Adoption of AI Tools in Software Development in Germany: Curse or Blessing for the SME Sector?

Robin Blischke

Dissertation written under the supervision of Professor Peter V. Rajsingh.

Dissertation submitted in fulfillment of requirements for the MSc in Management, at the Universidade Católica Portuguesa, 04.01.2025

Abstract

This study investigated the adoption of Artificial Intelligence (AI) tools in Germany's software development sector, focusing on small and medium-sized enterprises (SMEs). AI technologies are reshaping the industry, presenting opportunities and challenges. Using a mixed-methods approach, including 14 expert interviews and a survey of 262 participants, the research identified key factors affected by AI adoption, such as efficiency gains, code quality, competitive pressure, security risks, organizational readiness, technical debt, and ethical concerns. These factors were identified through a literature review, tested via expert interviews, and validated through a survey, adhering to the principle of triangulation.

Currently, AI adoption in German SMEs remains in its infancy, primarily focused on enhancing productivity in routine tasks, with strategic integration still limited. Expert insights highlighted SMEs' agility in deploying off-the-shelf AI tools but noted constraints from limited resources and technical expertise. In contrast, large enterprises (LEs) leverage robust infrastructure and R&D investments for more comprehensive AI integration.

While AI tools were viewed as an efficiency innovation, findings indicated their disruptive potential to democratize coding, bridge skill gaps, and drive long-term transformations. However, systemic barriers, including security vulnerabilities, ethical dilemmas, and insufficient organizational readiness, continue to hinder widespread adoption.

By integrating dynamic capabilities and innovation theories, this research extrapolated AI's trajectory from incremental efficiency gains to disruptive innovation, fundamentally altering workflows and competitive dynamics. The study offers actionable recommendations to foster readiness, address ethical and security concerns, and promote targeted upskilling for a transformative future.

Keywords: artificial intelligence (AI) adoption, software development, small and medium-sized enterprises (SMEs), efficiency innovation, disruptive potential

Title: Adoption of AI Tools in Software Development in Germany: Curse or Blessing for the SME Sector?

Author: Robin Blischke

Sumário

Este estudo investigou a adoção de ferramentas de Inteligência Artificial (IA) no setor de desenvolvimento de software na Alemanha, com foco em pequenas e médias empresas (PMEs). As tecnologias de IA estão remodelando o setor, apresentando oportunidades e desafios. Utilizando uma abordagem de métodos mistos, que incluiu 14 entrevistas com especialistas e uma pesquisa com 262 participantes, a pesquisa identificou fatores-chave impactados pela adoção de IA, como ganhos de eficiência, qualidade do código, pressão competitiva, riscos de segurança, prontidão organizacional, dívida técnica e preocupações éticas. Esses fatores foram levantados por uma revisão da literatura, testados em entrevistas com especialistas e validados por uma pesquisa, conforme o princípio da triangulação.

A adoção de IA nas PMEs alemãs ainda está em estágio inicial, focada principalmente no aumento da produtividade em tarefas rotineiras, enquanto a integração estratégica permanece limitada. Especialistas destacaram a agilidade das PMEs na utilização de ferramentas prontas, mas apontaram restrições relacionadas a recursos limitados e baixa especialização técnica. Em contraste, grandes empresas (GEs) utilizam infraestrutura robusta e investimentos em P&D para integração mais abrangente de IA.

Embora vistas como inovações de eficiência, as ferramentas de IA apresentam potencial disruptivo ao democratizar a programação, reduzir lacunas de habilidades e impulsionar transformações de longo prazo. Contudo, barreiras sistêmicas, como vulnerabilidades de segurança, dilemas éticos e baixa prontidão organizacional dificultam sua ampla adoção.

Integrando teorias de capacidades dinâmicas e inovação, este estudo extrapola a trajetória da IA de ganhos incrementais para inovação disruptiva e oferece recomendações para superar barreiras e promover transformação setorial.

Palavras-chave: adoção da inteligência artificial (IA), desenvolvimento de software, pequenas e médias empresas (PME), inovação em termos de eficiência, potencial disruptivo

Título: Adoção de ferramentas de IA no desenvolvimento de software na Alemanha: Maldição ou bênção para o sector das PME?

Autor: Robin Blischke

Acknowledgements

I would like to acknowledge the invaluable support of those who made the completion of this research possible. First and foremost, I extend my deepest gratitude to my thesis supervisor, Peter V. Rajsingh, whose unwavering guidance and tireless assistance were instrumental to the success of this study. I am also sincerely thankful to the experts who participated in the interviews, as their insights significantly enriched the quality of this research.

Additionally, I express my appreciation to my friends for their constant encouragement throughout this journey. The experiences shared and memories created will be cherished always.

Lastly, I convey my profound gratitude to my brother for exemplary leadership and to my parents, without whom this endeavor would not have been conceivable. I am eternally indebted to them.

This work includes language refinements assisted by AI tools, while the content and ideas remain the sole responsibility of the author.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AGI	Artificial General Intelligence
AI	Artificial Intelligence
AIaaS	AI-as-a-Service
ANI	Artificial Narrow Intelligence
DEV	Developer
DL	Deep Learning
DOI	Diffusion of Innovation
ERP	Enterprise Resource Planning
GPT	Generative Pretrained Transformer
IT	Information Technology
ITK	Information Technology and Telecommunications
LE	Large Enterprise
ML	Machine Learning
NLP	Natural Language Processing
OLR	Ordinal Logistic Regression
PM	Project Manager
R&D	Research & Development
SaaS	Software-as-a-Service
SDLC	Software Development Life Cycle
SME	Small and Medium-Sized Enterprise
TAM	Technology Acceptance Model

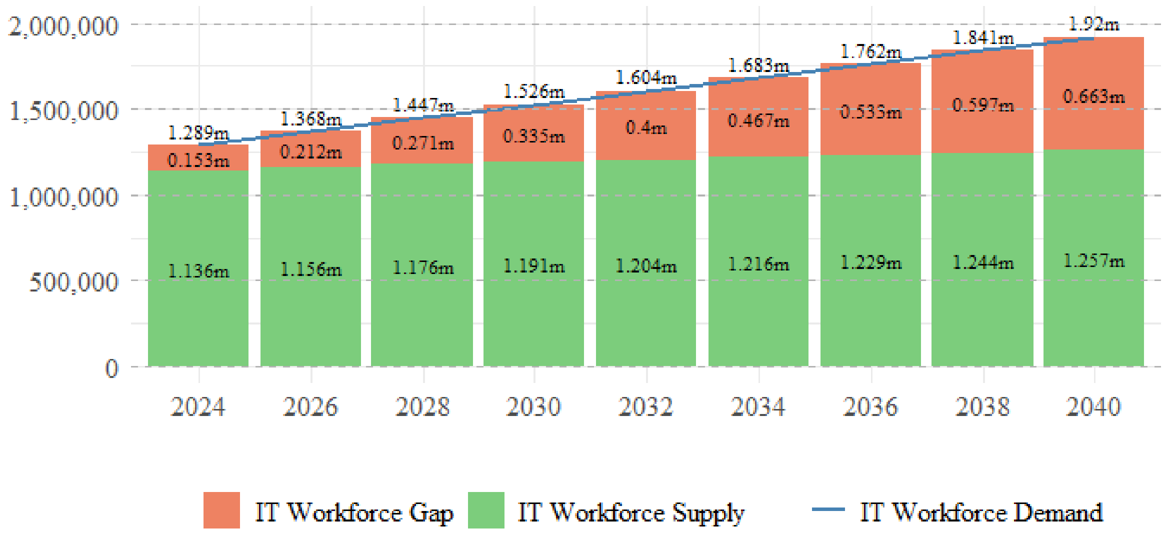
1. Introduction

Software development stands as a cornerstone of contemporary technology, evolving from rigid development methodologies like the waterfall model (Royce, 1987; Cocco et al., 2011) to agile practices that emphasize adaptability, collaboration, and iterative delivery of high-value features (Highsmith, 2009). This evolution reflects the growing demand for innovative solutions to address the programming complexities of a digitized world. Germany, as Europe’s largest software market, accounts for nearly one-fourth of the continent’s software industry value, showcasing remarkable resilience amid global challenges (Germany Trade & Invest, n.d.). The sector has experienced steady growth, driven largely by the dynamic ecosystem of small and medium-sized enterprises (SMEs) that form a substantial customer base for software solutions (Germany Trade & Invest, n.d.).

The country’s software development workforce has also expanded, with approximately 1.1 million developers as of 2023, growing at a rate of 4% per year since 2021 (Bayer, n.d.-b). However, this growth is overshadowed by a pressing challenge: an acute shortage of skilled IT professionals. By 2023, Germany faced 149,000 unfilled IT positions, a number projected to rise to 663,000 by 2040 without intervention as shown in Figure 1 (Bitkom, 2023; Bitkom, 2024). This shortage threatens the nation's ability to meet its digital transformation goals.

Figure 1

Development of the Shortage of IT Workforce in Germany



Note. Adapted from Bitkom (2024)

Amid this talent gap, the rapid integration of artificial intelligence (AI) and automation in software development is reshaping the industry. AI’s capability to streamline tasks like bug fixing, code refactoring, and testing introduces both opportunities and challenges (OECD, 2024). While tools powered by AI, such as GPT models, have revolutionized areas like natural language processing (Devlin et al., 2019), their adoption demands significant upskilling among developers. NVIDIA CEO Jensen Huang highlights the democratizing potential of AI, asserting that "everybody in the world is now a programmer," which underscores its capacity to transform software development by broadening access to programming and reshaping technological innovation (Collins, 2024). Similarly, Li (2015) notes that the future of AI extends beyond intelligence utilization, emphasizing that "we will not only use the machines for their intelligence. We will also collaborate with them in ways that we cannot even imagine," pointing to an evolving human-AI partnership that fosters unprecedented collaboration.

