

This thesis investigates the integration of GenAI tools into the software development industry, focusing on a real-world case study of a larger software development organization adopting Google's Gemini. The study addresses the complexities of managing technological change, understanding trust in AI-assisted development, and navigating the evolving dynamics of human-AI collaboration. The observation period of the study spanned from mid-March to mid-August 2025. This research provides insights into how developer interaction with GenAI tools influences developers' perceptions, trust, creativity, autonomy, and collaboration patterns, as well as how organizations determine what the main selection criteria are and which roll-out strategies are used when embarking on their GenAI journey.

This context not only contributes to the academic understanding of GenAI adoption but also offers practical guidance for software engineering practitioners and organizations facing similar integration and adoption challenges. The following sections outline the main research objective and present the research questions that guide the study.

1.2 Problem Statement

In today's software development industry, the following problems have been identified:
Decision problem (management lens): Organizations lack an evidence-based insight into the selection and roll-out of GenAI coding tools that balance expected productivity gains with legal/IP risks, IDE/infrastructure integration, cost, and developer trust.
Practice gap (developer lens): There is no clear, empirical picture of how developers use GenAI (which tasks, when, and how), how they perceive it (tool, assistant, peer, etc.), or how this shapes productivity, autonomy, and confidence.
Context gap (organization): There is limited evidence on developer perceptions of GenAI—ease of use, usefulness, and trust—in low power-distance organizations, which is needed as a baseline to judge adoption pathways and outcomes in such contexts.

The goals of this study are to provide an integrated, evidence-based view that connects management drivers and governance with actual developer practices and perceptions, offering a theoretical approach with a practical example to guide the selection and adoption of GenAI tools in software development.

1.3 Research Questions

To try to solve the problems highlighted in Section 1.2 above, the following research question has been formulated:

How do developers perceive and integrate GenAI into their workflows, what adoption-and-trust patterns emerge, and which organizational factors drive tool selection and adoption in large software companies?

To facilitate answering this broad question, it is divided into the following subquestions:

- RQ1.1 How do developers perceive GenAI's role in their development workflow — as a tool, assistant, peer, competitor, or something else?
- RQ1.2 How do developers perceive the usefulness and reliability of GenAI tools in supporting their coding tasks?
- RQ1.3 In what ways are developers integrating GenAI into their daily workflow, and for which types of tasks is it most commonly used?
- RQ1.4 What factors influence developers' trust in and willingness to rely on GenAI-generated code?
- RQ1.5 How do developers perceive GenAI tools as affecting their productivity, autonomy, and confidence in solving coding problems?
- RQ1.6 What are the main drivers in adoption of GenAI tools in software development, from a higher-up perspective, in larger software development organizations?

2

Literature Review

This chapter will be an in-depth reflection of the preceding literature relevant to the case study or other companies or organizations looking to adopt or integrate GenAI tools for software development.

2.1 Generative AI in software development

Ever since the large GenAI tools were released (See Table 2.1), people are using and applying them to almost everything (starting a business, as personal trainers generating workout schedules and training programs, writing song lyrics, etc.). Since 2021, when OpenAI and Github released *Github Copilot*, coding and software engineering practices have been included in the list of jobs where GenAI tools are now available to aid and assist. In the past three years since GitHub Copilot came out, ChatGPT, and later Google's Gemini also released new versions that could aid in coding and software development, and these three tools are among the most used for software development (Olavsrud, 2025). When these tools were released, and ever since, statements are continuously made that they improve productivity, creativity, and more (section 2.1.1). However, the tools have barely been around long enough for them to be properly integrated into the industries. Even though scientific and empirical evidence exists for the use of these tools in the software development industry, the amount of released and available research on the topic is still limited enough that major companies and corporations still want more proof before adopting them (or other underlying reasons exist).

GenAI Tool	Released
Github CoPilot	2021, October (GitHub, 2021)
ChatGPT	2022, November (Southern, 2024)
Google Gemini	2023, December (Google, 2023)

Table 2.1: The release dates for three of the biggest AI tools for developers (Olavsrud, 2025)

2.1.1 Accelerating development and reducing effort

Generative AI tools have the potential to reduce skill barriers and learning curves, thereby accelerating software development substantially (Rajbhoj et al., 2024). LLMs are especially beneficial in the early stages of software development projects, and by generating structural application code, like service layer code and user interfaces, they can provide a boost in initial application development (Rajbhoj et al., 2024; Rasnayaka et al., 2024). Furthermore, groups that have been observed to utilize GenAI tools for software development have been measured to have an acceleration in task completion of 55.8% faster than their respective control group, as well as a productivity increase of between 21%-89% (Peng et al., 2023). This is also emphasized by Rajbhoj et al. (2024) who recorded a noticeable productivity increase of roughly 70%. While speed improvements are widely reported, Barke et al. (2023) note that developers also use GenAI tools for exploring alternative implementations and debugging—activities that may not always reduce time but do reduce cognitive load and development effort. However, not all studies report clear-cut productivity gains. Vaithilingam et al. (2022) observed a gap between developers' expectations and actual experience, citing usability challenges and the need for careful validation of AI-generated code, reducing the increase in productivity.

2.1.2 Effectiveness depends on human oversight and prompt engineering

The quality, correctness, and effectiveness of the response from the GenAI tool are dependent on the quality and precision of the prompt given to the tool (Rajbhoj et al., 2024; Pinto et al., 2024). For good prompting and prompt engineering, it is crucial for the effectiveness of the validation of the generated code that there is involvement of a subject matter expert (Rajbhoj et al., 2024). In the current state, people still need to supervise GenAI and validate the outputs to ensure the tool produces a suitable solution and to limit and prevent the propagation of possible errors or unwanted functionality (Rajbhoj et al., 2024; Kalliamvakou, 2024; Vaithilingam et al., 2022). Barke et al. (2023) found that developers often rely on iterative prompting and interpretation to shape useful responses.

2.1.3 Most value as supportive collaborators

AI tools and agents, by lowering the cognitive burden on developers, allow developers to focus more on so-called *system thinking* while prompting, supervising, and validating what the tools are producing (Kalliamvakou, 2024). It has also been shown that LLMs in their current state are mainly useful as assistants or tools, and they have not yet reached a level where they could replace human developers (Coello et al., 2024). Vaithilingam et al. (2022) found that developers often went in with high expectations of autonomy from the AI but quickly learned that human input and supervision remain essential, supported by Barke et al. (2023) arguing that developers treat Copilot more like an interactive collaborator than a replacement.

2.2 Collaboration

In this subsection, different types of collaboration will be discussed. The main focus will be 1) traditional collaboration (also referred to as human-human collaboration) since this is the standard and traditional collaboration type for software development; and 2) human-AI collaboration, since this is the shift we are seeing in the industry over the past decade and more recently due to the release of GenAI for software development.

2.2.1 Traditional Collaboration

As generally stated by several, including the Oxford dictionary, collaboration is "*the act of working with another person or group of people to create or produce something*" (Dagli, 2023; Press, 2025), which in software development has traditionally been either with other people or rubber ducks[1]. Besides working together to create or produce something, this can also be extended to reaching a common goal of some sort. For software development specifically, this could be something as simple as getting help in producing a line of code, or something as complex as successfully running an application live in the cloud as a team or company. As discussed by Apicella and Silk (2019), cooperation and collaboration between homo sapiens has been around for thousands of years, and is one of the main reasons for the success and survival of the human race. Collaboration is an important quality in current human society increasing motivation, task attention, and task persistence (Carr and Walton, 2014), and helps improve the future of society by reaping all the benefits of having *cognitively diverse collaboration* (Ultimo, 2024).

With the rapid development of computer hardware, computing, and AI on the rise, we are starting to see trends of asking and collaborating with AI agents instead of colleagues. This is what is paving the way for *Human-AI collaboration*.

2.2.2 Human-AI Collaboration

Human-AI collaboration is trending now with all the development and progress of GenAI tools such as the previously mentioned ChatGPT, Gemini, and Copilot. That being said, the ideas and theories behind human-AI collaboration date back to at least the 1960s. J.C.R. Licklider was one of the first people to discuss the possible symbiosis between man and machine in his article *Man-Computer Symbiosis* (Licklider, 1960). Already in 1960 he spoke about the future development of cooperation and collaboration of people and computers, and reflected that *"The basic dissimilarity between human languages and computer languages may be the most serious obstacle to true symbiosis."*. Generally speaking, there have been discussions at IT forums, conferences, and by experts whether or not AI would take over the job of software development. More recent discussions and research however points towards collaboration between people (in this case software developers) and

[1] *Rubber Ducking / Rubber duck debugging: a technique where a software developer explains their code, line by line, to an object (most commonly a rubber duck)*

GenAI (Licklider, 1960; Wang et al., 2020). In terms of guidelines for Human-AI interaction, and the design of AI from that perspective, Amershi et al. (2019) did a study which resulted in a thorough literature review on how to design AI from a human-computer interaction perspective, as well as 18 guidelines for the design of how AI should interact with people and vice versa.

2.3 Trust and Interaction

Generally speaking, there is a different level of trust between people and between people and computers or AI. In terms of trust in AI system quality, output quality, and functional value greatly influence developers' trust in these tools (Choudhuri et al., 2024b). However, GenAI for software development hasn't been around for long enough for it to have gained the trust of developers. This is why developers feel like they have to supervise and guide the tools (Kalliamvakou, 2024), and that the effectiveness and quality produced by these GenAI tools are still too dependent on the precision and appropriateness of the prompt formulation (Pinto et al., 2024; Choudhuri et al., 2024b). In terms of the trust in these tools, when considering that rumors about AI taking software developers' jobs in the future were (and are still) circling in the industry (Horowitch, 2025; Times, 2025; Abril and O'Donovan, 2025), most research seems to point towards AI becoming more of a collaborator or assistant (Kalliamvakou, 2024; Coello et al., 2024; Licklider, 1960; Pinto et al., 2024). While Licklider (1960) assumed back in the day that people would remain in full control with AI supporting, Wang et al. (2020) mention the possibility of AI assuming a more dynamic and participatory role. Furthermore, Eiband et al. (2019) emphasize that transparency in AI-generated decisions is key to building trust, further reinforcing the role of human judgment in validating GenAI outputs.

2.4 Managing Change

When talking about and discussing change in an organization or in the way of working in an organization, there are two terms that can be discussed which are similar yet different: *change management* and *management of change*. Change management focuses on the issues and difficulties involved in altering working practices, addressing challenges such as preparing for change, implementing it, and managing resistance. In contrast, management of change refers to a more systematic approach for evaluating change, including activities like risk assessment.

For change management and implementing change in organizations, one of the first big hits was the article by Kotter and Schlesinger (1979) about choosing strategies for change, and a 6 step model was defined on how to deal with resistance to change. Bridges (1991) came out with a book on managing transitions which defined 3 stages of transitioning, namely 1) ending, losing, and letting go, 2) the neutral zone, and 3) the new beginning. These three phases or steps became the basis for organizational change and leadership development. Later on, Kotter reiterated the models previously made, combining his 6-step model with for example Bridges' 3 phases, and came up with an 8 step change model in 1996. The 8-step model has since been a world standard which has been further updated over the years (Kotter, 2012). His approach combines planning, implementing, highlighting internally, and anchoring changes through the 8 steps defined in his works. All three of these pieces of literature are still relevant in today's industries and form a large part of the base for the subject of change management, which is why they are recommended by organizations such as MIT Human Resources (2025).

Alternatively to change management, management of change is a slightly safer approach and is used mainly prior to implementation to make sure that no new hazards are introduced, and to try to ensure safety and high-risk mitigation (Faulk and da Fonseca, 2022; Mullins, 2024).

2.5 Organizational Adoption

At the organizational level, the Technology–Organization–Environment (TOE) framework explains adoption through three areas: the technology itself, the organization, and the external environment (Oliveira and Martins, 2011). Prior work also shows that top-management support, business alignment, readiness (skills, budget, infrastructure), governance/risk (e.g., legal/intellectual property (IP)), and external pressures (vendors, competition, regulation) strongly affect whether interest turns into actual use (Ali et al., 2022). Culture can further shape the path and speed of adoption: Lee et al. (2013) shows that in low-power-distance, more individualist, and lower-uncertainty-avoidance settings, adoption relies more on early autonomous innovation than on imitation, favoring pilot champions and decentralized experimentation.

3
Methodology

This study employs a mixed-methods approach, combining case-study methodology guided by the framework presented by Runeson and Höst (2009) with surveys and interviews to conduct an empirical investigation. With a single-case count and literal replication logic, this study aims to provide insight into this real-world case of an organization adopting a GenAI tool for software development. The case-study design is further strengthened by utilizing methods and theories by Robson and McCartan (2016), Kotter (2012), and Yin (2018).

Kotter (2012), argues that a true mixed-method design both increases overall understanding and mitigates the inherent weaknesses of single-method studies—quantitative data offers breadth, while qualitative data provides the necessary depth into context and meaning. Runeson and Höst (2009) further recommend employing multiple data-collection procedures to enhance construct validity[1] and provide complementary perspectives on software-engineering phenomena. In addition, Yin (2018), shows that triangulating multiple sources of evidence deepens insight into complex, contextualized events and strengthens internal validity[2] through converging lines of inquiry. Therefore, this mixed-method approach uses quantitative measures (surveys) for a broad view of adoption patterns, and qualitative techniques (interviews, observations) for contextual insight, enabling a triangulated, empirically robust investigation of a real-world GenAI test rollout (Robson and McCartan, 2016; Runeson and Höst, 2009; Yin, 2018).

Yin (2018) further reflects upon the reliability: the ability for another researcher to reproduce your procedures and findings, and external validity: can the findings be transferred or generalized to other settings or populations. Regarding that, Yin distinguishes between single-case and multiple-case designs: the former provides detailed, in-depth insights into a particular phenomenon, while the latter enables analytic generalization through cross-case comparison. Replication logic for case studies can accordingly be parted into two categories; *Literal* and *Theoretical Replication*. Literal replication is when replicating the case study would be expected to produce similar results and demonstrates the reliability of findings by repeating the experiment under the same or similar conditions. Theoretical replication is when additional cases are selected specifically to produce different results to show how dif-

[1] construct validity: "measurement accuracy" - concerns the fit between theory and measurement

[2] internal validity: "causal inference" — whether observed changes can be attributed to the changes rather than surrounding factors

ferent contexts and variations lead to different outcomes. By combining case-study count and replication logic, researchers can confirm core propositions and explore boundary conditions, striking a balance between depth and breadth in empirical studies.

This study draws on two complementary models. The Technology Acceptance Model (TAM) by Davis (1989), which explains how *perceived usefulness* and *perceived ease of use* drive adoption intentions. Survey items and interview probes are mapped to these constructs, ensuring alignment with a well-validated theory of technology uptake. Jacovi et al. (2021) offers a human–AI trust framework with a multidimensional view of trust in AI systems—encompassing predictability, reliability, and transparency. Incorporation of these dimensions into the instruments guides both the phrasing of questions and the interpretation of developers' responses through an AI-specific perception and trust lens .

3.1 Context and rationale

This study focuses on a single-case study of a larger Sweden-based technology company—referred to as "the company"—which is leading within its industry. The company employs over 3,000 people, including more than 400 in-house software developers. It builds and maintains large-scale, high-availability applications used by millions of end-users daily, both for its own brands and through partnerships with other businesses. Given the scale and complexity of its systems, any technological or process change within the development organization is likely to have a significant impact.
The case was selected based on a critical-case rationale from Yin (2018), with the organization representing a strategically important example of GenAI adoption in the software industry. At the time of the study, the company had just begun integrating GenAI tools in its software development processes, offering a unique opportunity to observe the early stages of organizational adoption. This timing aligns with a time in the industry where many companies face the strategic decision of whether to adopt GenAI technologies or risk being outpaced by competitors. The case is thus particularly well-suited to generate insights relevant to both practitioners and researchers.
Several characteristics make the organization a compelling subject for this research. Its size and maturity imply that the selection and integration of GenAI must be supported by comprehensive preparations—not only technical, but also legal, ethical, and procedural. This distinguishes the case from early-stage startups or informal trials, positioning it as a high-validity example of enterprise-level adoption.
The development teams within the company vary in size and specialization, with a mix of backend-focused groups (primarily using Java), frontend teams (working mainly with React), full-stack developers, teams more infrastructure-focused, and more. As part of the GenAI rollout, a browser-based AI assistant was made available to all developers, enabling widespread, low-barrier experimentation. In parallel, more deeply integrated GenAI tools — embedded directly in integrated development environments (IDEs) — were deployed selectively to a testing group of 40 developers

tasked with piloting and evaluating these capabilities.
This study adopts Yin's longitudinal single-case design, investigating how developers in a large, mature software organization perceive and engage with generative AI tools over time. The case is both critical—as the organization represents a strategically important and high-validity example of enterprise-level AI adoption — and revelatory, providing timely access to early integration stages. Developer attitudes were captured through two surveys: an initial baseline survey conducted during the early rollout phase and a follow-up survey after ∼4 months of use. The first survey focused mostly on the developers' views on GenAI for coding, and the second one-in addition to views and engagement-was guided by the TAM, focusing on perceived usefulness and ease of use. This design aligns with Yin's guidance on longitudinal case studies, as it enables a clearer view of how developer perceptions change over time during the adoption process. Studying these changes across multiple time points also helps strengthen the internal validity of the findings.
This study used a mixed-methods approach combining quantitative surveys and qualitative interviews to capture both broad trends and in-depth insights into the company's adoption of GenAI tools in software development.

3.2 Literature selection

A mixed approach was used for finding relevant literature. The traditional approach of querying established academic databases with keywords, such as IEEE Xplore, arXiv, ACM Digital Library, and more, was used. Titles and abstracts are then scanned to assess relevance, then selected articles were skimmed to evaluate their methodology, use of theory, and alignment with the thesis topic. Literature that is frequently cited or published in reputable journals is prioritized due to credibility. Reference lists from key papers were investigated to uncover additional sources that may not have appeared in initial database searches - also known as snowballing.

In addition to traditional searching using keywords in conventional databases, generative AI tools like ChatGPT and Gemini were used as supplementary resource finders to broaden the search and identify additional relevant literature.
When using AI to search for literature, a focused and iterative prompting strategy was used to enhance the relevance of literature suggestions. The search process typically began with broad thematic queries like *"academic papers on AI in software engineering"* or *"literature on generative AI for coding in the industry"* to get suggestions of articles, books, and related sources. Every prompt for literature would generally produce between 5-10 suggestions, of which on average 1-3 of them would appear relevant. Seemingly relevant literature suggestions would then be further investigated with more targeted prompts. Things such as summaries, key findings, alignments with other papers, or citations would be queried for selected literature. If a piece of literature after this iterative prompting still seemed relevant, it would be looked up in the respective database (ieeexplore, arxiv, etc.) where it would be skimmed to determine whether or not the information provided by AI was accurate, and if it suited this study. Known researchers or article titles were occasionally referenced to guide the AI more effectively, an example being *"please find 5 pieces*

of academic literature similar to Jacovi et al. (2021)". To ensure academic credibility, all AI-suggested sources that appeared relevant (and reflected what AI had summarized about them) were manually verified in their respective databases for authenticity, publication status, and where they were published. Priority was given to peer-reviewed articles and works published by recognized academic publishers or conferences. Sources with high citation counts or referenced in other scientific publications were favored to strengthen the reliability of the literature base.

With this mixed approach of prompting AI and searching for keywords in databases, more than 200 pieces of literature were suggested/found. Most of the AI suggestions were immediately filtered out based solely on irrelevance for this study as, for example, lots of studies came up on how to develop AI.

3.3 Surveys

Surveys are this study's main source of quantitative data. Yin (2018) identifies questionnaires as a primary evidence source in case studies and recommends using them alongside interviews for triangulation. Two rounds of surveys were conducted with all approximately 200 developers invited to answer via relevant company-wide Slack channels. Participation was voluntary and anonymous; 50 developers responded to the first (initial) survey and 40 to the second (closing) survey. Of those in the second survey, 52.5% reported having taken the initial survey as well, but individual changes could not be tracked over time due to the fully anonymous design. The surveys were sent out to all developers, meaning respondents potentially spanned all teams (over 20) and all levels of seniority, closely matching the company's overall demographics.
Survey items were adapted from established TAM and trust-in-AI scales, pilot-tested with 3 developers, and demonstrated good reliability when the pilots were inquired about the survey. The initial survey assessed prior GenAI experience, trust in AI-generated outputs, and expectations of its role in software development. Although not explicitly labeled as TAM constructs, several items directly measured perceived usefulness and perceived ease of use.
To enable direct comparison between surveys, four questions were repeated in the second survey. Three of these were slightly rephrased to fit updated wording and format, while preserving their underlying constructs. This approach provided a limited basis for cross-sectional comparison of key attitudes.

3.4 Interviews

Interviews are this study's main source of qualitative data. Yin (2018), identifies interviews as a primary evidence source in case studies and recommends open-ended "how" questions to elicit rich, process-oriented narratives without putting the interviewee on the defensive. Runeson and Höst (2009) complement this by describing three levels of structure—unstructured, semi-structured, and fully structured—and suggest following specific patterns for the order of the questions (funnel, pyramid,

hourglass) to balance comparability with exploration.
The interviews serve to complement and triangulate the findings in combination with the surveys, offering a richer understanding of the organization's decision-making processes and the contextual factors influencing adoption. This aligns with Yin (2018) and Robson and McCartan (2016) recommendations to use interviews as a core source of qualitative evidence in case study research and as a way to enhance internal validity through a mixed-method approach.

To gain deeper insight into the company's strategic rationale and governance approach to GenAI adoption, one semi-structured interview was conducted with 2 people from middle management, responsible for evaluating and implementing AI tools across the development organization. That interview was conducted remotely, recorded with consent, and transcribed by Gemini for analysis. Only one interview was conducted with management, involving the two primary stakeholders responsible for AI tool adoption and selection. Including both decision-makers in the same session allowed them to discuss, support, and correct each other in real time. This joint format targeted the core voices in the adoption process and helped mitigate individual bias by enabling each stakeholder to validate or refine the other's responses. The interview with management covered areas such as tool evaluation criteria, organizational goals, compliance concerns, perceived gains in productivity, training practices, and anticipated future roles of AI in the software lifecycle. These topics were shaped both by exploratory aims and by the TAM, particularly focusing on perceived usefulness and ease of use—core constructs for understanding the developer-facing tool adoption (Davis, 1989). Additionally, questions related to governance, legal risk, and trust in AI-generated code were informed by the human–AI trust framework proposed by Jacovi et al. (2021), which emphasizes transparency, reliability, and predictability in AI systems.

To broaden perspectives beyond management, a senior tester and a senior full-stack/backend developer were interviewed. They were selected on the basis of having responded to at least the closing survey as well as volunteering for the interview. No incentives were offered. All interviewees received information regarding the topics of the interview beforehand and gave informed consent to recording, transcription, and anonymized quotation. Consistent with Runeson and Höst (2009) and Yin (2018), confidentiality was assured through pseudonyms and removal of identifying details and participants could withdraw at any time. For all conducted interviews, participants were offered member checks to verify factual accuracy and anonymity; interpretive judgments remained the researcher's responsibility (Robson and McCartan, 2016).
Following Runeson and Höst (2009) guidance, a semi-structured format was used in all interviews to ensure consistency across key topics while allowing flexibility to explore emergent themes. No specific question-order pattern (e.g., funnel, pyramid, hourglass) was enforced; instead, the draft interview outline and questions were sent to the to-be-interviewees in advance to read through and raise questions. They reviewed the questions for clarity and provided feedback on wording, sequence, and suggested minor adjustments to ensure correct understanding from their perspective.

Incorporating their input before the actual session helped balance structure and depth, reduce misunderstandings, and allowed the interviewees to prepare necessary information before the interview. Also, incorporating their input before the actual session helped balance structure with responsiveness, guaranteeing the capacity to capture unanticipated insights through better preparation and responsiveness.

3.4.1 Thematic Analysis

To systematically analyze the qualitative data obtained from the management interview (transcript), a step-by-step thematic analysis process, proposed by Naeem et al. (2020), was employed. This approach provides a clear pathway from raw text to conceptual model, ensuring both rigor and transparency.

Following these stages, the thematic analysis for the management interview looked as follows:

1. **Data Familiarization:** The full interview transcript was read multiple times to immerse the researchers in the data and note initial impressions.
2. **Generating Initial Codes:** Relevant segments of text — phrases or sentences reflecting distinct ideas about AI adoption — were systematically coded and a total of 48 relevant codes were defined (see end of Appendix C), an example being: “usability of it is rather important.” Codes were added until code saturation was reached.
3. **Searching for Themes:** The 48 codes were reviewed and redundant codes were merged, reducing 48 initial codes to 18 refined codes feeding into the themes. The 18 refined codes were then categorized into 9 candidate themes based on conceptual similarity, yielding preliminary themes such as *strategic alignment*, *training events*,and *perceived usefulness and ease-of-use*.
4. **Reviewing Themes:** Through an iterative process, the 9 candidate themes were reviewed against the data and the entire transcript to ensure coherence and distinctiveness. Overlapping or under-supported themes were merged or discarded. Through 3 iterations, the 9 themes became 6, and then became 4 which are highlighted in Section 4.2.1. An example of a theme that was sorted out was *training events*.
5. **Defining and Naming Themes:** Remaining themes were clearly defined and named, with concise descriptions of their core meaning and boundaries in relation to our research questions.
6. **Developing a Conceptual Model:** Finally,the finalized themes were organized into a conceptual model (Figure 4.5) illustrating how factors such as developer experience, legal constraints, productivity motives, and governance considerations interact to shape generative AI adoption decisions.

By following the step-by-step process proposed by Naeem et al. (2020), we ensured that the analysis remained tightly grounded in the data while building a coherent conceptual model that connects empirical findings to theory. Moreover, adhering to a transparent, reproducible methodology allows future researchers to replicate or extend this study, thereby validating or challenging its conclusions. Furthermore, interpretation of themes and the construction of the conceptual model draw on three firm-level perspectives: the Technology–Organization–Environment (TOE) lens to structure the context (Oliveira and Martins, 2011); the Ali et al. (2022) lens to label key organizational drivers (management support, alignment, readiness, governance/risk, external pressures); and the Lee et al. (2013) lens to consider culture as a moderator of adoption pathways. These lenses offer a clear scaffold for situating the qualitative themes and linking them to the research questions, while avoiding any claims about results at this stage. By following a step-by-step thematic analysis process, and using these lenses as a clear scaffold for situating the qualitative themes, it also allows for easy coupling of the findings to relevant literature and research questions.

4
Results

The following chapter will present and analyze the gathered data from the study. This includes the interview data, a conceptual model made based on theories highlighted in earlier chapters and data from the management interview, as well as a representation of the survey data including a comparison of the overlapping themes and questions.

4.1 Survey results

The study included two surveys: one at the release of the GenAI tool in the organization—initial survey— and one after ∼4 months of tool availability and usage—closing survey. Both surveys were shared and sent out through internal communication channels and shared among the developers throughout the company.

4.1.1 Initial survey

The initial survey to try to understand developers' experience with, trust in, and views on GenAI was sent out through internal channels in the company in April, after access to Gemini in the Integrated Development Environment (IDE) (only for a pilot group), as well as using it in the browser (entire organization) was rolled out roughly 2 weeks before.