Despite its transformative potential, the adoption of AI in software development is uneven, particularly within Germany’s SME-dominated sector. The investigation of factors such as efficiency gains, job displacement, technical debt, ethical concerns, and organizational readiness remains underexplored. This creates a gap in understanding how AI disrupts traditional workflows and impacts long-term innovation, especially in a context where SMEs drive much of the industry’s dynamism.

The overarching goal of this thesis is to investigate the primary and secondary factors influencing the adoption of AI in software development within German SMEs. Primary factors include efficiency gains, automation, job displacement, and code quality, while secondary factors encompass ethical concerns, organizational readiness, security risks, and competitive pressures. These elements are analyzed through the lens of established management theories, such as dynamic capabilities, innovation theories, and the technology acceptance model, to understand their interplay.

This research contributes to the ongoing discussion of AI's role in software development by addressing gaps in the existing literature. While much has been written about AI's potential to enhance productivity and streamline operations, less attention has been paid to its disruptive implications within Germany’s SME landscape. By examining both the benefits and challenges of AI adoption, this study provides actionable insights for stakeholders navigating this transformation. Moreover, by incorporating management theories, the research offers a robust framework for understanding how organizations can adapt to technological change while maintaining competitiveness. As a result, the central research question guiding this study was:

RQ: How is AI disrupting the German SME software development industry?

The study focuses on Germany's SME software development industry, while also drawing insights from large enterprises for comparative perspective. It explores the drivers of adoption of AI tools in this context, with particular attention to the interplay between technical and organizational factors. The research does not extend to non-software-related sectors or general IT applications, ensuring a targeted analysis of the industry’s specific challenges and opportunities.

This thesis is organized into five chapters. Chapter 1 outlines the research context, objectives, and significance. Chapter 2 reviews literature on software development, AI integration, and innovation theories. Chapter 3 explains the mixed-methods research design, combining expert interviews and surveys. Chapter 4 presents findings and discusses implications in relation to existing theories. Chapter 5 concludes with key insights, limitations, and recommendations for future research. Through this structured approach, the thesis provides a nuanced understanding of AI’s disruptive impact on the German SME software development sector and practical recommendations for navigating this paradigm shift.

2. Literature Review

The literature review explores the software development industry, the influence of AI, and their intersection in driving innovation. It covers how AI transforms development processes and integrates key management theories, such as dynamic capabilities and innovation, to highlight how organizations adapt and stay competitive.

2.1. Introduction to Software Development

Software development is a structured and iterative process that converts user requirements into functional, reliable, and maintainable software systems. As Frederick P. Brooks (1986) famously noted in *The Mythical Man-Month*, "The hardest single part of building a software system is deciding precisely what to build," (p. 13) highlighting the importance of effective planning and requirement analysis in this process.

The software development life cycle (SDLC) comprises distinct stages. It begins with requirements elicitation, where functional and non-functional requirements are gathered from stakeholders. Followed by the analysis and design phase, which centers on developing an optimal system architecture, considering cost-efficiency and precision. The development phase involves implementing the design using suitable programming tools. The testing phase validates software functionality. The last phase, deployment and maintenance stage, ensures the software operates smoothly within its intended environment (Radack, 2009).

Traditional methods like the Waterfall model laid the foundation for software development techniques. The Waterfall model is a structured, sequential approach that progresses through the SDLC phases. Each phase must be entirely completed before progressing to the next (Royce, 1987; Cocco et al., 2011). This model is suited for projects with clearly defined and rigid requirements. However, its rigidity limits its flexibility, leading to inefficiencies and delays in dynamic or complex projects when new requirements arise after a phase is completed (Balaji & Murugaiyan, 2012; Saravanos & Curinga, 2023; Sommerville, 2011).

The advent of agile frameworks marked a transformation in the software development process, addressing the limitations of traditional methods through the integration of collaborative approaches, flexibility, and incremental progress. Scrum, the most used agile framework presently, focuses on project management rather than technical practices. It structures work into fixed-length sprints, during which teams focus on delivering potentially shippable product increments. Scrum centers on key responsibilities that collaboratively focus on prioritizing and delivering the most valuable features (Al-Saqqa et al., 2020; Schmidt, 2016). Scrum's flexibility and focus on incremental progress have made it a widespread choice for managing both small and large-scale software projects.

Various specialized roles are essential for the success of agile methodologies, each contributing uniquely to the SDLC. Agile teams often see overlapping responsibilities where individuals manage their workload, shift tasks based on needs, and participate in decision-making (Highsmith, 2009). For example, a developer may also handle task prioritization or quality assurance, improving flexibility and coordination in smaller, cross-functional teams (Strode et al., 2022).

In agile teams, at the strategic level, the product manager oversees development to align market needs with company goals, supported by the product owner, who manages the backlog to maximize value. Business analysts bridge the gap between business and development, ensuring market demands are met. Scrum masters facilitate Agile processes and manage sprints, while team leads oversee team dynamics, task prioritization, and technical mentoring. Engineering managers lead technical teams, fostering collaboration, with support from software architects who design software structures. Developers handle coding, UX/UI designers focus on usability and aesthetics, and QA engineers and testers ensure quality through rigorous testing. Together, these roles enable efficient and high-quality software development (Alcor, n.d.).

2.2. Software Development Industry in Germany

The subsequent chapters discuss current developments within the software development industry in Germany.

2.2.1. Status Quo

Germany represents the largest software market in Europe, comprising nearly one-quarter of the continent's software market value. Notably, Germany's software market has demonstrated resilience, maintaining stability despite global economic challenges (Germany Trade & Invest, n.d.). According to Bayer (n.d.-b) the software market in Germany has experienced consistent growth within the broader information technology and telecommunications (ITK) sector. In 2023, the software market reached a value of €46.6 billion, representing an increase of 9.8% compared to the previous year. This upward trend is expected to continue, with projections indicating a market size of €51.2 billion by 2025, corresponding to an average annual growth rate of 9.9%.

Major players like IBM, Microsoft, and SAP are key participants in the German software market. However, the market is largely defined by the presence of numerous dynamic and highly specialized SMEs, known as Germany's renowned "Mittelstand". These companies dominate the country's economic and industrial landscape, forming a diversified SME ecosystem with globally active hidden champions. As these SMEs drive demand for software solutions, they also create a substantial customer base. In addition to strong growth potential, there are promising opportunities for emerging suppliers and new market entrants, especially for those offering specialized, industry-specific software products and services (Germany Trade & Invest, n.d.).

The industry has seen steady growth in the number of companies over the years. From 2007 to 2022, the number of information technology (IT) companies increased from 18,605 to 32,992, with a particular focus on software development as a driving force within the sector. The number of software firms reflects the robust demand for digital transformation solutions across industries, positioning Germany as a leader in the European market. Notably, IT companies have played a critical role in this expansion, contributing significantly to Germany's economic resilience and innovation capacity (Bayer, n.d.-a).

The software industry in Germany has also experienced consistent workforce growth. Between 2021 and 2023, employment rose by approximately 4%, from 1.154 million to 1.235 million employees. This trend demonstrates the increasing demand for software development expertise and the central role it plays in Germany's software development industry. Employment forecasts predict further growth, with workforce numbers expected to continue rising through 2025, reflecting the sector's expanding contribution to both the national economy and the global digital marketplace (Bayer, n.d.-b).

2.2.2. Category Creation: Software-as-a-Service (SaaS)

Software-as-a-Service (SaaS) has transformed software delivery, shifting from on-premises systems to cloud-based subscription models. Armbrust et al. (2010) highlight SaaS's ability to deliver scalable, cost-effective solutions, benefiting SMEs by reducing infrastructure requirements and capital costs. Mell and Grance (2011) emphasize SaaS's flexibility through centralized management, continuous updates, and internet accessibility. Cusumano (2010) identifies SaaS as a category-creating innovation that fosters efficiency and democratizes access to advanced software. This model revolutionized the software industry, enabling widespread adoption of enterprise-grade tools and laying the foundation for innovations like AI integration. AI-as-a-Service (AIaaS) is emerging as a promising alternative for organizations struggling with in-house AI complexities, offering scalable, accessible, and cost-effective solutions to simplify AI implementation and accelerate innovation (Lins et al., 2021).

2.2.3. Key Challenges

Key challenges in software development include the pace of technological change, talent shortages, and the need for upskilling. Rapid technological advancements, particularly the integration of AI, are a growing concern for Germany as it strives to maintain global competitiveness. AI's impact accelerates change, requiring companies to adopt new tools and processes swiftly (OECD, 2024). A Bain & UiPath (2023) report notes that 45% of executives foresee AI drastically transforming industries soon, intensifying pressure on developers to acquire skills in AI-driven automation. Workforce readiness remains a challenge, as Germany faces persistent shortages of skilled developers amidst technological shifts. European Tech Insights highlights public concerns over job displacement and reskilling, with 68% of Europeans supporting AI restrictions to safeguard jobs, reflecting the broader economic and social challenges posed by rapid technological adoption (IE, 2023).