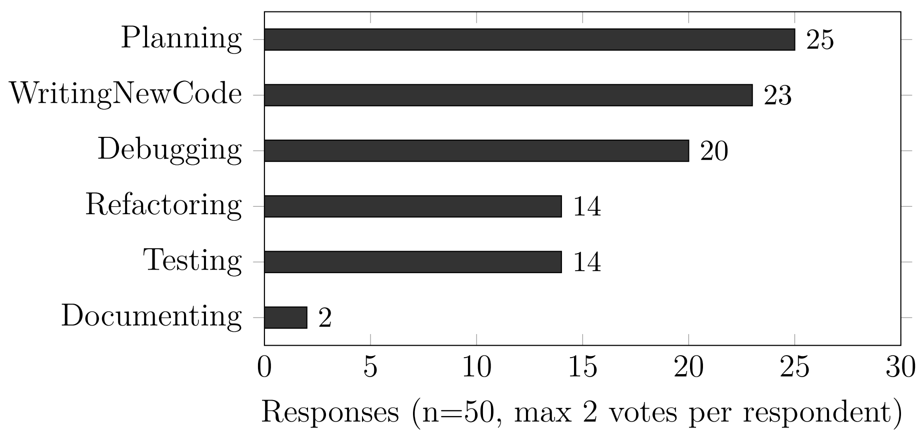

Figure 4.1: Responses for most time-consuming tasks during development, initial survey (n=50, max 2 votes per respondent).

The respondents of the initial survey consisted of 50 people. 80% of the responses

were male, 14% female, and the remaining 6% preferred not to answer. Locations of the respondents were divided between Sweden (68%), Greece(12%), India (6%), Uruguay (6%), Canada (4%), and the remaining 2% did not answer.
The respondents were asked what tasks they typically spend the most time on during development to gauge where GenAI could possibly have a big impact on time saving. Each respondent could select a maximum of two of the options. It was possible to add answers; however, none of the participants chose to do so. The results are shown in figure 4.1, and shows how developers perceive that writing new code and debugging take up second and third most time, however taking most time with 25 votes is the planning phase of the development process.
The developers were then asked about the frequency of their collaboration with their coworkers and questions about AI's impact, how much they trust it for general questions, and how much they trust the code it generates (Figure 4.2). How frequently developers collaborated with their coworkers on their tasks ranged from *never*(1) to *Multiple times for every task*(5). Based on the median being 3.5, respondents have a tendency to, more often than not, collaborate with their peers. When asked how much they trust the correctness of a coworker's help or code suggestions, ranging from not trusting it (1) to fully trusting the correctness (5), the respondents seemed to have a lot of faith in their peers' skills with a median of 4.

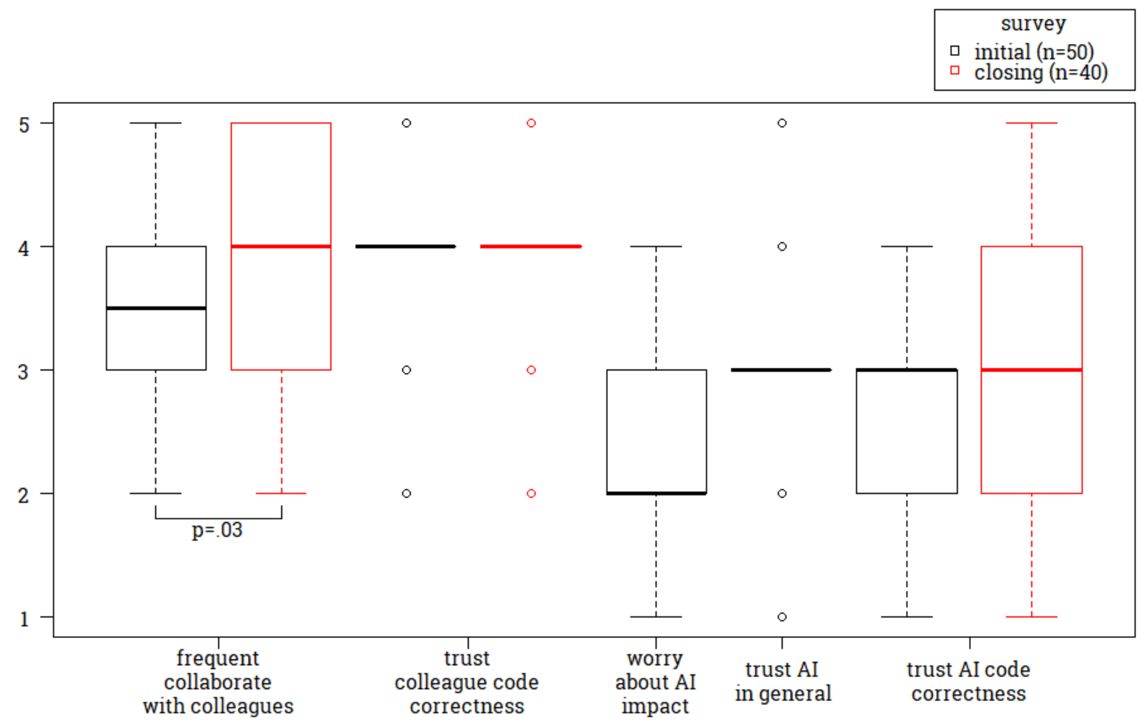

Figure 4.2: Likert scale question response distribution. Black boxes represent responses from the initial survey (Section 4.1.1), red from the closing survey (Section 4.1.2). Statistical comparisons between initial and closing survey responses' median values were done using Wicoxon rank sum tests.

Looking at the highest and lowest boxes and medians from the initial survey, devel-

opers seem to have high trust in their coworkers' suggestions and solutions (median of 4) while they do not seem to be worried about them being replaced by AI (1: not worried, 5: worry often) with a median of 2 (Figure 4.2).
Asking the developers how much they had used GenAI for coding, before the company started their pilot and adoption phase, showed that 54% of the developers use GenAI for coding and code generation on at least a monthly basis, of which 38% used it on a weekly basis or more often.

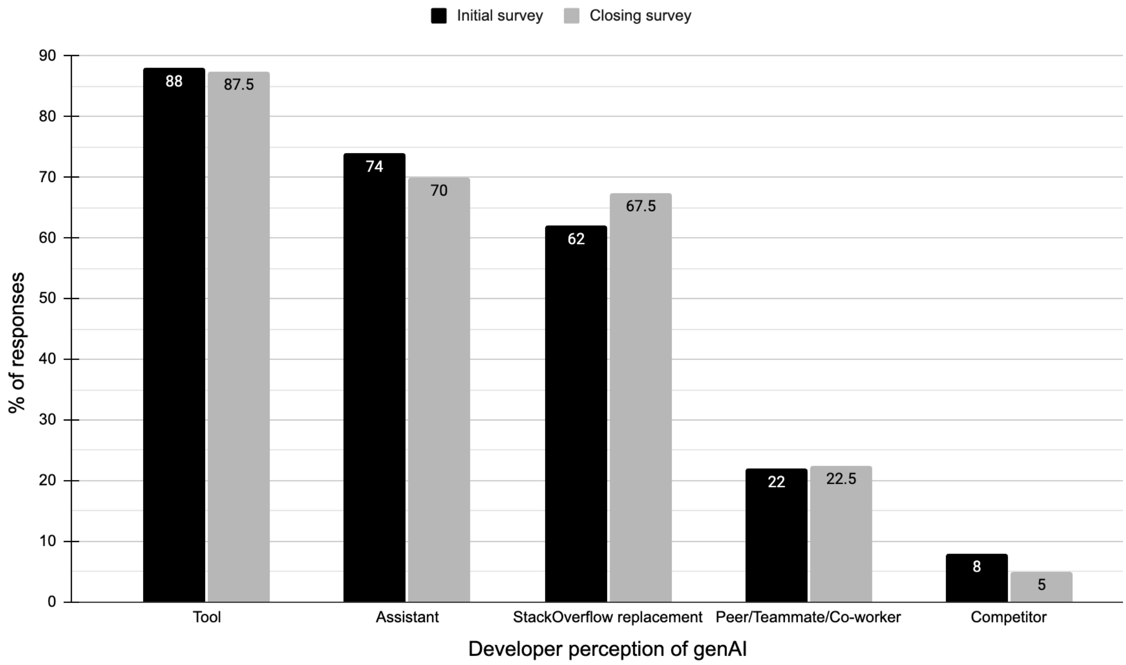

Figure 4.3: Response genAI tool perception of developers. Black = % of initial survey respondents (n=50), gray = % of closing survey respondents (n=40)

The developers were also asked how they view GenAI for software development. This was a multiple answer type of question, and developers could choose one of the existing answers or add another answer themselves. From the black bars in Figure 4.3 we can see that 88% of the developers saw GenAI in software development as a tool (44 votes) at the time of the initial survey. What is interesting is that the second highest voted option, having 74% votes (37 total), was *assistant*, giving the GenAI a more human connotation.

For replicability and transparency, the data from the initial survey can be found in Appendix A.

4.1.2 Closing survey results

The second and last survey was done ∼4 months after the approval and release of Gemini in the organization. With this survey, the intention was to have a few questions to compare changes over the time period, as well as assessing the developers' perception of GenAI and how it had effected their workflow over/after the 4-month period.

To enable meaningful comparison across the two surveys, a subset of questions from the initial survey was retained in the closing survey, with some rephrased for improved clarity and alignment with theoretical constructs and the new questions. These revisions were made to better reflect the terminology and dimensions found in relevant literature—particularly the Technology Acceptance Model (Davis, 1989) and trust in AI frameworks (Jacovi et al., 2021)—while maintaining comparability with the original phrasing. The consistent use of Likert-scale items across both surveys allows for analysis of changes in developer perceptions of usefulness, trust, ease of use, and intended future use of GenAI tools over time.

In Table 4.1, you will find the 4 questions that have been added to the second survey, 3 of them slightly rephrased for better alignment with the rest of the added questions and their phrasing. The responses for the three mapped questions in the closing survey that use the Likert-scale are represented in Figure 4.2 as the red boxes to aid visible comparison between the two surveys.

Initial survey	**Closing survey**
On a scale 1-5, how frequently would you say that you collaborate with your co-workers, ask them questions or discuss with them?	I collaborate frequently with my colleagues, multiple times per task.
On a scale 1-5, how much do/would you trust the correctness of a co-workers help or code suggestions?	I trust the correctness of code suggestions provided by my co-workers.
On a scale 1-5, how much do/would you trust the correctness of AI generated code?	I trust the correctness of code generated by generative AI tools.
How do you currently view genAI for software development? (select as many as appropriate)	How do you currently view genAI for software development? (select as many as appropriate)

Table 4.1: Mapping of questions which are present in both surveys, some of which have been rephrased.

When a GenAI tool is adopted, the logical assumption would be that people would collaborate less with coworkers due to the introduction of a tool that can replace some of that interaction from a logical perspective. However, it is observed during the study that people actually collaborated more with their coworkers (comparison tested using Wilcoxon Rank sum, p =.03). Furthermore, a small change was observed in the trust in AI-generated code; however, the increase is not of statistical significance.
All the other questions from the closing survey that used the Likert-scale are displayed in Figure 4.4. It was observed that the experience of the developers using the GenAI tools was generally positive, with the lowest first quartile (Q1) being 2.5 (*"predictability of GenAI tool behavior"* and *"confidence in using GenAI tools for production coding"*). The data shows that the *median* ≥ 3 for all the Likert-scale

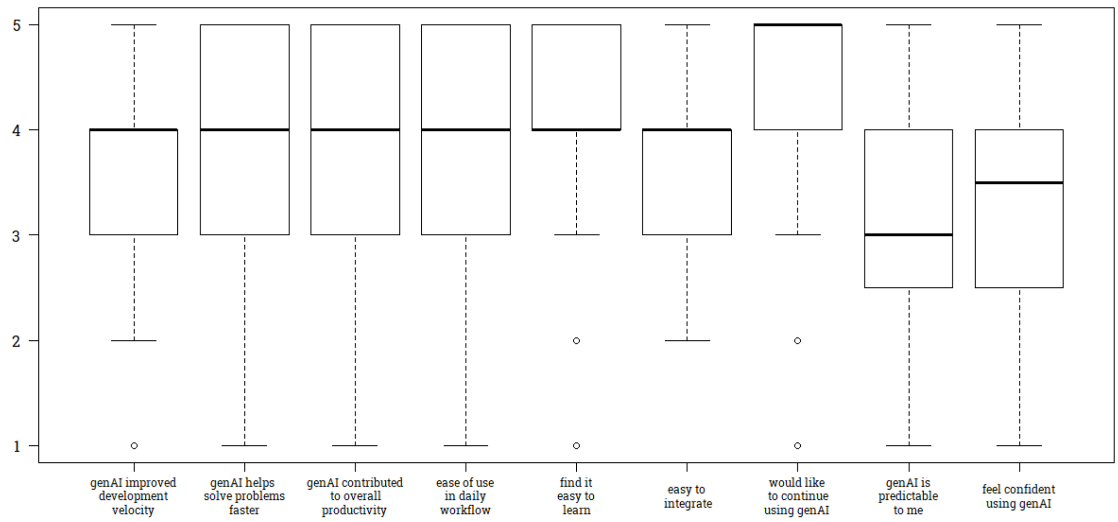

Figure 4.4: Likert scale question response distribution, closing survey (n=40).

questions in the closing survey. Given these numbers, a sub-conclusion can be made that the developers perceive the GenAI tools as useful as well as perceiving the tools as relatively easy to integrate and use in their daily workflow.
The three questions regarding improvement of development velocity, faster problem solving, and contribution to overall productivity, all having medians of 4 and upper-quartiles at or near 5, underline that the developers feel or perceive that GenAI tools speed them up and boost their overall output. The highest scoring question of the closing survey was the response to the question of whether they would like to continue to use GenAI tools going forward. The question was met with a median of five, and a Q1 of four, highlighting the enthusiasm for the use of the tools and indicating that once trialed, most developers wish to keep GenAI tools as part of their workflow. Predictability of GenAI behavior and confidence in using GenAI for code that will go into the production environment are the two only questions where neutral or slightly positive answers—and the widest spread of answers—are observed. The spread of responses to the predictability question highlights that the unpredictable outputs or solutions produced by GenAI tools are a worry or frustration for developers. The spread of responses regarding the confidence in GenAI tools' outputs that will go into production is highlighting similar concerns when it comes to the trust in these tools, as well as possible concerns regarding needing further training or governance safeguards.

4.2 Interviews results

The study included two types of interviews. One with two people from middle management and 2 interviews with developers who had utilized the GenAI tool in their workflow during the duration of the case study.

4.2.1 Interview with management

A semi-structured interview was conducted with a Systems Architect/Team Lead and a Platform Manager at the company, which lasted one hour. This provided rich insights into the organizational considerations, selection criteria, and initial experiences associated with adopting GenAI tools for software development. Both technical and managerial perspectives were explored, addressing critical aspects such as usability and developer experience, compliance and legal constraints, strategic alignment, informal adoption processes, and anticipated long-term effects on team structure and productivity. Although the company is still in an exploratory phase, without formalized Key Performance Indexes (KPIs) or structured training programs, the interview revealed patterns of consideration aligning with established theoretical frameworks, notably TAM (Davis, 1989) and Trust in AI frameworks (Jacovi et al., 2021). The following 4 points are a representation of the themes, identified and named through the thematic analysis, from the management interview. The questions for the interview, as well as the 48 initial codes found in through the thematic analysis can be found in Appendix C.

Management perspective When asked about higher management's motivation for adopting GenAI tools, the manager stated *".. overall I think management is pushing to adopt new tooling for many reasons. First and foremost is to boost productivity of the personnel."*, highlighting how perceived usefulness, in the shape of expected productivity benefits, is a key driver for technology acceptance and change. Furthermore, the manager also later in the interview stated that having a GenAI tool for coding could potentially help make the company more attractive for future employees, thereby adding another potential benefit from a management and hiring perspective.

Selection Criteria When asking early in the interview about selection and evaluation criteria, both the architect and manager pitched in. The architect commented that the general usability of the tool is important, pointing out how the perceived usefulness and perceived ease-of-use of the tools are of high importance from a code production perspective since the developer experience (DX) is central. Adding onto what the architect stated, the manager specified *"we also had to include a legal aspect, more particular the intellectual property of the company needs to be protected."*, which relates more to company-specific selection criteria in terms of legal and compliance constraints and criteria.

Adoption Strategy and Alignment When asked whether they had looked at other AI tools and why the company had selected Gemini for this first bigger test-adoption, the manager commented that they had selected Gemini as a holistic AI tool that could be enabled in other products and for other departments as well, not only in development tooling, highlighting how Gemini specifically was selected with a broader strategic alignment in mind.
Another strategic choice the company had made was not to use a formal change management approach. When asked if any specific change management models or styles had been used, the manager answered *".. the short answer would be no. it's*

more of exploratory slash collaborative approach of evaluating those tools... I don't think we have it [change management processes] at all in this company.". This shows that the company does not use a formal, top-down change management process. Instead, it relies on a collaborative and exploratory approach to adopting new tools. This reflects a culture of low power-distance (Hofstede, 2001), where decisions are made together rather than imposed by management as well as Lee et al. (2013) which stated that in low-power-distance, more individualist, and lower-uncertainty-avoidance settings, adoption relies more on early autonomous innovation than on imitation, favoring pilot champions and decentralized experimentation. Furthermore, in terms of strategy, the manager shortly after highlights that no formal training had been provided. Besides an internal hackathon[1] where it was encouraged to use AI tools the company has only prioritized informal knowledge sharing[2].

Measures and Governance Another factor of GenAI tool adoption is the impact it might have on velocity, quality, etc. When, if, and/or how we measure things like assisted development velocity and code quality, the manager answered *"we based our evaluation in collaboration and discussions based on verbal feedback from the participants on the evaluation"*, highlighting how the developers' opinions are collected in a qualitative manner. The manager states that they have no intention of making a distinction between human-written and AI-generated code. This could either be due to the level of trust in these tools, or it could be related to the difficulties in making the distinction between the two when all code is delivered in the developer's name and no trace is left of what part was AI-generated if/when the code is refactored. *"We have no intention of, let's say, introducing some kind of distinction on this code has been produced by humans, this code has been produced by AI tooling."* The architect also states *" We haven't used them [GenAI tools] long enough to actually be able to establish proper KPIs."*, highlighting the challenges of evaluating new technologies in a rapidly evolving context. Furthermore, due to the rapidly evolving context, the manager and architect acknowledged a lack of clear governance guidelines, largely due to the inherent uncertainty and rapidly evolving nature of GenAI tools. This volatility makes it difficult to establish long-term generalizations and stable measurement methods. Furthermore, as the architect highlighted, differentiating between human-written and AI-generated code is inherently difficult because all code is committed by developers who ultimately assume responsibility for its quality. Thus, current organizational practices do not distinguish between human-authored and AI-generated contributions, reinforcing the complexity in governing the use of these tools[2].

These findings highlight broader strategic and practical considerations that go beyond immediate usability and adoption metrics, aligning with concerns discussed in the literature on Trust in AI frameworks (Jacovi et al., 2021).

[1] A collaborative event where programmers and other parts of an organization can come together to work intensively on projects or ideas.

[2] The company later informed that it is planning to soon roll out formal training programs and updated policies and guidelines for the use of AI to address operational needs and regulatory development.

Following Naeem et al. (2020), the four themes were organized into a conceptual model shown in Figure 4.5 in combination with additional factors from for example TOE.

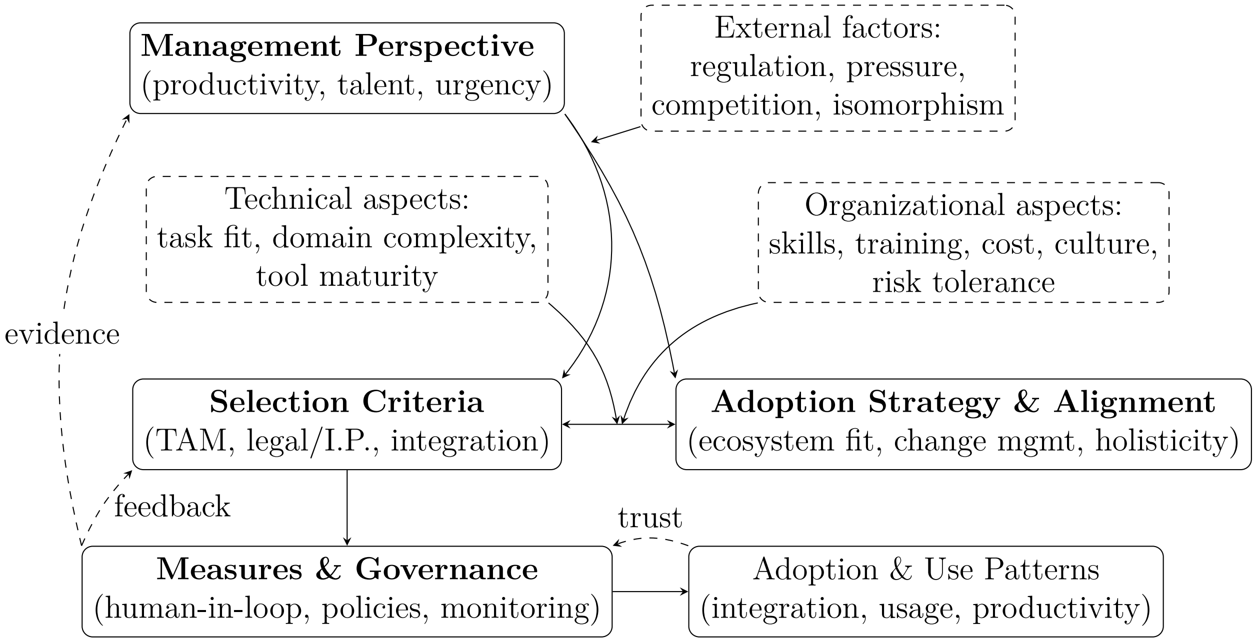

Figure 4.5: Conceptual model of organizational GenAI adoption, grounded in themes found through thematic analysis of the management interview and adoption theories from Section 2.5. Dashed boxes act as moderators: affecting the strength and direction of the relationship (arrow) between two variables. Dashed arrows signify an upstream impact creating a loop. Captions highlighted in bold are the themes from the thematic analysis of the management interview.

Management Perspective directs both *Selection Criteria* and *Adoption Strategy & Alignment* while being influenced by *External factors* such as changing industry standards or regulations. *Selection Criteria* and *Adoption Strategy & Alignment* influence each other bidirectionally: strategic choices (e.g., a holistic workspace rollout) shape acceptance criteria, while legal/IP and IDE-integration constraints steer strategy. *Selection Criteria* then flows into *Measures & Governance*, which in turn leads to *Adoption & Use Patterns* which is the final step describing individual developer adoption. *Measures & Governance* provides *feedback* to refine selection criteria and supplies *evidence* upward to inform the management perspective, creating an iterative feedback loop. The level of trust that developers have in GenAI, based on the interaction through *Adoption & Use Patterns*, influences the *Measures & Governance* due to trust being a regulator on how "in-the-loop" humans are and will be, creating a loop. Two moderators, namely *Technical aspects* and *Organizational aspects* condition the strength and direction of the relationship between *Selection Criteria* and *Adoption Strategy & Alignment*, as these aspects influence the relationship between strategy and criteria, ultimately affecting the success of the adoption. Using the model links TAM drivers with trust/predictability under governance (human-in-the-loop) (Davis, 1989; Jacovi et al., 2021), TOE and adoption (Oliveira and Martins, 2011; Lee et al., 2013) and follows case-study logic for analytical generalization (Yin, 2018; Runeson and Höst, 2009).

Summary: Overall, the interview underscores that the company's adoption of GenAI tools is primarily motivated by enhancing productivity and development experience, strongly aligning with constructs such as perceived ease-of-use and perceived usefulness. At this stage, adoption remains exploratory, informal, and reliant on qualitative feedback, with continued strategic considerations regarding governance, legal compliance, and company-wide tool accessibility and usability. Moving forward, introducing formal training programs, clear governance policies, and measurable KPIs could provide a structured pathway to sustainable integration and deeper organizational acceptance of GenAI solutions.

4.2.2 Interviews with developers

After the closing survey was completed, two 30-minute semi-structured interviews were conducted with employees to complement and enrich the survey findings by probing four theory-grounded dimensions: **First**, the developers' mental models and relational metaphors for GenAI (e.g., "assistant," "rubber duck," "tool") were explored to uncover emotional framing and a sense of agency in line with human–AI interaction frameworks (Amershi et al., 2019; Wang et al., 2020). **Second**, questions on task integration (e.g., "Which phases or tasks do you habitually use Gemini for?") were posed to elicit concrete examples of how GenAI embeds into development, maintenance, debugging, scaffolding, and bulk edits—adding contextual richness to quantitative measures of use, integration and its limitations. (Peng et al., 2023; Rasnayaka et al., 2024). **Third**, trust and willingness to rely on GenAI were examined by asking what conditions increase or decrease confidence in AI-generated code, reflecting the emphasis on predictability and transparency in trust-in-AI frameworks (Jacovi et al., 2021; Eiband et al., 2019). **Finally**, perceived productivity, autonomy, and cognitive impact (e.g., "Has GenAI changed how you solve problems or how much mental effort you invest?") were assessed to engage the TAM's core constructs—perceived usefulness and ease of use — and the cognitive-load considerations highlighted by Davis (1989) and Naeem et al. (2020). Together, these interviews provide a rich, theory-grounded qualitative layer that both triangulates and deepens the statistical trends observed in the surveys.

4.2.2.1 Tester Interview

An interview was conducted with a senior software tester who has incorporated Gemini into daily workflows for both test-case generation and broader development support. Initially treated as an experimental side tool, Gemini is now relied upon regularly to accelerate repetitive tasks—while the tester retains critical oversight and maintains full control over all AI-generated outputs.
Externalized Thinking Aid: The tester compared GenAI to a "rubber duck" early in the interview, explaining: *"I just throw things at it and see what kind of*

analysis or tips and tricks it comes up with." This metaphor echoes Pinto et al. (2024) on AI as a contextual partner that reduces cognitive friction.
Test-Case Scaffolding: When the tester prompts GenAI to create new tests, it creates several for each case. These are all analyzed and then some are discarded while others are kept. This aligns with Li et al. (2024) and Rasnayaka et al. (2024) on AI's strength in high-volume, boilerplate tasks.
Calibrated Trust: Trust in the GenAI is carefully managed. When the tester was asked about what makes him trust or distrust GenAI, he responded that it depends on how much data it already has, and that one needs to keep track when it goes off-rails. This underlined his perceived importance of reliability and predictability as prerequisites for human trust in AI systems (Jacovi et al., 2021).
Human-in-the-Loop Verification: The tester insists on full comprehension before acceptance. When he was asked whether he has ever used AI-generated code without fully understanding it he responded that *"so far, no [I haven't used code without understanding it]... I have understood the code it generated so far"*, exemplifying the continuous validation loop advocated by Amershi et al. (2019).
Productivity Gains: At the end of the interview, the tester was asked if he feels like GenAI has freed up time for more creative or challenging tasks. He responded that while GenAI *"has freed up time... I still need to think about the solution."* The tester confirms that GenAI accelerates mechanical work without relinquishing decision responsibility, supporting Peng et al. (2023) on measurable productivity improvements alongside sustained developer oversight, while hinting towards that time might have been removed from one task to then be added to another existing task.