Germany's software development industry is challenged by a critical talent shortage, with 149,000 IT vacancies in 2023, up by 12,000 from 2022. This hinders the nation's digital transformation goals, affecting both private and public sectors. IT positions remain unfilled for an average of 7.7 months, exacerbating the supply-demand gap. By 2040, demand for IT specialists is expected to reach 1.92 million, leaving a projected shortfall of 663,000 without intervention (Bitkom, 2023; Bitkom, 2024). Demographics provide short-term relief, as only 32.5% of the current IT workforce will retire by 2040, compared to 50.5% in other sectors. Solutions include expanding IT education, reducing dropout rates in computer science, encouraging lateral entrants, and retaining older workers. Immigration is pivotal, with Germany needing 321,000 IT professionals from abroad by 2040 to bridge the skills gap (Silicon Saxony, 2023; Bitkom, 2024).

AI adoption in software development necessitates extensive upskilling. As AI automates tasks like coding and debugging, developers must shift toward roles involving system architecture and AI tool management. Pluralsight reports that 74% of developers aim to enhance their AI-assisted coding skills, yet 45% express concerns about obsolescence in an AI-driven era.

Continuous learning is essential for maintaining relevance (Hicks, Lee, & Foster-Marks, 2023). The World Economic Forum's Future of Jobs Report (2023) predicts 25% of jobs will undergo significant changes within five years due to AI, compelling businesses to invest in upskilling to boost productivity and close skills gaps (Training Industry, 2023). Developers must master AI ethics, tool usage, and human-AI collaboration to remain effective and enable organizations to maximize AI's potential in software development (Souza & Magalhães, 2024; Kumar et al., 2023).

2.3. Introduction to Artificial Intelligence

The development of AI began with Alan Turing's pioneering 1950 paper "Computing Machinery and Intelligence," which proposed that machines could simulate human thinking. The "Turing Test" established a foundation for future AI research, inspiring efforts to create machines capable of problem-solving and decision-making which could fool a human into thinking that a natural intelligence was the source of responses to prompts (Turing, 1950). The term "Artificial Intelligence" was officially introduced by John McCarthy during the 1956 Dartmouth Conference, marking AI's emergence as a formal academic field focused on replicating human cognition (McCarthy, 2007). Early research centered on symbolic reasoning and rule-based systems, with programs like the General Problem Solver modeling logical human thought processes (Nilsson, 2009).

Despite early enthusiasm, the field faced setbacks due to limited computational power and algorithmic complexity. These challenges led to the AI Winter of the 1970s and 1980s, characterized by declining funding and interest (Harguess & Ward, 2022). Symbolic AI's inability to address real-world complexities and to parse natural language highlighted the limitations of manually encoded rules (Crevier, 1993).

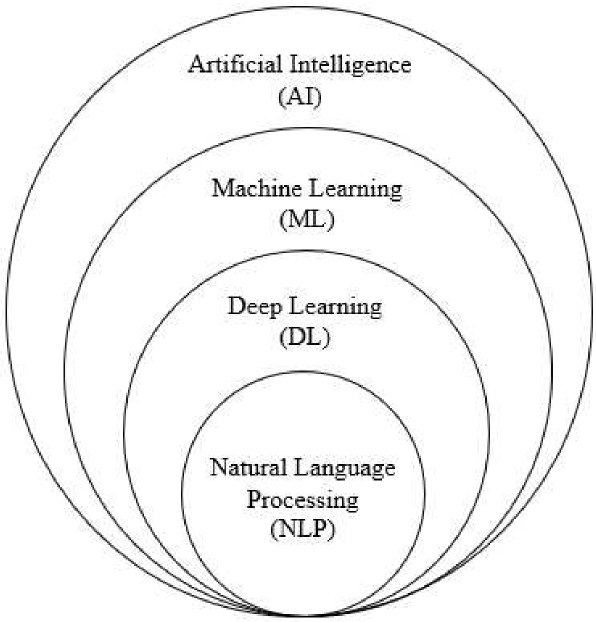
AI saw a resurgence in the 1980s with advancements in neural networks. The backpropagation algorithm introduced by Rumelhart et al. (1986) enabled systems to learn from data, marking a shift from symbolic AI to machine learning. This era saw breakthroughs in speech recognition, image classification, and language translation, demonstrating AI's potential to move beyond rigid, rule-based approaches (IBM, n.d.; LeCun, Bengio, & Hinton, 2015).

The 2010s witnessed rapid advancements in deep learning (DL), a subset of machine learning (ML) using multi-layered neural networks to process large datasets. Enhanced computational power and big data integration allowed AI systems to excel in complex tasks like computer vision and natural language processing (NLP) (Russell & Norvig, 2021), as illustrated in Figure 2. Transformer architectures, exemplified by models like GPT, revolutionized natural language processing, enabling machines to generate coherent text, translate languages, and answer questions with remarkable accuracy (Devlin et al., 2019).

Contemporary AI is pervasive, powering virtual assistants, autonomous vehicles, and numerous industry applications (Lennox, 2020; Google Cloud, n.d.). While current AI systems are artificial narrow intelligence (ANI), designed for specific tasks, research into Artificial General Intelligence (AGI) aims to create machines capable of performing any intellectual task humans can (Kissinger et al., 2021). Achieving AGI remains speculative but could transform industries, society, and human-technology interactions (Goertzel, 2014)

Figure 2

Hierarchical Structure of Artificial Intelligence and Its Subfields



Note. Adapted from IBM (n.d.) and Russell & Norvig (2021)

2.4. AI Tools in Software Development

As Li (2015) noted, "[…] We will not only use the machines for their intelligence. We will also collaborate with them in ways that we cannot even imagine […]", emphasizing the evolving partnership between AI and humans. The following chapters will examine how AI adoption impacts primary and secondary factors, ultimately transforming the software development industry.

2.4.1. Primary Factors

Primary factors which have a direct impact on the software development industry involve changes in development efficiency, modifications to code quality, potential job displacement, and challenges that may arise in managing technical debt.

2.4.1.1. Efficiency

AI integration in software development has significantly boosted productivity and efficiency. A survey with 65,000 developers from Stack Overflow (2024) reveals the widespread use of AI tools like ChatGPT (82.1%), GitHub Copilot (41.2%), and Google Gemini (23.9%), with 81% of developers highlighting productivity gains. These tools assist with coding (82%), answer searches (67.5%), and debugging (56.7%). Research highlights AI's role in automating tasks, saving time, and improving code quality, allowing developers to focus on complex tasks.

Weber et al. (2024) and Daigle (2024) emphasize that tools like GitHub Copilot streamline coding, documentation, and testing. Rajbhoj et al. (2024) report a 70.66% productivity boost by reducing reliance on technical expertise and automating phases of the SDLC. Ebert and Louridas (2023) also highlight reduced cognitive load and faster development cycles through task automation. However, some researchers caution against over-reliance on AI, which may limit developers' autonomy and problem-solving skills (Coutinho et al., 2024; Peng et al., 2023). Concerns also exist around AI's limitations in tasks requiring deep contextual understanding, transparency, and ethical decision-making (Kokol, 2024; Bano et al., 2024).

AI tools enhance project management by automating administrative tasks, offering real-time analytics, and aiding strategic decision-making, as noted by Kanbur, Prakash, and Kulkarni (2023). Another significant benefit of efficiency gains is cost reduction, as AI automates repetitive tasks and streamlines various phases of the SDLC. By reducing the need for manual coding, AI tools enable faster and more consistent results (Weber et al., 2024). However, initial costs for custom AI models and ongoing maintenance remain challenges, though time savings often offset these expenses (Daigle, 2024). Despite short-term cost reductions, managing long-term expenses is a critical consideration.

2.4.1.2. Code Quality

AI tools like ChatGPT and GitHub Copilot improve code quality by producing cleaner, more consistent, and concise code. Dantas et al. (2023) highlight benefits such as enhanced readability and adherence to naming conventions, crucial for software quality. However, human oversight is necessary to ensure alignment with current practices and semantic accuracy, especially as AI performance declines with task complexity (Liu et al., 2024).

Yetiştiren et al. (2023) emphasize software maintenance concerns, noting that AI-generated code often suffers from technical debt and "code smells", requiring developer intervention for long-term maintenance. Beck and Fowler (1999) describe code smells as indicators of potential design errors in the code that point to the need for a refactoring to improve readability, simplicity and maintenance. Ebert and Louridas (2023) also warn of AI's non-deterministic output, which can introduce security risks. Despite AI's benefits in automating tasks, substantial human involvement is essential to ensure code reliability and quality.

2.4.1.3. Job Displacement

The adoption of AI in software development is transforming workforce dynamics, with AI automating various routine tasks, enabling workers to focus on more creative and complex activities (Chen and Tajdini, 2024). Babashahi et al. (2024) highlight that while some traditional roles are displaced, new ones like AI system managers and machine learning engineers are emerging.

Meesters et al. (2022) add that AI creates multidisciplinary roles requiring a blend of technical, analytical, and strategic skills, such as AI engineers. This shift emphasizes the need for skill transformation to navigate these changes (Filippi et al., 2023). Naveen et al. (2023) warn that traditional software roles may disappear but note that specialized positions, like software forecasters and AI system specialists, mitigate total job displacement by requiring human oversight of AI.

Kanbur et al. (2023) and Filippi et al. (2023) mention that AI also generates insights for decision-making, leading to roles like AI strategists and project analytics managers. Overall, AI is not just replacing jobs but creating new roles that require higher-order thinking and human-AI collaboration.

2.4.1.4. Technical Debt

Technical debt in software development refers to the extra work and costs needed to fix issues caused by using quick, less optimal solutions (Jaspan & Green, 2023; Tom et al., 2013). While AI tools boost productivity, they also risk increasing technical debt due to improper code generation and limited adaptability for complex tasks (Yetiştiren et al., 2023). AI-generated code often introduces cognitive complexities that make long-term maintenance harder.

Studies emphasize the need for continuous human oversight. AI struggles with non-standard requirements like scalability and security, which, if unmanaged, exacerbate technical debt (Armyanova & Aleksandrova, 2023; Russo, 2024). Over-reliance on AI can lead to inflexible solutions and increased maintenance challenges (Bader et al., 2021). Additionally, AI's unpredictability and lack of explainability further complicate long-term adaptability and software quality (Ebert & Louridas, 2023).

2.4.2. Secondary Factors

Secondary factors which have an indirect impact on the software development industry involve organizational readiness, security risks, ethical concerns, and competitive pressure related to AI integration.

2.4.2.1. Organizational Readiness

The literature consistently emphasizes that organizational readiness is a critical factor for successful AI adoption within the software development sector. Saleem et al. (2023) argue that this readiness extends beyond mere technological capability to include cultural shifts within organizations, particularly regarding workforce upskilling and continuous training. The introduction of AI tools brings a level of complexity that requires employees to not only learn new technical skills but also adopt a mindset that embraces innovation and change. Russo (2024) expands on this by stressing the importance of structured training programs to equip employees with the necessary skills to handle AI-driven transformations. This approach not only addresses skill gaps but also mitigates resistance to change, a common barrier to AI integration.

Both Saleem et al. (2023) and Russo (2024) underscore the necessity of fostering a culture of learning and innovation, which is vital for navigating the challenges posed by AI in software development. In this regard, organizational readiness involves creating an environment where continuous learning is encouraged, allowing employees to adapt to new roles and workflows as AI reshapes traditional software development processes. Holmström and Carroll (2024) add that this cultural transformation is indispensable for maintaining competitiveness in an AI-driven landscape, highlighting the need for companies to redefine roles and processes to fully harness AI's potential.

2.4.2.2. Security Risks

The increasing integration of AI tools into software development introduces significant security vulnerabilities, alongside hidden biases. As highlighted by Chui et al. (2023), the unchecked adoption of these technologies can lead to security flaws, making systems particularly susceptible to cyberattacks. The complex nature of AI-driven processes necessitates robust governance frameworks and continuous human oversight to manage the inherent cybersecurity risks. This complexity is further illustrated by Russo (2024), who notes that while AI tools can enhance the identification of security vulnerabilities, they also present challenges related to biases in training data and the absence of explainability in decision-making processes. This duality underscores a critical dilemma: developers must leverage the advantages of AI while remaining vigilant against its pitfalls.

Moreover, the reliance on AI-generated code can lead to unintended consequences, as highlighted by Nguyen-Duc et al. (2023). They emphasize that biases present in training data can directly impact software quality assurance activities, necessitating manual corrections to identify and rectify vulnerabilities. The authors caution against over-reliance on AI systems, as this may expose software projects to novel cybersecurity threats. Such vulnerabilities, as discussed by Humphreys et al. (2024), include data poisoning and adversarial prompting, which can manipulate outputs and compromise sensitive information. This manipulation is exacerbated by the inherent complexity of generative AI models, making it essential for organizations to establish strict data handling protocols to mitigate risks associated with data leakage and privacy violations.

2.4.2.3. Ethical Concerns

AI tools in software development also raised ethical concerns related to privacy, accuracy, and intellectual property (Baddi et al., 2024). Thomson and Schmoldt (2001) note that AI influences decision-making, making ethical practices crucial for successful outcomes. Gogoll et al. (2021) argue that traditional Codes of Conduct are too vague and advocate for incorporating ethical deliberation into agile processes to address ethical issues dynamically as projects evolve.

Johnson and Smith (2021) emphasize the societal risks of AI, such as bias and safety failures, stressing the need for comprehensive ethical frameworks. Ali et al. (2023) highlight the gap between AI ethics principles and corporate practices, suggesting government regulation and penalties as solutions. Al-Kfairy et al. (2024) further address challenges like intellectual property, reinforcing the need for internal ethics and external oversight in AI development.

2.4.2.4. Competitive Pressure

The integration of AI in software development offers both opportunities and risks. Russo (2024) and Chen (2024) emphasize that companies slow to adopt AI technologies face a significant competitive disadvantage. AI automates complex tasks, leading to a "paradigm shift" in development, with firms that lag behind struggling to maintain market relevance. Schneider et al. (2024) argue that AI lowers costs for data migration, integration, and user training, enabling new entrants to challenge established firms. Chui et al. (2023) predict that generative AI will disrupt the competitive landscape within three years, intensifying pressure on firms to act quickly. Bhavsar et al. (2019) further note that AI enhances productivity, giving early adopters a competitive edge. Collectively, these authors argue that AI adoption is essential for sustaining market power in the rapidly evolving software development sector.

2.5. Key Concepts of Innovation

Innovation serves as a critical factor in fostering organizational resilience and long-term success in dynamic environments. Schumpeter defined creative destruction as the process that "incessantly revolutionizes the economic structure from within, incessantly destroying the old one, incessantly creating a new one" (Schumpeter, 2006, p. 83). Rogers considers innovation as "an idea, practice, or object that is perceived as new by an individual or other unit of adoption," highlighting that innovation is determined by its perceived novelty (Rogers, 1983, p. 11). Both perspectives emphasize the transformative power of innovation in reshaping markets and driving progress.

Building on these foundational views of innovation, the following sections present specific frameworks that guide organizational adaptation and strategic decision-making.

2.5.1. Dynamic Capabilities

The dynamic capabilities framework is useful for understanding how firms adapt to rapidly changing environments. As Teece et al. (1997) state, “a firm’s dynamic capability refers to its ability to integrate, build, and reconfigure internal and external competences to address rapidly changing environments” (p. 98). Building upon this, Teece (2007) emphasizes that the key to sustained profitable growth is the capability to recombine and reconfigure assets and organizational structures as markets and technologies change, highlighting the critical role of adaptability in sustained success.

Scholars offer nuanced perspectives on the efficacy of dynamic capabilities in securing competitive advantage. Eisenhardt and Martin (2000) define dynamic capabilities as specific and identifiable processes such as product development, strategic decision-making, and alliancing, emphasizing their role in integrating and reconfiguring resources. They acknowledge that these capabilities are necessary but not sufficient conditions for competitive advantage, suggesting inherent limitations.

Barreto (2010) expands on this by viewing dynamic capabilities as a multidimensional construct formed by a firm's propensity to sense opportunities and threats, make timely decisions, and alter its resource base. This perspective aims to overcome some of the limitations of earlier definitions, highlighting the complexity of systematically solving problems in dynamic markets.

O'Reilly and Tushman (2008) focus on the managerial aspect, claiming that the ability of senior managers to seize opportunities through the arrangement and incorporation of both new and existing assets is at the core of dynamic capabilities. They introduce the concept of ambidexterity, stating that “ambidexterity, the ability of a firm to simultaneously explore and exploit, enables a firm to adapt over time.” (p. 185). However, they also note that “ambidexterity as a dynamic capability is not itself a source of competitive advantage but facilitates new resource configurations that can offer a competitive advantage” (p. 196). They link these capabilities to activities that foster coordination and organizational learning, such as innovation and alliances.

In contrast, Collis and Anand (2019) caution against overestimating the effectiveness of dynamic capabilities. While they enable firms to integrate, build, and reconfigure internal and external resources, they argue that dynamic capabilities are not the ultimate source of sustainable competitive advantage due to important limitations like imitation and the tension between flexibility and efficiency.

2.5.2. Disruptive, Sustaining, and Efficiency Innovation

Scholars provide a multi-faceted analysis of innovation types, emphasizing the distinctions between disruptive, sustaining, and efficiency innovations. Disruptive innovation arises in low-end or new-market footholds, targeting neglected or non-consuming customers, and gradually moves upmarket to challenge established firms. Christensen, Raynor, and McDonald (2015) state that misapplying the term “disruptive” risks weakening its strategic value, a point reiterated across their work (Christensen & Raynor, 2003). Sustaining innovations, in contrast, enhance existing products for high-end customers without creating new markets, focusing on improving performance within existing frameworks. This distinction is crucial for strategy, as sustaining innovations are less likely to challenge incumbents directly. The authors argue that different innovation types require tailored approaches for effective decision-making (Christensen & Raynor, 2003). Efficiency innovations, designed to reduce costs and boost productivity, can prioritize short-term gains at the expense of long-term growth. Christensen, Raynor, and McDonald (2015) warn that an overreliance on efficiency innovations may stifle growth by neglecting market-creating innovations, which expand access and foster economic development (Christensen & van Bever, 2014).

Critics, such as Markides (2006), argue that conflating types of innovation can obscure their distinct competitive effects. Radical innovations, which change consumer behavior, should be treated separately from disruptive innovations, as each requires different strategic responses. This reinforces concerns that failure to distinguish between innovation types, like business model and product innovations, weakens theoretical frameworks guiding managerial decisions.