4.2.2.2 Full-stack/Backend Developer Interview

The second interview involved a senior full-stack/backend developer who now uses Gemini in both front-end (React/Next.js) and back-end (Java/Spring) workflows. Having transitioned from pure back-end to full-stack six months ago, they moved from cautious experimentation to daily reliance—especially for refactoring and code scaffolding—while retaining critical oversight.
The developer has integrated Gemini into most of their workflows, invoking it habitually for clarification, UI enhancement, and boilerplate cleanup. They value its ability to generate "very elegant" generic solutions that drastically reduce manual effort and accelerates bug-fixing, yet they test and review every suggestion to avoid over-commenting, logic errors, or misinterpretation. Overall, they view Gemini as a strong productivity and learning aid—particularly for mastering front-end frameworks—while maintaining healthy skepticism in complex or novel scenarios and ensuring full comprehension before merging AI-generated code.
Refactoring & Boilerplate: When asked what kinds of development tasks the full-stack developer found GenAI most helpful for, he responded that it had been helpful with refactoring work, and that *"Gemini was able to find generic solutions which were very elegant and which worked well with our code."* This aligns with Li et al. (2024) and Rasnayaka et al. (2024) on AI's effectiveness at boilerplate generation and refactoring. Unlike the tester—who saw only a one-off test failure—this developer reports consistent success, suggesting outcome variability depends on prompt quality

and task/context complexity.
Collaborative Dialogue: Where the tester likened GenAI to a "rubber duck," this developer described it as a "dialogue": *"I feel I'm collaborating... it's a dialogue: I upload code, get code back, test it, and improve continuously"*, mirroring Pinto et al. (2024) on AI as a contextual partner. The "dialogue" metaphor positions GenAI as an active, iterative collaborator rather than a static tool.
Human-in-the-Loop Verification: When discussing when and how much the developer trusts the code produced by GenAI, the developer responded that he does so *"[when] I have thoroughly tested the code and I see that it works"*, exemplifying the continuous validation loop advocated by Amershi et al. (2019). Even when domain expertise is lacking, the developer relies on testing to maintain control.
Trust Through Expertise: The developers' perspective, in terms of his trust towards GenAI, was very self-reflective. Regarding specific experiences increasing or decreasing the trust in GenAI over time, he responded that *"I tested it quite a lot and there were some issues in the beginning... but then I learned better prompting... and the code quality was better as well."* This reflects Jacovi et al. (2021)'s emphasis that trust grows with both system reliability and user proficiency, as improved prompting skills yield more predictable outputs.
Productivity Gains: At the end of the interview, the developer was asked if he felt like utilizing GenAI for his coding freed up more time for more creative or challenging tasks, he replied that debugging time had significantly dropped, and that issues that could take him up to half a day to fix now sometimes only take 30 minutes. This, in turn, frees up a lot of time for more challenging tasks according to the developer, which supports Peng et al. (2023) on measurable productivity improvements, as a developer can note occasional passivity until their prompting skills evolves.

Result summary Based on the insight gained from this study, along with other statements made by various researchers and writers, the following statement made by Licklider (1960) seems to perfectly resonate with the results: *"In the anticipated symbiotic partnership, men will set the goals, formulate the hypotheses, determine the criteria, and perform the evaluations. Computing machines will do the routinizable work..."*. The human-AI collaboration patterns which were found by data collection, or interpretation of said data, underline the fact that people have not let go of the control to AI, or if they ever will. Developers today remain critical and cautious of AI-generated code, and through triangulation of surveys, interviews, and observations at the company, we see that the relationship and collaboration have not matured enough to know how this new relationship might grow or change in the future.
Trust and integration success are going to be dependent on how well AI can be integrated into existing workflows and processes, and how genAI is developed and shaped to become more trustworthy. When developers, through AI transparency, get the idea that GenAI tools grasp the context of the prompt as well as the task they have been given, a higher level of trust can be obtained. But in cases where it is unclear whether the tool has understood the assignment or not, a higher level of distrust and skepticism will be present.

5 Discussion

The following chapter will discuss and reflect upon constructs such as reliability, transparency, validity, and more. Here, potential biases and weaknesses will be highlighted, and things that should be considered if this study (or replications of it) would be conducted again in the future.

Construct Validity: Survey questions were adapted from TAM (perceived usefulness/ease of use) and trust-in-AI work, and they were pilot-tested with three developers for understandability. The interview guide followed the same process. The two survey waves were combined with three interviews and kept a clear "chain of evidence" in the methodology, results, and appendices (instruments, coding notes, quotes) for triangulation and traceability (Yin, 2018; Runeson and Höst, 2009; Robson and McCartan, 2016). Remaining risks are typical biases (self-report bias, proximity bias, confirmation bias, etc.) and limited scale validation beyond pilot feedback. This was reduced by checking survey patterns against interview themes.

Internal Validity: Causation is not claimed. Changes over four months could come from other stimuli (tool updates, team priorities, season/holidays), not only GenAI tools. Because surveys were anonymous, people could not be matched between the initial and closing survey (21 said they answered both), and three repeated questions were slightly rephrased — both of these being potential threats. To limit this, the same Likert scale was kept, each wave was analyzed on its own, and all conclusions were cross-checked with interview data. In future work, pseudonymous IDs would allow anonymous yet within-person comparisons (Yin, 2018; Runeson and Höst, 2009).

External Validity: These results are meant to be generalized *analytically* to similar settings, not statistically to everyone: in this case, a larger software company adopting GenAI under legal/compliance rules, with full-scale workspace integration (in this case, Google Workspace). The single-site design and the specific tools (Gemini, IDE assistants) limit generalization. To help readers judge transferability, the context has been described in as much detail as possible and the procedure has been documented so others can repeat or extend this study (Yin, 2018; Robson and McCartan, 2016; Runeson and Höst, 2009).

Transparency and Reliability: To strengthen reliability and transparency, this study documents a clear chain of evidence (Yin, 2018): data-collection instruments, coding protocol, analysis decisions provided throughout the thesis, linked together with quotations from the interviews, and data from the surveys (which are also provided in the appendices). The end-to-end process for data gathering, transformation, and interpretation is described in detail to allow replication. Furthermore, this thesis will be critically analyzed and exposed to independent scrutiny during the presentation and defense of the results (Runeson and Höst, 2009; Robson and McCartan, 2016). Because of confidentiality constraints, interviewees have been deidentified using generic names like *Tester* and *Architect* which have in turn been used for references and quotations. Altogether, these materials enable readers to verify how conclusions were reached and allow other researchers to reuse the protocol in comparable contexts—aiming for analytic reproducibility rather than identical outcomes.

Further thoughts and improvements: One finding was that collaboration scores were higher in the closing survey than in the initial survey. This change may partly reflect that nearly half of closing-survey respondents did not participate in the first survey. Alternatively, the timing of the closing survey - conducted during the summer holiday - may have forced developers to rely more on colleagues outside their usual networks, making them more aware of how much they depend on and collaborate with others due to the increased awareness when asking unfamiliar people for help or guidance. The slight rephrasing of the question could also have impacted this result.

Secondly, in terms of reliability and validity, the people in the company that did the survey were people that applied for or willingly decided to use GenAI tools for their development. For the IDE extension tool, one had to apply to get a license, and the browser version of Gemini was also available to people willing to keep the tab open and use it. This means that it is likely that most people that chose to use the tools, which in return would be the people answering the surveys and interviews, are people that could be considered enthusiastic about learning how to use and experiment with these tools in their workflow. This could be the source of a potential bias.

Because of the anonymous format of the surveys, paired analyses of individual respondents were not possible. The inability to link responses across surveys is a limitation, as it prevents precise measurement of attitude changes over time. If this study was to be run again in the future, surveys should consider using pseudonymous identifiers to enable paired analyses while preserving respondent confidentiality.

Ethical Considerations: Following some of the guidelines of Runeson and Höst (2009), all participants provided informed consent; the company and individuals are anonymized, the surveys were completely voluntary and anonymous, and respondents of the interviews were offered the opportunity to check the transcripts to verify factual accuracy and anonymity. Interpretation and comparison of the data remained the responsibility of the researchers.

All data (survey results, interview recordings, transcripts) were stored in Google Drive, only allowing selected people access to the documents while only giving editorial access to researchers.

Result reflections: Comparing the initial and closing surveys (Figure 4.3), the most notable changes over time concern the perceptions of the tool as a *Stack-Overflow replacement* and as a *Competitor*. Although neither change is statistically

significant (both p-values » .05), they remain noteworthy and could merit further investigation in future research.
In addition, several respondents made use of the “other, please specify” option, each of which received only a single vote. In the initial survey, these were: *“As an overly confident assistant who also makes mistakes”* and *“Search Engine replacement.”* In the closing survey, they were: *“StackOverflow supplement,” “Inspiration prompt,” “Enabler – makes scaffolding go a lot faster,”* and *“Bulk code writer.”* Because each option was selected by only one participant (2% in the initial survey, 2.5% in the closing survey), they were not included in the main analysis.
This study and its conclusions are based on research performed at an organization with a Scandinavian company culture and structure. For generalization and applicability to an organization or company with a vastly different culture, structure, or general mindset, the results of this study might not be accurate.

6
Conclusion

Circling back to the main research question in Section 1.3 of how developers perceive and integrate GenAI into their workflows, what adoption-and-trust patterns emerge, and which managerial factors drive tool selection in large software organizations, we can now make conclusions based on the results of the study highlighted in Chapter 4 and after critical reflection in Chapter 5.

RQ1.1 - GenAI's role in development workflow: Most developers use GenAI as either a *tool* (combined $\sim$ 88%), an *assistant* (combined $\sim$ 72%), or a *Stack-Overflow replacement* (combined $\sim$ 65%) in their development workflow (survey data, Figure 4.3). Based on the survey results, only a minority consider it a *co-worker* or *teammate*, underscoring that, at this stage, GenAI is still largely seen as augmentative rather than peer-like. Even in the qualitative data, GenAI surfaced as a "rubber-duck" - an externalized thinking aid that developers throw problems at to gain new perspectives. This metaphor was made by the tester and mirrors Pinto et al. (2024), who describe AI assistants serving as contextual partners that reduce cognitive friction. Unlike the tester's "rubber-duck" metaphor, the full-stack developer (section 4.2.2.2) described a two-way "dialogue," indicating a more interactive, partnership-like relationship once users gain prompting proficiency—further supporting Pinto et al. (2024) contextual partner model.

RQ1.2 – Usefulness and Reliability: From the interviews, GenAI is perceived as highly useful for offloading repetitive work like scaffolding bulk test cases or code templates, thereby boosting productivity which further supporting the claims of Peng et al. (2023), Li et al. (2024), and Rasnayaka et al. (2024). Reliability and trust are treated as dynamic constructs: outputs are checked for domain relevance and consistency, and reliability and trust are continuously assessed based on the quality and fit of the output. Every suggestion is reviewed to ensure full understanding before integration, reflecting the prerequisites of reliability, predictability, and transparency described by Jacovi et al. (2021) and Amershi et al. (2019). The full-stack developer's experience of initial failures followed by improved outputs via refined prompts illustrates how user expertise with prompting directly enhances perceived reliability — reinforcing the trust prerequisites of predictability and user competence in the framework presented by Jacovi et al. (2021).

RQ1.3 - Integration into daily workflow: Based on the interviews and survey data, GenAI has become integral to developers' routines, especially for automating high-volume, boilerplate tasks such as test-case generation, CI/CD[1] pipeline scripting, SQL[2]/reporting script assistance, and bulk search-and-replace. Many participants describe GenAI as a *StackOverflow replacement*, preferring in-context prompts for code snippets or explanations rather than navigating external forums. The developer habitually turns to GenAI when "enhancing the product" (rather than starting every task by prompting it) highlights a just-in-time integration pattern, where GenAI serves as an on-demand support rather than a default. This dynamic assistance supports Pinto et al. (2024), who characterizes AI assistants as contextual partners that step in precisely when developers need cognitive offloading or help.

RQ1.4 – Influences on trust and willingness to rely: Survey data (Figure 4.2) shows that developers consistently trust their colleagues' code suggestions more than those generated by GenAI. However, we do see an increase in the trust in correctness of AI-generated code between the initial and closing surveys (although not statistically significant). Furthermore, by the closing survey (Figure 4.4), most developers report that GenAI behavior feels sufficiently predictable and express confidence in using AI-generated code in production environments. Trust and reliance clearly vary by task and recent tool experience: routine or well-scoped tasks (e.g., test scaffolding and boilerplating) lead to greater confidence, whereas open-ended or complex tasks still instill greater caution. These findings underscore Jacovi et al. (2021) and Amershi et al. (2019) claims that *reliability* and *predictability* form the foundation of human trust in AI, and that trust must be continually calibrated as humans work with AI tools.

RQ1.5 - Productivity, autonomy and confidence: Based on findings from the closing survey highlighted in Figure 4.4, developers reported that they felt productivity, development velocity, and problem-solving speed to have all improved while using GenAI. The closing survey also showed that most developers find GenAI easy to integrate into their workflows - indicating both high *perceived usefulness* and high *perceived ease of use*, as GenAI simultaneously was easy to integrate into existing processes and accelerated routine tasks. The tester confirmed some of these effects in practice, noting that they maintained autonomy by demonstrating sustained confidence and ownership over AI-generated outputs. Furthermore, the tester also specified that GenAI offloads work without taking away developers' sense of control.

RQ1.6 – Drivers from the organizational perspective: Based on the interview with the platform manager and systems architect, four key drivers underlie their GenAI adoption strategy. First, increasing developer productivity is paramount: positioning GenAI as a strategic tool for faster delivery and reduced manual effort. Second, protecting intellectual property and ensuring legal compliance impacted their tool choice, which reflected and underlined the trust-in-AI prerequisite of clear usage terms (Jacovi et al., 2021). Third, strategic alignment with the broader Google ecosystem drove the selection of Gemini as a "holistic AI tool" usable across development as well as other departments of the organization. Finally, GenAI tooling

[1] CI/CD is short for Continuous Integration / Continuous Deployment/Delivery
[2] SQL is a scripting language for querying databases

serves as a talent-attraction and retention measure, offering modern capabilities that appeal to prospective hires. Cost considerations were secondary to compliance, productivity, and seamless integration into existing workflows.