2.5.3. Technology Acceptance Model and Diffusion of Innovation

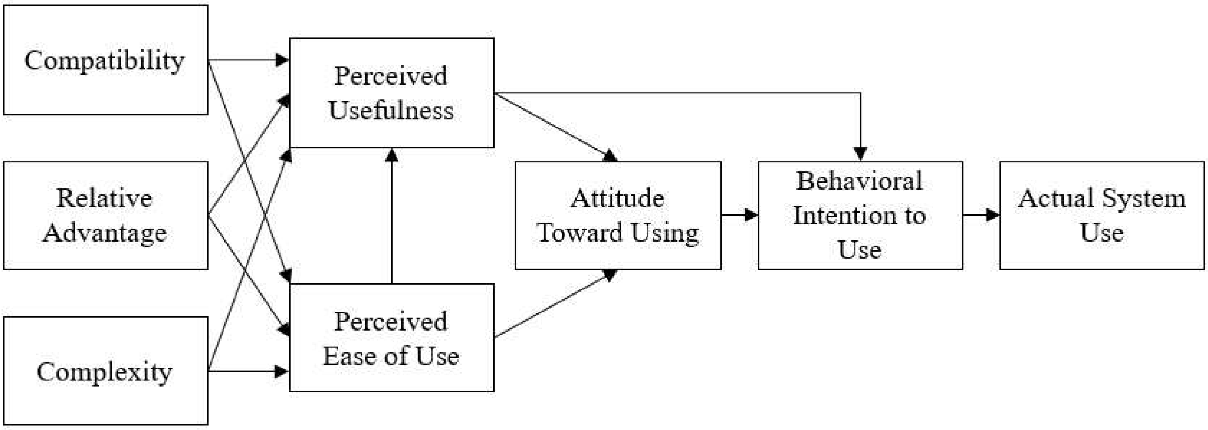
The integration of the Technology Acceptance Model (TAM) with the Diffusion of Innovation (DOI) framework provides companies with a more comprehensive understanding of technology adoption. TAM, through its core constructs, perceived usefulness and perceived ease of use offers a foundational model for predicting user behavior and acceptance (Russo, 2024). However, critics question its applicability across all technological innovations, suggesting that its explanatory power may be limited without the inclusion of complementary frameworks like DOI. By incorporating constructs from both TAM and DOI, such as the relative advantage of new technologies (Rogers, 1983) and the importance of perceived system characteristics (Davis, 1985), businesses can better navigate the complexities of technology adoption.

This complementary approach allows companies to enhance their interventions and manage the adoption process more effectively. For example, TAM's emphasis on early user motivation testing and prototype development (Davis, 1985) aligns with DOI's focus on trialability and compatibility, providing a strategic advantage to companies in managing user concerns and ensuring a smoother integration of new technologies (Rogers, 1983). Furthermore, the blending of these models allows companies to address the nuances of different organizational contexts, as suggested by Lou and Li (2017), who argue that combining TAM and DOI as shown in Figure 3 enables businesses to respond more dynamically to technological shifts.

By leveraging both TAM and DOI, companies can design targeted interventions that address not only the perceived ease of use and usefulness but also broader factors like innovation complexity and organizational fit. This holistic approach not only increases the likelihood of successful adoption but also provides managers with practical tools for assessing the readiness of their organization for new technology (Venkatesh et al., 2003). The critical evaluation of these models suggests that while TAM provides a solid base, its integration with DOI or other frameworks enhances its applicability in real-world business scenarios.

Figure 3

Combination of Technology Acceptance Model and Diffusion of Innovation



Note. Adapted from Lou and Li (2017).

3. Research Methodology

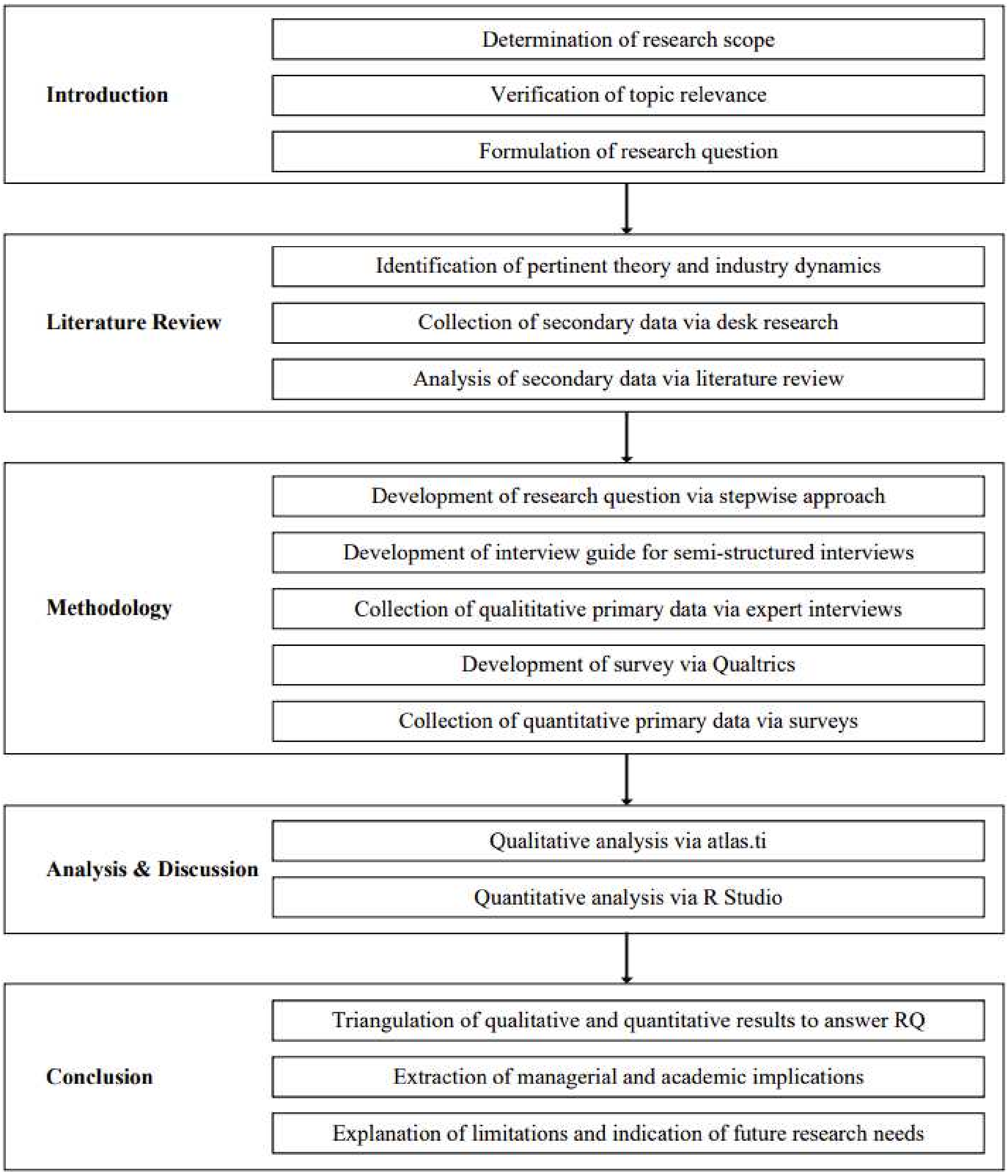
Saunders et al. (2007) define research as a systematic investigation designed to uncover new knowledge, following structured procedures to ensure the validity and reliability of the results. The following sections outline the methods employed in this research.

3.1. Research Design

This research investigates the impact of AI technologies on software development in Germany. Figure 4 below outlines the methodology applied.

Figure 4

Research Design



The study employed a mixed-methods approach, incorporating triangulation through the integration of secondary data from reviewing literature, qualitative primary data from expert interviews, and quantitative primary data from the survey (Saunders et al., 2007). The literature review identified factors influenced by AI adoption in software development. To assess these factors, first semi-structured interviews were conducted with experts from Germany, including developers and project managers from SMEs and large enterprises. Expert interviews were identified as a suitable method for obtaining in-depth insights (Qu & Dumay, 1995). A subsequent survey was conducted to validate the insights obtained from the interviews (Solans-Domènech et al., 2019). The analysis employed both inductive and deductive reasoning, ensuring a comprehensive and reliable evaluation of the findings (Woiceshyn & Daellenbach, 2018).

3.2. Data Collection

The following chapters discuss the data collection methods applied.

3.2.1. Primary Data – Expert Interviews

Qualitative interviewing is used to gain deep insights into participants' perspectives and experiences, allowing researchers to explore complex social phenomena in detail. It enables the collection of rich, nuanced data, which can reveal underlying meanings and contexts that are essential for understanding the research subject (Bryman & Bell, 2011). Semi-structured interviews were chosen as they are beneficial by harmonizing structure and flexibility, allowing to obtain comprehensive, detailed insights while preserving a level of uniformity between interviews for comparability (Rowley, 2012).

The interview guide was developed using an inductive approach grounded in the literature review and comprised 16 questions covering six key areas: background of interviewee, AI adoption, primary and secondary factors, management theories, SME/LE comparison, and general perceptions of the future potential of AI in software development. Interviewees were recruited through the researcher’s personal network and referrals, with interviews lasting an average of 1 hour and 4 minutes, conducted either face-to-face, via Microsoft Teams or e-mail. The 14 participants were categorized by their roles as project managers (PM) and developers (DEV) and company size (SME or LE) for comparative analysis. Responses were analyzed via atlas.ti with deductive coding to validate or refute contemporary theories obtained from the literature. Table 1 provides an anonymized overview of the experts, including their current roles, areas of expertise, and professional experience. Thus, the study aligns with the ideal range for data saturation, typically achieved within twelve interviews (Guest et al., 2006). The interview guide can be found in Table 2 in Appendix B and summaries of the transcripts in Appendix E.