In conclusion, in an organization with a Swedish structure and culture and after GenAI has been utilized for development for an extended period of time, the vast majority of developers view GenAI primarily as a productivity-boosting assistant — used wherever boilerplating or cognitive offloading is needed. High perceived ease of use and developer enthusiasm drove fast and smooth integration into both testing and coding workflows. In this low-power-distance, more individualist, and lower-uncertainty-avoidance setting, the adoption— which relied more on early autonomous innovation, favored pilot champions and decentralized experimentation—seemed to be successful, based on the perceptions of the developers. Trust in AI outputs is situational: clear, well-prompted interactions built confidence, while extended and iterative improvement loops that failed to meet requirements significantly decreased both confidence and trust. Across interviews, developers adhered to a strict human-in-the-loop collaboration pattern, checking and testing every AI suggestion before peer reviews and merging into main code bases. From management's perspective, cost was secondary to perceived usefulness, ease of integration, and alignment with existing workflows — underscoring that strategic fit, legal risk management, and developer perception—not price—govern organizational GenAI adoption.

7
Use of AI

For the making of this thesis, both ChatGPT and Gemini have been used for the following:

Searching for literature can be daunting and very time-consuming. During the initial part of searching for literature, once one relevant article was found, ChatGPT was used to quickly find related and similar articles. Articles were then critically skimmed and reviewed before selecting the most relevant for the purpose of this thesis.
Grammar and word choice is always a discussion when writing a scientific paper. ChatGPT was used as a tool to highlight potential issues with word choices, find synonyms that were more suitable, as well as double-check grammar. Prompting ChatGPT with questions similar to: '*for the following sentence, please highlight any grammatical issues or words that could be removed or replaced to make the sentence more clear: "..."* '. Suggestions were then taken into consideration and applied where they were deemed an improvement. At times the same would be done for shorter paragraphs, prompting for example '*Please give suggestions as to if there is a way to make the following paragraph shorter and clearer: "..."* '. Similarly, suggestions were then taken into consideration and corrected where deemed an improvement.
Human-AI collaboration has been a subject of this thesis, so it was fitting to use AI as a type of collaborative and reflective partner to try to ensure non-bias and clarity, since the thesis was written by one individual. Similar to grammar and word choices, an example prompt could be: '*for the following conclusion, please analyze and highlight any potential biases, correlations or inverse correlations i might have missed, or potential ways this could be misinterpreted: "..."*'. It would normally respond with a few suggestions which then could be considered. This allowed for additional reflection, double-checking results, and reducing biased thinking during the thesis work.

Bibliography

Abril, D. and O'Donovan, C. (2025). Bosses want you to know ai is coming for your job. *The Washington Post.* Accessed June 25, 2025. Top U.S. CEOs including Amazon, IBM, Salesforce, and JPMorgan warn that AI will significantly transform or eliminate software-developer jobs.

Ali, O., Murray, P. A., Muhammed, S., Dwivedi, Y. K., and Rashiti, S. (2022). Evaluating organizational level it innovation adoption factors among global firms. *Journal of Innovation & Knowledge*, 7(3):100213.

Amershi, S., Weld, D. S., Vorvoreanu, M., Fourney, A., Nushi, B., Collisson, P., Suh, J., Iqbal, S., Bennett, P. N., Inkpen, K., Teevan, J., Kikin-Gil, R., and Horvitz, E. (2019). Guidelines for human-ai interaction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, New York, NY, USA. Association for Computing Machinery.

Apicella, C. L. and Silk, J. B. (2019). The evolution of human cooperation. *Current Biology*, 29(11):R447–R450. Accessed: 2025-04-30.

Barke, S., James, M. B., and Polikarpova, N. (2023). Grounded copilot: How programmers interact with code-generating models. *Proceedings of the ACM on Programming Languages*, 7(OOPSLA1):Article 78, 1–27.

Binns, R., Kleek, M. V., Veale, M., Lyngs, U., Zhao, J., and Shadbolt, N. (2018). 'it's reducing a human being to a percentage': Perceptions of justice in algorithmic decisions. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*, pages 1–14. ACM. Accessed on May 16, 2025.

Bridges, W. (1991). *Managing Transitions: Making the Most of Change.* Addison-Wesley, Reading, MA.

Carr, P. B. and Walton, G. M. (2014). Cues of working together fuel intrinsic motivation. *Journal of Experimental Social Psychology*, 53:169–184. Accessed: 2025-05-05.

Choudhuri, R., Trinkenreich, B., Pandita, R., Kalliamvakou, E., Steinmacher, I., Gerosa, M., Sanchez, C., and Sarma, A. (2024a). What guides our choices? modeling developers' trust and behavioral intentions towards genai. *arXiv preprint arXiv:2409.04099.*

Choudhuri, R., Trinkenreich, B., Pandita, R., Kalliamvakou, E., Steinmacher, I., Gerosa, M., Sanchez, C., and Sarma, A. (2024b). What guides our choices? modeling developers' trust and behavioral intentions towards genai. In *Proceedings of the 47th IEEE/ACM International Conference on Software Engineering (ICSE)*, Ottawa, Canada. IEEE/ACM.

Coello, C. E. A., Alimam, M. N., and Kouatly, R. (2024). Effectiveness of chatgpt in coding: A comparative analysis of popular large language models. *Digital*, 4(1):114–125.

Dagli, K. (2023). Collaboration vs. teamwork: Key differences and when to use each. `https://www.togetherplatform.com/blog/collaboration-vs-teamwork-key-differences-and-when-to-use-each`. Accessed: 2025-04-30.

Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly*, 13(3):319–340. Accessed on May 16, 2025.

Eiband, M., Schneider, H., Bilandzic, M., Fazekas-Con, J., Haug, M., and Hussmann, H. (2019). Transparency and trust in ai: How humans perceive ai's decision-making. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, Glasgow, Scotland, UK. ACM.

Faulk, N. and da Fonseca, C. C. (2022). Moc 101—fundamentals for effective change management. *Process Safety Progress*, 41(3):492–502.

GitHub (2021). Introducing github copilot: Your ai pair programmer. Accessed: 2025-04-10.

GitHub (2025). Github copilot: Your ai pair programmer. *N/A*. Accessed: 2025-03-28.

Google (2023). Introducing gemini: Our largest and most capable ai model. Accessed: 2025-04-10.

Google (2025). Gemini. Accessed: 2025-03-28.

Hofstede, G. (2001). *Culture's Consequences: Comparing Values, Behaviors, Institutions and Organizations Across Nations*. Sage Publications, Thousand Oaks, CA, 2nd edition.

Horowitch, R. (2025). The computer-science bubble is bursting. *The Atlantic*. Accessed June 25, 2025. Discusses how AI is contributing to declines in CS enrollment and entry-level developer hiring.

Jacovi, A., Marasovic, A., Miller, T., and Goldberg, Y. (2021). Formalizing trust in artificial intelligence: Prerequisites, causes and goals of human trust in ai. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency (FAccT '21)*, pages 624–635. Accessed on May 16, 2025.

Kalliamvakou, E. (2024). A developer's second brain: Reducing complexity through partnership with ai. `https://github.blog/news-insights/research/a-developers-second-brain-reducing-complexity-through-partnership-with-ai/`. Accessed: 2025-04-30.

Kotter, J. P. (2012). *Leading Change: With a New Preface by the Author*. Harvard Business Review Press, Boston, MA.

Kotter, J. P. and Schlesinger, L. A. (1979). Choosing strategies for change. *Harvard Business Review*, 57(2):106–114.

Lee, S.-G., Trimi, S., and Kim, C. (2013). The impact of cultural differences on technology adoption. *Journal of World Business*, 48(1):20–29.

Li, Z. S., Arony, N. N., Awon, A. M., Damian, D., and Xu, B. (2024). Ai tool use and adoption in software development by individuals and organizations: a grounded theory study. *arXiv preprint arXiv:2406.17325*.

Licklider, J. C. R. (1960). Man-computer symbiosis. *IRE Transactions on Human Factors in Electronics*, HFE-1(1):4–11.

Liu, S. Y. (2020). Artificial intelligence (ai) in agriculture. *IT Professional*, 22(3):14–15. Accessed: 2025-04-10.

MIT Human Resources (2025). Recommended resources: Managing change. `https://hr.mit.edu/learning-topics/change/resources`. Accessed: 2025-04-30.

Mullins, A. (2024). What is the management of change (moc) process? `https://coastapp.com/blog/management-change-moc/`. Accessed: 2025-04-30.

Naeem, M., Besharat, F., Bashir, M., and Arif, M. (2020). A step-by-step process of thematic analysis to develop a conceptual model in qualitative research. *Quality & Quantity*, 54(3):1335–1352.

Olavsrud, T. (2025). 10 most used genai tools in the enterprise. Accessed: 2025-04-26.

Oliveira, T. and Martins, M. R. (2011). Literature review of information technology adoption models at firm level. *1566-6379*, 14.

OpenAI (2025). Chatgpt. Accessed: 2025-03-28.

Pearce, H., Ahmad, B., Tan, B., Dolan-Gavitt, B., and Karri, R. (2022). Asleep at the keyboard? assessing the security of github copilot's code contributions. *Communications of the ACM*.

Peng, S., Kalliamvakou, E., Cihon, P., and Demirer, M. (2023). The impact of ai on developer productivity: Evidence from github copilot. *arXiv preprint arXiv:2302.06590*. Accessed: 2025-04-28.

Pinto, G., de Souza, C., Rocha, T., Steinmacher, I., de Souza, A., and Monteiro, E. (2024). Developer experiences with a contextualized ai coding assistant: Usability, expectations, and outcomes. In *Proceedings of the 2nd Conference on AI Engineering (CAIN)*, Lisbon, Portugal. Association for Computing Machinery. Accessed: 2025-04-28.

Press, O. U. (2025). Collaboration. `https://www.oxfordlearnersdictionaries.com/us/definition/english/collaboration`. Accessed: 2025-04-30.

Rajbhoj, A., Somase, A., Kulkarni, P., and Kulkarni, V. (2024). Accelerating software development using generative ai: Chatgpt case study. In *Proceedings of the 17th innovations in software engineering conference*, pages 1–11.

Rajpurkar, P., Chen, E., Banerjee, O., and Topol, E. J. (2022). Ai in health and medicine. *Nature Medicine*, 28(1):31–38. Accessed: 2025-04-10.

Rasnayaka, S., Wang, G., Shariffdeen, R., and Iyer, G. N. (2024). An empirical study on usage and perceptions of llms in a software engineering project. In *Proceedings of the 1st International Workshop on Large Language Models for Code (LLM4Code)*, ICSE Workshops. Association for Computing Machinery.

Robson, C. and McCartan, K. (2016). *Real World Research: A Resource for Users of Social Research Methods in Applied Settings*. Wiley, Chichester, UK, 4th edition.

Runeson, P. and Höst, M. (2009). Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*, 14(2):131–164.

Soegoto, E. S., Utami, R. D., and Hermawan, Y. A. (2019). Influence of artificial intelligence in automotive industry. In *Journal of Physics: Conference Series*, volume 1402, page 066081. IOP Publishing. Accessed: 2025-04-10.

Southern, M. G. (2024). History of chatgpt: A timeline of major developments. Accessed: 2025-04-10.

Times, E. (2025). Amazon, Microsoft use AI to generate 25% of code in US: Will it take away jobs of software engineers in 2025? *The Economic Times*. Accessed June 25, 2025. Reports that AI is now generating roughly 25% of code at companies like Amazon and Microsoft, impacting software-engineer roles.

Ultimo, C. (2024). What is collaboration? `https://blog.webex.com/collaboration/what-is-collaboration/`. Accessed: 2025-04-30.

Vaithilingam, P., Zhang, T., and Glassman, E. (2022). Expectation vs. experience: Evaluating the usability of code generation tools powered by large language models. In *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems (CHI '22 Extended Abstracts)*, pages 1–7, New Orleans, LA, USA. ACM.

Wang, D., Churchill, E. F., Maes, P., Fan, X., Shneiderman, B., Shi, Y., and Wang, Q. (2020). From human-human collaboration to human-ai collaboration: Designing ai systems that can work together with people. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, New York, NY, USA. Association for Computing Machinery.

Yang, Q. V., Zimmerman, J., Mazumder, A. A., and Zhu, H. (2020). Re-examining whether, why, and how human-ai interaction is uniquely difficult to design. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*, pages 1–13. ACM. Accessed on May 16, 2025.

Yin, R. K. (2018). *Case Study Research and Applications: Design and Methods*. Sage Publications, Thousand Oaks, CA, 6th edition.

A

Appendix 1 - initial survey data

The following contains the CSV data for the initial survey, which can be directly transferred into a CSV file and then analyzed:

Timestamp,Score,Gender,Location,What tasks do you typically spend the most time on during development? (Select at most 2),On a scale 1-5; how frequently would you say that you collaborate with your co-workers; ask them questions or discuss with them?,On a scale 1-5; how much do/would you trust the correctness of a co-workers help or code suggestions?,How frequently; if at all; have you used genAI in your personal life; before and up until March 2025?,Overall; which genAI have you used (or tried) the most; before and up until March 2025? (Select up to 3 most used; if you haven't used any select never used'),On a scale 1–5; have you ever thought or worried about genAI greatly impacting or even replacing your role as a software developer?,On a scale 1-5; how much do you trust the answers that an AI provides you for general prompts?,How frequently have you used genAI for software development; before the company started integrating it?,On a scale 1-5; how much do/would you trust the correctness of AI generated code?,How do you currently view genAI for software development? (select as many as appropriate)
4/11/2025 13:49:17,,Female,Sweden,Writing new code; Planning / Thinking,4,5,Weekly basis,ChatGPT; Gemini,3,2,Weekly basis,2,Tool; Assistant
4/11/2025 13:50:15,,Male,Sweden,Testing; Planning / Thinking,4,3,Weekly basis, ChatGPT,2,3,Monthly basis,3,Assistant; Peer/teammate; StackOverflow replacement
4/14/2025 8:47:08,,Male,Sweden,Writing new code; Refactoring,3,4,A few times per year (tested it a few times; bi-yearly; or quarterly),ChatGPT,2,3,Never used it,3,Tool; Peer/teammate; StackOverflow replacement
4/14/2025 9:41:07,,Male,India,Writing new code; Debugging,3,4,Monthly basis, ChatGPT; Gemini; DeepSeek,1,4,Never used it,4,Tool; Assistant; StackOverflow replacement
4/14/2025 9:41:40,,Male,Greece,Planning / Thinking; Understanding the hell that is the IBE codebase,4,3,Monthly basis,ChatGPT; DeepSeek,1,2,A few times per year (tested it a few times; bi-yearly; quarterly),2,Tool; StackOverflow replacement
4/14/2025 9:41:50,,Male,Sweden,Writing new code; Refactoring,3,4,Monthly basis, ChatGPT; CoPilot,3,3,Weekly basis,3,Tool; Assistant
4/14/2025 9:42:11,,Female,Sweden,Writing new code; Debugging,5,5,Weekly basis, ChatGPT; Gemini; DeepSeek, 4,3,Weekly basis,3,Tool; StackOverflow replacement
4/14/2025 9:42:39,,Male,Sweden, Debugging; Planning / Thinking,3,4,A few times per year (tested it a few times; bi-yearly; or quarterly),Gemini; DeepSeek; Claude,1,2,

A few times per year (tested it a few times; bi-yearly; quarterly),1,Tool; StackOverflow replacement

4/14/2025 9:44:06„Male, Sweden, Writing new code; Refactoring,4,4,A few times per year (tested it a few times; bi-yearly; or quarterly),ChatGPT,2,3,A few times per year (tested it a few times; bi-yearly; quarterly),3,Tool; Assistant; StackOverflow replacement

4/14/2025 9:44:24„Male,Sweden,Writing new code; Debugging,3,3,Daily basis, ChatGPT; OpenAI; Cursor,3,5,Daily basis,4,Tool; Assistant; Peer/teammate; Competitor

4/14/2025 9:45:46„Male,Sweden,Writing new code; Debugging,5,4,Weekly basis, ChatGPT; Gemini,4,4,Never used it,3,Tool; Assistant; StackOverflow replacement

4/14/2025 9:47:11„Male,Greece,Testing; Planning / Thinking,4,4,Weekly basis, ChatGPT,2,3,Weekly basis,3,Tool; StackOverflow replacement

4/14/2025 9:47:42, , Male,Sweden,Testing; Documenting,4,4,Daily basis,ChatGPT; Gemini,2,4,Weekly basis,4,Tool; Assistant

4/14/2025 9:47:42„Male,Sweden,Debugging; Planning / Thinking,3,4,A few times per year (tested it a few times; bi-yearly; or quarterly)„1,3,Never used it,4,Assistant; StackOverflow replacement

4/14/2025 9:47:46„Male,Sweden,Testing; Planning / Thinking,3,4,Monthly basis, Gemini,1,3,A few times per year (tested it a few times; bi-yearly; quarterly),3,Tool; Assistant; StackOverflow replacement; Search Engine replacement

4/14/2025 9:47:52„Male,Sweden,Writing new code; Testing,4,4,Weekly basis,Gemini; CoPilot,3,3,Weekly basis,4,Tool; Assistant; Peer/teammate

4/14/2025 9:49:10„Male,Sweden,Debugging; Planning / Thinking,4,4,Monthly basis,Gemini,2,3,Monthly basis,3,Tool

4/14/2025 9:51:16„Male,India,Writing new code; Debugging,4,4,Weekly basis, ChatGPT; Gemini; DeepSeek,1,3,Weekly basis,2,Assistant

4/14/2025 9:52:20„Male,Sweden,Refactoring; Planning / Thinking,2,2,A few times per year (tested it a few times; bi-yearly; or quarterly),ChatGPT; Gemini; CoPilot,4,1,Never used it,1,Tool; Assistant

4/14/2025 9:52:31„Male,Greece,Refactoring; Planning / Thinking,4,4,Weekly basis,ChatGPT,3,4,Weekly basis,4,Tool; Assistant

4/14/2025 9:53:59„Female,Greece,Writing new code; Testing,3,4,Daily basis, ChatGPT; OpenAI,3,3,Never used it,3,Tool; Assistant; StackOverflow replacement

4/14/2025 9:59:39„Male,Sweden,Debugging; Planning / Thinking,4,3,Daily basis, ChatGPT; CoPilot,3,4,Weekly basis,3,Tool; Assistant; StackOverflow replacement

4/14/2025 10:00:08„Male,Sweden,Writing new code; Planning / Thinking,3,4,Weekly basis,Gemini,4,4,A few times per year (tested it a few times; bi-yearly; quarterly),4,Tool; StackOverflow replacement

4/14/2025 10:04:15„„Writing new code; Planning / Thinking,3,4,A few times per year (tested it a few times; bi-yearly; or quarterly),ChatGPT; Gemini; Le chat,2,2,Never used it,2,Tool; Assistant; StackOverflow replacement

4/14/2025 10:05:15„Male,Sweden,Testing; Planning / Thinking,3„A few times per year (tested it a few times; bi-yearly; or quarterly)„2,3,A few times per year (tested it a few times; bi-yearly; quarterly),2,Tool; StackOverflow replacement

4/14/2025 10:05:18„Male,Sweden,Refactoring; Testing,4,4,Weekly basis,ChatGPT;

CoPilot; Ollama on local machine,2,3,A few times per year (tested it a few times; bi-yearly; quarterly),3,Tool; Assistant; Peer/teammate; StackOverflow replacement
4/14/2025 10:05:38,,Male,Sweden,Debugging; Planning / Thinking,4,4,Weekly basis,ChatGPT,2,3,Monthly basis,3,Tool; StackOverflow replacement
4/14/2025 10:30:21,,Male,Sweden,Debugging; Planning / Thinking,3,3,Daily basis, ChatGPT,3,3,A few times per year (tested it a few times; bi-yearly; quarterly),2,Tool; Assistant; Peer/teammate; Competitor; StackOverflow replacement
4/14/2025 10:30:22,,Female,Sweden,Testing; Documenting,3,3,Daily basis,ChatGPT; Gemini; OpenAI,1,3,A few times per year (tested it a few times; bi-yearly; quarterly),4,Assistant; Peer/teammate; StackOverflow replacement
4/14/2025 10:44:05,,Prefer not to say,Sweden,Debugging; Testing,3,5,Weekly basis,ChatGPT; Gemini; CoPilot,1,3,Never used it,3,Tool; Assistant
4/14/2025 11:17:46,,Male,Sweden,Writing new code; Planning / Thinking,5,4,Daily basis,ChatGPT; Gemini; Claude,3,4,Daily basis,4,Tool; Assistant; Competitor; StackOverflow replacement
4/14/2025 13:51:21,,Male,Sweden,Writing new code; Refactoring,3,4,Monthly basis,ChatGPT; CoPilot,3,2,A few times per year (tested it a few times; bi-yearly; quarterly),2,Tool; Assistant
4/14/2025 14:18:42,,Female,Sweden,Debugging; Refactoring,4,4,Daily basis, ChatGPT; Gemini; CoPilot,1,4,Monthly basis,3,Tool; Assistant; StackOverflow replacement
4/14/2025 14:59:59,,Male,Sweden,Writing new code; Debugging,3,5,Daily basis, ChatGPT; CoPilot,3,3,Daily basis,4,Tool; Assistant; Peer/teammate; StackOverflow replacement
4/14/2025 19:20:28,,Female,Sweden,Debugging; Refactoring,3,3,Daily basis,CoPilot, 2,3,Weekly basis,3,Peer/teammate; StackOverflow replacement
4/15/2025 8:59:14,,Male,Sweden,Writing new code; Debugging,3,5,Daily basis, ChatGPT; Gemini; Perplexity.ai,1,3,Weekly basis,3,Tool; Assistant
4/15/2025 9:50:11,,Male,Sweden,Testing,5,5,Weekly basis,ChatGPT,3,,Monthly basis,3,Tool; Assistant; StackOverflow replacement
4/15/2025 10:17:57,,Male,Greece,Writing new code; Planning / Thinking,5,4,A few times per year (tested it a few times; bi-yearly; or quarterly),ChatGPT; Gemini; Llama,4,2,A few times per year (tested it a few times; bi-yearly; quarterly),3,Tool; Assistant
4/15/2025 16:19:26,,Male,Uruguay,Testing; Planning / Thinking,3,5,Daily basis, ChatGPT; Gemini; Claude,2,4,Daily basis,3,Tool; Assistant; StackOverflow replacement
4/15/2025 23:43:50,,Prefer not to say,Canada,Writing new code; Planning / Thinking,4,3,Weekly basis,ChatGPT; Gemini; Llama,3,2,A few times per year (tested it a few times; bi-yearly; quarterly),3,Tool; Competitor
4/16/2025 9:54:19,,Male,Greece,Writing new code; Refactoring,3,3,Weekly basis, ChatGPT; Gemini; CoPilot,2,2,Monthly basis,2,Tool; Assistant
4/16/2025 10:08:23,,Male,India,Debugging; Planning / Thinking,4,3,Weekly basis, ChatGPT; Gemini; Grok,2,3,Never used it,3,Tool; Assistant; StackOverflow replacement
4/16/2025 10:13:39,,Male,Sweden,Refactoring; Planning / Thinking,4,4,A few times per year (tested it a few times; bi-yearly; or quarterly),ChatGPT; CoPilot,2,3,A few

times per year (tested it a few times; bi-yearly; quarterly),1,Tool; Assistant
4/16/2025 11:34:47,,Male,Sweden,Writing new code; Refactoring,5,4,Daily basis, ChatGPT; Claude,3,3,Weekly basis,4,Tool; Assistant; Peer/teammate; StackOverflow replacement; As an overly confident assistant who also makes mistakes
4/16/2025 11:39:44,,Male,,Writing new code; Debugging,3,2,Weekly basis,ChatGPT, 1,2,Weekly basis,2,Assistant
4/16/2025 14:34:41,,Male,Canada,Refactoring; Planning / Thinking,3,4,Monthly basis,ChatGPT; Gemini; CoPilot,2,2,Monthly basis,1,Tool; Assistant
4/21/2025 17:52:21,,Female,Uruguay,Debugging; Testing,4,5,Monthly basis,Claude, 2,2,Monthly basis,2,Tool; StackOverflow replacement
4/22/2025 9:34:39,,Male,Sweden,Writing new code; Testing,3,4,Daily basis,ChatGPT; Gemini; Claude,4,4,Weekly basis,4,Tool; Peer/teammate; StackOverflow replacement
4/22/2025 14:10:24,,Male,Uruguay,Debugging; Planning / Thinking,5,4,Daily basis,ChatGPT; Gemini; DeepSeek,2,4,Never used it,3,Tool; Assistant
4/23/2025 22:54:20,,Male,Sweden,Refactoring; Planning / Thinking,3,4,Weekly basis,ChatGPT; Gemini; OpenAI,1,3,Weekly basis,3,Tool; Assistant; StackOverflow replacement

B

Appendix 2 - closing survey data

The following contains the CSV data for the closing survey, which can be directly transferred into a CSV file and then analyzed:

Timestamp,Score,Gender,Location,Did you answer the initial survey in March?,I collaborate frequently with my colleagues; multiple times per task.,I trust the correctness of code suggestions provided by my co-workers.,I trust the correctness of code generated by generative AI tools.,How do you currently view genAI for software development? (select as many as appropriate) ,Using generative AI tools has improved my development velocity,GenAI tools help me solve problems faster.,GenAI tools have contributed to my overall productivity,GenAI tools are easy to use in my daily workflow,I find it easy to learn how to use genAI tools for development tasks,It is easy to integrate genAI tools into my normal development process,I would like to continue to use generative AI for software development going forward,The behaviour of generative AI tools is predictable to me in most cases,I feel confident using generative AI tools for tasks that will go into the production environment
7/10/2025 10:29:30,,Male,Sweden,No,5,4,1,Tool; Assistant; StackOverflow replacement,4,4,2,4,5,2,5,2,5
7/14/2025 11:38:06,,Male,Sweden,Yes,3,4,1,Tool,1,2,1,2,2,2,1,3,1
7/10/2025 9:40:16,,Male,Greece,No,5,5,1,Bulk code writer,2,1,2,2,3,2,1,1,1
7/10/2025 9:42:13,,Male,Sweden,No,5,5,1,Tool,3,2,2,1,1,2,1,2,1
7/10/2025 12:38:28,,Male,Sweden,No,4,2,2,Assistant; StackOverflow replacement; Tool; Enabler - makes scaffolding go a lot faster,4,4,5,5,5,5,5,2,5
7/15/2025 8:45:16,,Female,Greece,No,3,4,2,Tool; Teammate; Assistant,3,3,3,4,3,4,4,2,2
7/15/2025 18:56:40,,Female,India,Yes,5,4,2,Assistant; Tool; Co-worker; StackOverflow replacement; Competitor,4,4,4,4,4,4,4,4,3
7/10/2025 9:40:41,,Male,Sweden,Yes,3,4,2,StackOverflow replacement; Assistant; Tool, 4,4,4,4,4,4,5,5,3
7/10/2025 9:42:15,,Male,Greece,No,3,4,2,Tool,2,1,2,3,4,4,2,1,1
7/10/2025 9:44:56,,Male,Sweden,Yes,3,4,2,Assistant; Tool,4,4,4,3,4,4,5,3,4
7/10/2025 10:02:01,,Male,Greece,Yes,4,4,2,Tool; Assistant,4,3,4,2,3,2,5,2,2
7/10/2025 12:14:52,,Male,Sweden,Yes,4,4,2,Assistant; Tool; StackOverflow replacement,3,3,3,4,4,3,5,3,3
7/11/2025 16:38:25,,Male,UK,No,4,4,2,Tool; Assistant; StackOverflow replacement, 5,5,5,5,5,4,5,3,3
7/14/2025 11:16:41,,Male,Sweden,No,2,4,2,Tool,1,1,2,2,2,3,2,4,2
7/16/2025 13:49:33,,Male,Greece,Yes,5,4,2,StackOverflow supplement,3,3,3,3,4,3,3,3,1
7/10/2025 9:38:08,,other / prefer not to say,Greece,No,5,4,2,Tool; Assistant; Stack-