Table 1

Coding and Description of Interviewees

Code	Description of Interviewee
PM1	Digital Solutions Project Manager at an LE renewable energy company, leading AI initiatives. 3 years in digital transformation, deploying large language models.
PM2	AI Project Lead at an LE German automotive company, driving AI adoption in production and logistics. 6 years in digitalization and AI, with an engineering background.
PM3	Team Leader in Data & AI at an LE energy company, managing AI product deployment. 12+ years in digital transformation, with expertise in data strategy.
DEV1	IT specialist at an SME with 18 years' experience. Developed a full ERP system and holds a computer science degree.
DEV2	Freelance Senior Software Engineer with 10+ years in development, serving SMEs. Self-employed since 2017 with versatile programming expertise.
DEV3	Managing Director at an SME specializing in ERP systems, with 10+ years in IT. Focus on supply chain and production planning solutions.
DEV4	IT Specialist at an SME logistics company with 11+ years in programming. Skilled in SQL, C#, and transitioned from wholesale into IT.
DEV5	AI Consultant in an SME with 3 years' experience in AI security and explainable AI. Background in business informatics and hands-on AI development.
DEV6	IT professional with 30+ years' experience, serving SMEs since founding a business in 1999. Offers a wide range of IT services from hardware to programming.
DEV7	CEO of an SME with 11+ years in software development. Focus on customer inquiry automation and ERP system integration for SMEs.
DEV8	IT Manager at an SME with 30+ years' experience in infrastructure and programming. Background in electrical engineering and security management.
DEV9	Systems Engineer in an SME specializing in renewable energy, with expertise in meteorology and AI. Focuses on ERP implementation and process optimization.
DEV10	Senior IT consultant and developer at an LE with 5+ years in Dynamics 365, IoT, and cloud solutions. Holds a degree in computer science, driving IT innovation.

DEV11	Software developer and project manager at an SME with nearly 10 years in hardware-related software. Specializes in microcontroller programming and AI tools.

3.2.2. Primary Data – Survey

The literature review revealed both direct and indirect impact factors of AI tools on the software development sector. To explore these factors further, we developed a survey based on the conceptual model (see Appendix A), which modeled the survey questions detailed in Table 4 in Appendix D.

The survey consisted of 29 questions incorporating demographic questions, multiple-choice items, 5-point Likert scales, matrix formats, and rating scales across six key domains: (1) demographics, (2) AI tool use and integration, (3) primary factors, (4) secondary factors, (5) ranking factors by level of significance, and (6) perceptions of AI's future in software development. One attention-check question was implemented to identify erroneous responses. The survey was conducted online via Qualtrics, targeting a German sample aged 18 and older with a background in software development or a related field, without any researcher interaction during data collection. The survey employed non-probability sampling techniques, specifically convenience and snowball sampling. It was distributed in relevant forums, social media groups, and within a professional network to target individuals with pertinent expertise. This approach facilitated efficient data collection, with participants referring others to expand the sample. For an estimated population of approximately 1 million software developers (Bitkom, 2024), a sample achieving a 5% to 10% margin of error at a 95% confidence level is recommended (Bartlett et al., 2001). Sample sizes of 200 to 400 are typically sufficient for populations of this magnitude (Krejcie & Morgan, 1970).

A total of 375 participants initiated the survey, with 269 completing it. Among these, 7 responses failed to meet the qualifying criteria and were excluded. This left a final sample size of n=262.

4. Analysis and Discussion

This chapter parses the expert interviews presented in Section 4.1, offering insights into perspectives on the adoption of AI tools in software development in Germany. It is followed by an analysis of the survey results in Section 4.2, which highlights patterns and trends relevant to understanding the influence of AI tool adoption on the software development industry in Germany.

4.1. Expert Insights

A deductive coding approach was chosen as it enables the systematic testing of existing theories by applying predefined codes derived from theoretical frameworks. This method ensures alignment with established constructs and allows for the validation or refutation of theoretical assumptions by categorizing data within clear conceptual boundaries, enhancing the rigor and credibility of the analysis (Miles, Huberman, & Saldaña, 2014).

Coding themes were summarized in Table 5 in Appendix D, detailing code frequencies per interview and references to specific theories. The findings are analyzed in subsequent chapters, categorizing drivers and impediments of AI tools' impact on software development, organizational dynamics, management theory insights, and differences between SMEs and LEs.

4.1.1. Adoption Drivers

Experts identified efficiency and competitive pressure as drivers of AI tool adoption in software development and ranked them by code frequency from the deductive analysis.

The findings from the expert interviews highlighted varying perspectives on AI's role in enhancing software development efficiency. PM1, DEV1, DEV3, and DEV6 emphasized the productivity benefits of AI tools like GitHub Copilot and ChatGPT. PM1 noted that these tools streamlined coding and documentation. DEV1 highlighted AI’s ability to automate repetitive tasks, while DEV3 emphasized its role in enhancing debugging time. DEV6 agreed, pointing out that AI improved productivity by handling routine tasks. Conversely, PM2, DEV6, and DEV7 expressed concerns about over-reliance on AI. PM2 warned that AI might erode developers' critical thinking. DEV6 emphasized that AI could not replace human creativity or domain expertise, and DEV7 underscored the risk of stifling complex problem-solving. DEV9, DEV10, and PM3 also raised concerns about the high financial costs associated with AI adoption. PM3, DEV3, and DEV9 discussed AI’s limited contribution to decision-making and project tracking in project management. However, PM2 and DEV7 suggested that AI's use remained largely limited to administrative tasks rather than strategic purposes.

The competitive pressure from AI adoption found strong alignment among experts, who emphasized the risks of lagging in implementation. PM1, PM2, DEV5, and DEV9 stressed that AI integration is no longer optional but essential for sustaining competitiveness. PM1 noted how AI tools have become integral to workflows, driving significant productivity gains. Similarly, PM2 highlighted that companies adopting AI at a slower pace risk increasing inefficiencies in the competitive landscape. The democratizing effect of AI was also emphasized, with DEV6 and DEV8 pointing to AI's ability to lower barriers for non-experts to contribute. PM1 provided an example of using AI tools to address skill gaps and enable broader participation in software development. However, experts acknowledged challenges in keeping pace. PM2 and DEV7 observed that smaller firms with limited resources might struggle to leverage AI effectively, highlighting disparities in readiness. Overall, experts agreed that failing to integrate AI not only jeopardizes a firm's market position but also threatens its long-term relevance. Early adoption was seen as critical for securing efficiency, innovation, and sustained market presence.

4.1.2. Adoption Impediments

Experts identified security risks, job displacement, organizational readiness, technical debt, and ethical concerns as impediments to AI adoption, ranked by code frequency from deductive analysis.

Expert interviews highlighted security vulnerabilities associated with AI tools in software development. Risks such as GDPR non-compliance, external server use, and data leakage were identified by PM1, DEV5, and DEV9. DEV11 noted that employees in large corporations sometimes exploited AI systems to gain unauthorized access to databases, compromising critical credentials such as usernames and passwords. These observations underscored the need for robust governance frameworks to address both technical and human vulnerabilities. The need for manual intervention to address biases in AI-generated outputs was another recurring theme. PM2, DEV7, and DEV10 emphasized challenges posed by the lack of transparency in AI-generated code, which complicated debugging processes and heightened susceptibility to data poisoning. These issues demonstrated the critical importance of addressing both technical and ethical risks to ensure secure and reliable AI implementation.

Experts emphasized the emergence of new roles and skill demands driven by AI in software development. PM1 and PM2 observed a shift toward problem-solving and AI oversight tasks. Additionally, PM1, DEV4, DEV7 stressed that AI adoption required new skills, particularly in problem-centered thinking and higher-order cognitive abilities. However, there was divergence on the extent of job displacement. Most experts argued that rather than eliminating jobs, AI reshaped them, with employees who failed to adapt being at risk of displacement. PM1 noted that the increase in efficiency through AI might make certain tasks redundant, but new roles requiring human oversight of AI processes have emerged. While DEV3 and PM3 emphasized the continued need for human oversight in traditional programming tasks like debugging and writing code, DEV9 and PM2 highlighted the potential for automation in specific tasks, such as developing software tests. Ultimately, experts generally viewed AI as reshaping rather than eliminating roles, requiring the workforce to evolve with the technology and adapt to AI-augmented environments.

Expert interviews highlighted the necessity of structured training and fostering a culture of innovation to prepare the workforce for AI-driven transformations, revealing role-specific nuances. DEV4 and DEV7 emphasized that non-technical roles, such as project managers, required significant training to understand AI workflows and overcome resistance to change. Conversely, DEV6 and DEV8 noted that developers, often early adopters, typically integrated AI tools independently, reducing their need for formal training. However, DEV11 and PM3 cautioned that this independence risked neglecting ethical, security, and systemic concerns, emphasizing the importance of targeted interventions to address these gaps. Experts also stressed the need for role adaptation. PM1, DEV9, and DEV10 observed the emergence of hybrid roles blending AI expertise with traditional functions, such as debugging and strategic problem-solving. While developers were generally well-positioned to adapt, DEV2 and DEV5 highlighted the need for organizations to clearly define these roles and provide structured pathways for skill development. Informal peer learning, as noted by DEV6 and DEV8, proved effective among technical teams, fostering innovation and collaboration. However, DEV4 and PM3 argued that such approaches were insufficient for non-technical teams, where structured programs remained vital. While developers' adaptability reduced reliance on formal training, broader organizational needs, including ethical alignment and role clarity, required targeted programs. Balancing informal learning for technical teams with structured support for non-technical roles appeared to be a strategic effort to promote readiness and competitiveness.