Overflow replacement,4,4,5,5,5,4,5,4,2
7/11/2025 15:41:12,,other / prefer not to say,Sweden,No,3,4,2,Assistant; StackOverflow replacement,3,3,3,3,4,4,5,3,3
7/10/2025 14:12:37,,Female,Uruguay,Yes,5,5,2,Tool,2,3,2,2,3,3,3,2,2
7/10/2025 9:40:41,,Male,Greece,Yes,3,2,3,Tool; Assistant,4,3,2,5,5,4,4,3,4
7/10/2025 10:23:19,,Female,Sweden,Yes,5,3,3,Assistant; StackOverflow replacement; Co-worker; Teammate; Tool,5,5,5,5,5,5,5,5,5
7/10/2025 13:41:08,,Male,India,No,5,3,3,StackOverflow replacement; Tool; Teammate; Assistant,4,4,4,4,4,4,4,2,4
7/10/2025 9:48:00,,Female,Sweden,Yes,4,4,3,StackOverflow replacement; Tool, 4,4,3,4,3,4,5,4,4
7/14/2025 14:48:24,,Female,Sweden,No,3,4,3,Competitor; Teammate; Co-worker; StackOverflow replacement; Tool, 4,4,4,4,4,4,4,3,3
7/10/2025 11:14:36,,Male,Sweden,No,3,4,3,Assistant; StackOverflow replacement; Tool, 5,5,5,5,5,5,5,5,5
7/14/2025 11:55:56,,Male,Sweden,No,5,4,3,Tool; Assistant; StackOverflow replacement, 3,5,4,5,5,5,3,3,3
7/17/2025 20:10:28,,Male,Uruguay,Yes,3,4,3,Tool; Assistant; StackOverflow replacement; Inspiration prompt,4,4,4,2,4,2,5,4,5
7/23/2025 8:42:19,,Male,Sweden,Yes,4,4,3,Assistant; StackOverflow replacement; Tool, 4,3,4,4,4,4,5,2,3
7/11/2025 4:20:46,,Male,Sweden,Yes,5,5,3,StackOverflow replacement; Tool; Assistant; Co-worker,5,5,5,4,5,5,5,4,5
7/10/2025 9:43:38,,other / prefer not to say,Sweden,Yes,5,5,3,StackOverflow replacement; Tool,4,4,4,5,4,4,5,4,3
7/10/2025 9:38:54,,Male,Sweden,Yes,4,4,4,Tool; StackOverflow replacement; Assistant; Co-worker,4,4,4,4,4,3,5,5,4
7/10/2025 9:39:16,,Male,Sweden,Yes,4,4,4,Assistant,5,5,5,4,4,4,4,3,4
7/10/2025 9:43:25,,Male,Sweden,No,4,4,4,Tool; StackOverflow replacement, 5,5,5,5,5,4,5,4,4
7/10/2025 9:49:24,,Male,India,Yes,5,4,4,StackOverflow replacement; Tool; Assistant, 4,5,5,5,5,5,5,5,3
7/10/2025 10:02:38,,Male,Sweden,No,4,4,4,Tool; StackOverflow replacement; Assistant,4,4,4,4,4,4,4,4,4
7/10/2025 10:53:01,,Male,Sweden,Yes,4,4,4,Assistant; StackOverflow replacement; Tool,4,4,4,3,3,3,4,3,4
7/10/2025 11:40:32,,Male,Sweden,No,3,4,4,Tool; StackOverflow replacement; Assistant,4,5,4,5,5,4,5,4,4
7/14/2025 11:04:27,,Male,Sweden,Yes,4,4,4,Assistant; StackOverflow replacement; Tool,4,4,4,4,4,4,5,4,5
7/14/2025 11:07:05,,Male,Sweden,No,5,4,4,Tool; Assistant; StackOverflow replacement; Co-worker,4,4,4,4,5,4,5,4,4
7/22/2025 10:27:51,,Male,Sweden,Yes,3,4,4,Tool; StackOverflow replacement; Assistant,5,5,5,3,4,4,5,3,4
7/14/2025 11:11:31,,Female,Sweden,No,4,4,5,Teammate; Co-worker,5,5,5,5,5,5,5,5,5

C

Appendix 3 - management interview

The following contains the outline for the management interview. The questions were used as a guide for the semi-structured interview, and were not followed one-after-one to allow the interview and discussions to flow more naturally. further below you can find the initial codes found during the thematic analysis of the interview transcript.

C.1 Interview guide/questions

The following questions were used as a guide for the interview/discussion with the manager and team-lead responsible for the genAI tool selection and piloting. The questions were not asked one by one, but rather used as a guide to lead the semi-structured interview from A-Z. References to key terminology and constructs were left included to easily draw references to relevant literature and frameworks.

1: Did you establish selection (and evaluation) criteria (e.g., cost, accuracy, integration effort) to compare ChatGPT, GitHub Copilot, Gemini, and other options? If yes, how? *[TOE–technology fit; integration; cost/benefit; governance/legal]*

2: Did you align the AI-tool selection with your broader product and business strategy? If yes, how? *[Strategic alignment; top-management support; holisticity]*

3: Did you follow a formal change-management model (e.g., Kotter's 8 steps, Bridges' transition model)? If yes, which one and why? *[Change management; sponsorship; communication; adoption path]*

4: Did you put in place training programs or resources to help developers adopt the new AI-assisted workflows? If yes, what did they consist of? *[Onboarding/training; organizational readiness; HAI guidance; enablement]*

5: Did you implement measures to monitor and assure the quality and security of AI-generated code? If yes, what were they? *[Human-in-the-loop; quality gates; security risk controls; testing]*

6: Did you establish governance policies (e.g., code-review gates, audit logs) specific to AI-generated outputs? If yes, what do they look like? *[Governance/-compliance; accountability; auditability; policy]*

7: Did you measure changes in development velocity since AI-tool adoption (e.g., cycle time, ticket throughput)? If yes, how? *[Productivity metrics; before/after design; instrumentation]*
8: Did you compare the actual gains (or losses) in coding speed and quality to your initial expectations? If yes, how? *[TAM–perceived usefulness/ease vs. outcomes; expectation–experience gap]*
9: Did you observe new patterns of collaboration between developers and AI—such as using AI as a 'second brain' versus a tutor? If yes, what emerged? *[Human–AI collaboration; interaction patterns; role metaphors]*
10: Did you consider specific factors to roll out AI tools beyond the initial pilot group? If yes, what were they? *[Org readiness; scaling; environment/ecosystem; support/process fit]*
11: How do you see the role of generative AI evolving in your software lifecycle over the next 3–5 years? *[Strategy; operating model; culture as moderator; lifecycle integration]*

C.2 Thematic analysis

Thematic analysis, 48 initial codes as quotes from the interview transcript, separated by commas: ["the DX, the development experience, usability of it is rather important.", "being able to ask or interact with the LLM … through a UI that is integrated into the development environment.", "generally speaking AIs are extremely slow…", "the responses that came back, you couldn't like copy and paste code… you had to copy the entirety of the code…", "there are quite a few types of problems where the AI tooling really shine…", "request response ones… are more for solving the problem here and now, whereas the session based ones seems to be much better at performing green field bigger operations.", "Gemini and Jet Brains AI assistant are the same… they are requests response-based.", "Juny is a session based one.", "the session based ones seems to be much better at performing green field bigger operations.", "let me first start by creating a set of tasks… place those in a plan file… stash that into a repository…", "they don't know anything about your domain specifics.", "they were emitting garbage—complete and utter garbage—when they get into one of those loops and start rambling.", "we're migrating towards another runtime called WildFly… sometimes… it was just plainly obvious that they were emitting garbage…", "cost is something that is a secondary to us… it's more of defining the yield.", "it took two weeks… to expire the limit of tokens. So, they're cost intensive.", "the cost... in an organization as big as ours… would be high—actually really high.", "the younger… the developer the more they're likely to use AI tooling.", "the intellectual property of the company needs to be protected.", "the legal department had been involved reviewing the terms and conditions… and the license agreements.", "the GENY API… can be used… even… where we have customer data involved.", "also some open-source models that can be hosted internally in our own infrastructure…", "they might use some kind of AI technology in the chatbot area in the CS department…", "we have selected Gemini as a holistic AI tool that can be enabled in other products not only in development tooling.", "we have it enabled in our Google workspace and we can interact with it with business related questions.", "the Gemini evaluation has been going on for quite some time... whereas the evaluation of Jet Brains AI assistant and Juny is roughly a month... maybe six weeks.", "Jet Brains AI assistant... interact with more or several backends—for example Claude or Chatg GPTC.", "Gemini is also, bizarrely enough, less focused on code than the Jet Brains brethren...", "Gemini and Jet Brains AI assistant somewhat equal—maybe slightly worse in Gemini's case...", "management just want to throw this thing out.", "let us please review them a bit first... come up with recommendations so people don't waste quite so much time...", "at the launch we have a sort of a kickoff... introduction to how you can use them in an efficient manner.", "the short answer would be no. it's more of exploratory slash collaborative...", "we had a few years ago an AI-themed hackathon... vendors... delivering some kind of workshops...", "I don't know if it was a management-mandated channel or not...", "we haven't established something formal to measure...", "we based our evaluation... on verbal feedback from the participants...", "after say six months... then we might be able to... find the KPIs...", "we have no intention... introducing some kind of distinction on this code has been produced by humans... by AI tooling.", "in the eyes of the organization the author of the code is the person that initiates the pull request.", "we need to place... some standard guidelines and... policies on how these tools should be applied...", "first and foremost is to boost productivity of the personnel.", "it can be an attractive characteristic... to have... better talent coming in.", "it has been so... invigorating... it has lowered a lot of barriers... I think creativity can take a really good boost with these tools.", "AI agents and the LLMs are not [free] and that will... put some... disturbance... with a subscription... introduce some discrepancy.", "I participate... in the IT governance group... we discuss on bi-weekly basis... cross-cutting technical things...", "we had involved the specific team from the DSR function... they are actually working more with AI technology...", "our domain is... old, very complex and very big.", "this particular tool has better integration... it can produce significantly vast changes in the code base in relatively shorter time."]

D

Appendix 4 - Developer interview guide

The following questions were used as a guide for the interviews with the developer and tester who had been using the genAI tool for during the 4 month period. The questions were not asked one-after-the-other, but rather used as a guide to lead the semi-structured interview from A-Z. References to key terminology and constructs were left included to easily draw references to relevant literature and frameworks.

Role of GenAI in workflow (RQ1).

1: If you had to describe the role of GenAI in your daily work, how would you describe it? (e.g., assistant, partner, helper, background tool, etc.) *[role metaphor; perceived agency; interaction framing]*

2: Do you see GenAI as more of a support tool or something that's actively changing how you work? *[adoption stage; workflow change; perceived usefulness]*

3: Do you ever feel like you're "collaborating" with the AI? If so, what does that feel like? *[human–AI collaboration; turn-taking/dialogue; cooperation pattern]*

4: Has GenAI changed how you interact with your team or how much you rely on colleagues? *[team communication; knowledge-seeking channels; reliance on colleagues]*

Integration into daily workflow and task fit. (RQ3)

1: What kinds of development tasks do you find GenAI most helpful for — and which ones not at all? *[task–tool fit; scope/complexity; boilerplate vs. novel work]*

2: Have you started using GenAI tools habitually for certain phases (e.g., debugging, scaffolding, tests)? *[habit formation; phase-specific use; workflow integration]*

3: Do you feel GenAI is part of your workflow — or something you still experiment with on the side? *[routinization vs. experimentation; normalization; adoption maturity]*

4: Are there tasks where you intentionally avoid using GenAI? Why? *[risk sensitivity; domain constraints; policy/compliance considerations]*

Trust and willingness to rely on GenAI. (RQ4)

1: When GenAI gives you a code suggestion, what makes you trust or distrust it? *[trust cues; predictability; transparency; domain relevance]*

2: Have you ever used AI-generated code without fully understanding it? If so, how did that feel? *[comprehension threshold; accountability; risk tolerance]*

3: Have you encountered wrong or misleading results? How did that affect your use? *[error experience; trust recalibration; safeguard strategies]*
4: Do you trust GenAI-generated code more for some environments (e.g., personal vs. production)? *[context sensitivity; criticality; environment-specific standards]*
5: Have any specific experiences increased or decreased your trust over time? *[learning curve; consistency over time; feedback loops]*

Productivity, autonomy, and problem-solving confidence. (RQ5)

1: Has using GenAI changed how you approach coding problems or design decisions? *[problem-solving strategy; design cognition; exploration vs. exploitation]*
2: Do you feel more or less in control of the solutions you build when GenAI is involved? *[perceived control; autonomy; authorship/ownership]*
3: Do you feel like GenAI frees up time for more creative/challenging tasks — or makes you more passive? *[time savings; cognitive load; motivation; deskilling concerns]*
4: In what ways (if any) has GenAI affected your learning or professional growth? *[skill acquisition; knowledge transfer; scaffolding/feedback]*

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY AND UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden
www.chalmers.se

AVANCEZ
1829

CHALMERS
UNIVERSITY OF TECHNOLOGY