Experts underscored the risk of technical debt and maintenance challenges inherent in AI-generated code, driven by cognitive complexity and its limitations in handling complex tasks. PM1, DEV1, and DEV5 noted that while AI accelerates coding, it often produces functional solutions with inadequate documentation, especially for more complex tasks, complicating long-term maintenance. DEV7 noted that over-reliance on AI by less experienced developers exacerbated these issues, as insufficient human validation led to errors embedded in foundational codebases. The need for continuous human oversight was widely acknowledged. PM3 and DEV9 stressed the necessity of thorough review to address AI's inability to effectively manage scalability and security issues. DEV4 and PM2 found that debugging AI-generated code, especially in more complex software projects, is a major challenge due to the complexity of the code and the lack of standardized procedures. These issues directly contributed to technical debt, particularly in large-scale projects where AI-generated code was implemented without thorough contextual understanding. Maintenance issues also arose from AI's tendency to prioritize immediate functionality over long-term adaptability. DEV11 and PM1 highlighted how AI's lack of explainability complicated updates or refinements, as developers often struggled to interpret or improve the generated code. This lack of modularity and foresight led to significant inefficiencies during software updates, posing challenges to sustainable software development.

The expert interviews highlighted ethical concerns in AI, particularly the need for frameworks to address privacy, intellectual property, and bias. PM1 and PM2 emphasized the importance of structured ethical oversight, with PM1 adopting tools like "LangDoc" to comply with GDPR. However, PM2 noted that compliance processes often caused corporate inertia, presenting challenges in balancing regulation with agility. Intellectual property issues were a major concern, with PM1 and DEV6 citing ambiguities in ownership of AI-generated content. Experts expressed frustration over the lack of clear policies, underscoring a gap between corporate practices and the ethical considerations required for responsible AI use. Bias was another divisive topic. PM3 and DEV8 raised concerns about systemic inequities stemming from flawed training data, while DEV4 and DEV10 argued that human validation could sufficiently address these risks. Safety challenges were also widely acknowledged, especially in high-stakes decision-making scenarios. PM1 and PM2 pointed out the dangers of delegating critical decisions to opaque AI tools. Overall, experts agreed that it is the user's responsibility to reconcile the results of AI tools with established ethical values.

All experts recognized the ability of AI tools to improve the code quality of simpler requirements by producing structured, consistent and well-documented results. PM1, DEV6 and DEV3 highlighted advantages such as better readability and adherence to naming conventions, which improve ease of maintenance, if the accuracy and quality of the input for the AI is kept high. Similarly, DEV7 and DEV9 noted that automation reduces errors and enhances the capabilities of novice developers. However, experts expressed differing views on the scope of these benefits. PM1 and DEV4 pointed out not only diminishing returns with task complexity but also practical challenges in debugging and maintaining AI-generated code. PM2 and DEV4 emphasized transparency and technical debt issues, noting that AI-generated outputs often require significant human intervention for long-term viability. Additionally, while some experts like DEV9 viewed AI as effective for simpler or isolated coding tasks, concerns about technical debt and code smells remained significant in more complex applications.

4.1.3. Comparative Analysis between SMEs and LEs

Experts observed notable differences in AI's impact on software development between SMEs and LEs. Many experts (PM1, PM2, PM3, DEV1, DEV3, DEV6, DEV8) noted SMEs often faced challenges like limited financial resources, lack of infrastructure, and skilled personnel, hindering AI integration. Conversely, LEs benefited from strong infrastructure, dedicated R&D budgets, and scalability (PM2, DEV4, DEV7, DEV9).

SMEs' agility and faster decision-making offered potential advantages for adopting niche or innovative AI applications (PM2, DEV5, DEV11). However, DEV1 and DEV 3 emphasized their reliance on off-the-shelf tools, due to expertise gaps, limited customization.

Opinions diverged on whether cloud-based AI tools leveled the field. PM1 and DEV2 argued these tools lowered barriers for SMEs by reducing infrastructure needs, but DEV8 and DEV10 highlighted significant costs and strategic limitations. PM3 and DEV6 seen cultural resistance and organizational unreadiness as greater obstacles for SMEs compared to LEs.

4.1.4. Perspectives on Management Theories

Most experts emphasized adaptability as essential for responding to changing environments. DEV3 and DEV7 highlighted AI's role in creating flexible workflows and fostering innovation, particularly in their coding practices. However, DEV5 and DEV9 noted that organizational inertia and technical debt often limited adaptability. PM1 and PM2 emphasized managerial commitment to AI, while DEV4 and DEV10 highlighted the importance of employee-driven initiatives to foster organizational learning and agility.

Efficiency innovation dominated the discussions across interviews. Experts DEV6, DEV8, and DEV11 noted that AI tools optimized tasks in the SDLC. However, DEV1 and DEV3 cautioned that overemphasis on efficiency could risk neglecting long-term growth and the potential for more transformative applications. Sustaining innovation was evident in incremental improvements, with DEV5 and PM3 highlighting how AI aided error detection and enhanced code reliability. Yet, most experts argued that truly disruptive innovation had yet to be fully realized.

Experts broadly affirmed the importance of perceived usefulness and ease of use in AI adoption. DEV2 and DEV9 highlighted intuitive AI interfaces as critical. However, DEV3 and DEV8 emphasized the importance of trust in AI outputs, noting challenges related to compatibility and complexity. DEV7 stressed the role of iterative experimentation in fostering acceptance, while DEV4 pointed to skepticism among less tech-savvy teams, underscoring the challenges in bridging diverse organizational competencies.

4.2. Survey Results

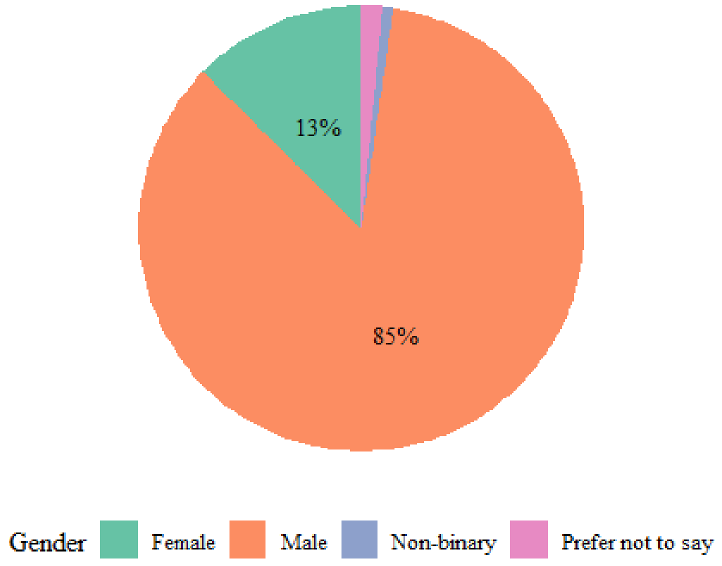
The following chapters present the results obtained from the survey.

4.2.1. Participant Overview

Figure 5 depicts the gender distribution of a population. Males constituted the majority at 85%, followed by females at 13%. Non-binary individuals and those who preferred not to disclose their gender represented minimal shares of the total.

Figure 5

Gender Distribution of Respondents

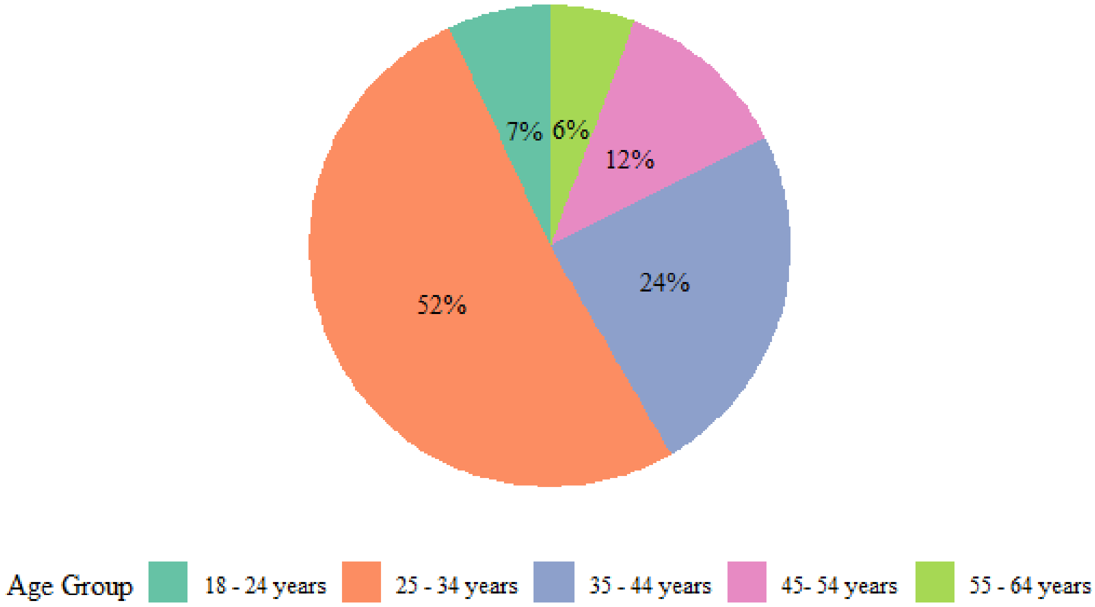


Note. Survey responses from n = 262 participants.

Figure 6 illustrates the distribution of age groups within a population. The largest segment, 25-34 years, constituted 52%, followed by 35-44 years at 24%. The 45-54 years group accounted for 12%, while the 18-24 years and 55-64 years groups represented 7% and 6%, respectively. The data highlights a predominantly younger demographic.

Figure 6

Age Distribution of Respondents



Note. Survey responses from n = 262 participants.

Most respondents fell within the Euro 40,000-59,999 range (42.4%), followed by Euro 60,000-79,999 (36.3%). Smaller proportions reported incomes of Euro 80,000-99,999 (12.2%), more than Euro 100,000 (4.2%), and less than Euro 40,000 (4.96%), indicating a concentration in mid-range income brackets.

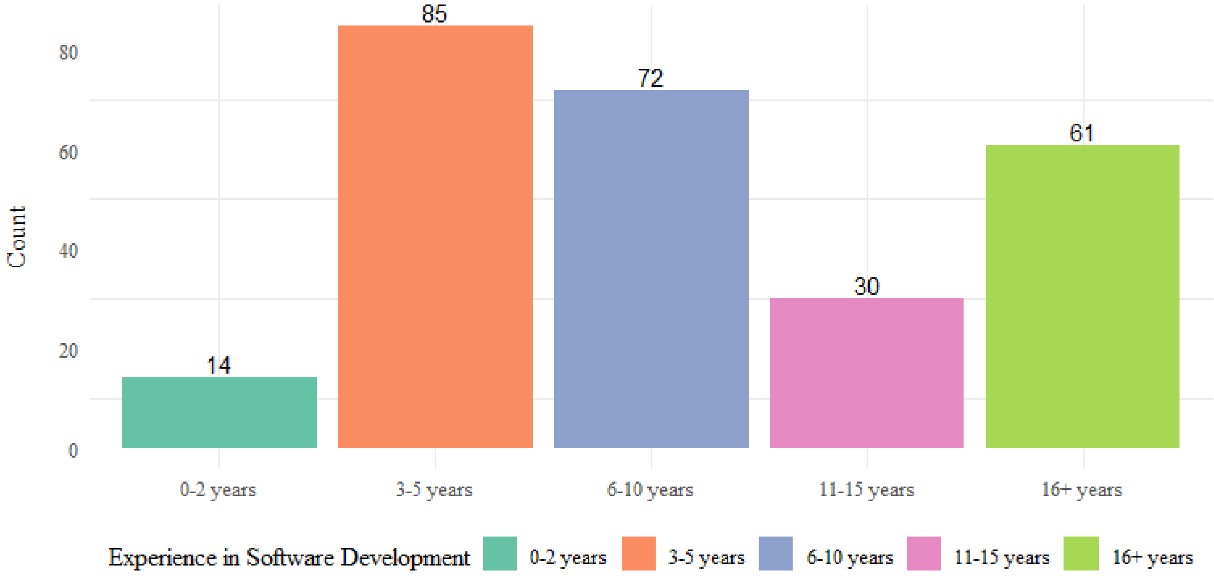
The largest group held a master's degree (43.1%), followed by those with a bachelor's degree (35.9%). Vocational training accounted for 11.8%, while 4.2% were high school graduates, and 3.44% had a doctoral degree. Only 1.53% possessed an entrance qualification for university, highlighting a predominance of higher education levels among respondents.

The largest proportion worked in companies with 501+ employees (27.9%), followed by 51-200 employees (27.1%). Companies with 11-50 employees and 201-500 employees each accounted for 19.8%, while the smallest segment was 1-10 employees (5.34%).

Figure 7 shows the distribution of respondents' experience in software development. The largest group had 3-5 years of experience (32.44%), followed by 6-10 years (27.48%) and 16+ years (23.28%). Respondents with 11-15 years and 0-2 years of experience constituted 11.45% and 5.34%, respectively, reflecting a predominance of mid-level experience in the sample.

Figure 7

Distribution of Experience in Software Development



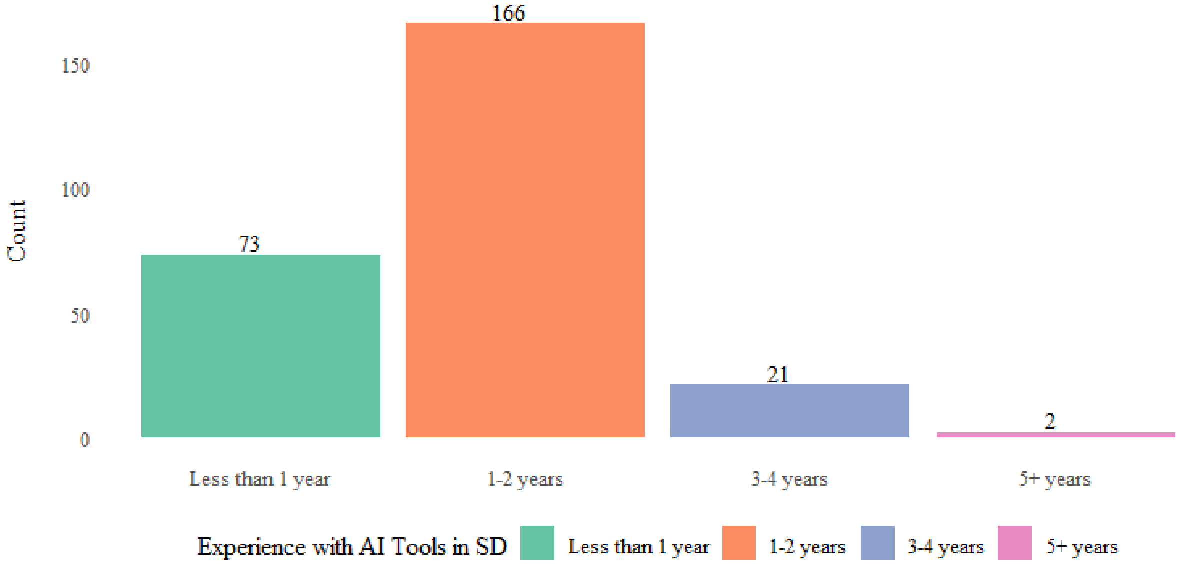
Note. Survey responses from n = 262 participants.

Software Developers constituted the largest group (240), followed by Software Architects (206) and Project Managers (64). Other roles, including QA Engineers, Business Analysts, and Product Owners, had significantly smaller counts, each below 40.

Figure 8 shows that most respondents had 1-2 years of experience with AI tools (64.1%), followed by less than 1 year (28.2%). Fewer had 3-4 years (8.1%) or 5+ years (0.8%) of experience, highlighting a concentration in early adoption stages.

Figure 8

Experience with AI Tools in Software Development



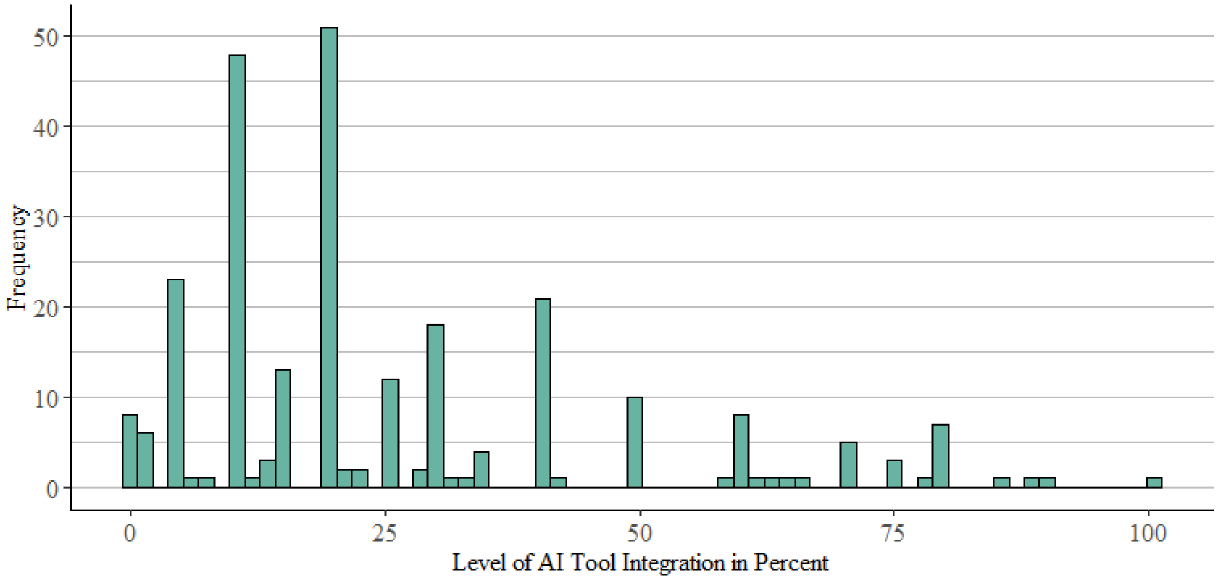
Note. Survey responses from n = 262 participants.

Furthermore, the majority identified as advanced users (51.5%), followed by intermediate users (30.5%). Novices constituted 9.16%, while experts and beginners accounted for 6.49% and 2.29%, respectively, indicating a strong concentration of advanced proficiency levels.

Figure 9 illustrates the distribution of AI tool integration levels among respondents, measured as a percentage. The data was heavily skewed toward lower levels of integration, with prominent peaks near 10% and 20%. Frequencies gradually declined as integration percentages increased, with sparse representation above 50%. This indicates a limited adoption of higher levels of AI integration.

Figure 9

Distribution of AI Tool Integration Levels in Software Development



Note. Survey responses from n = 262 participants.

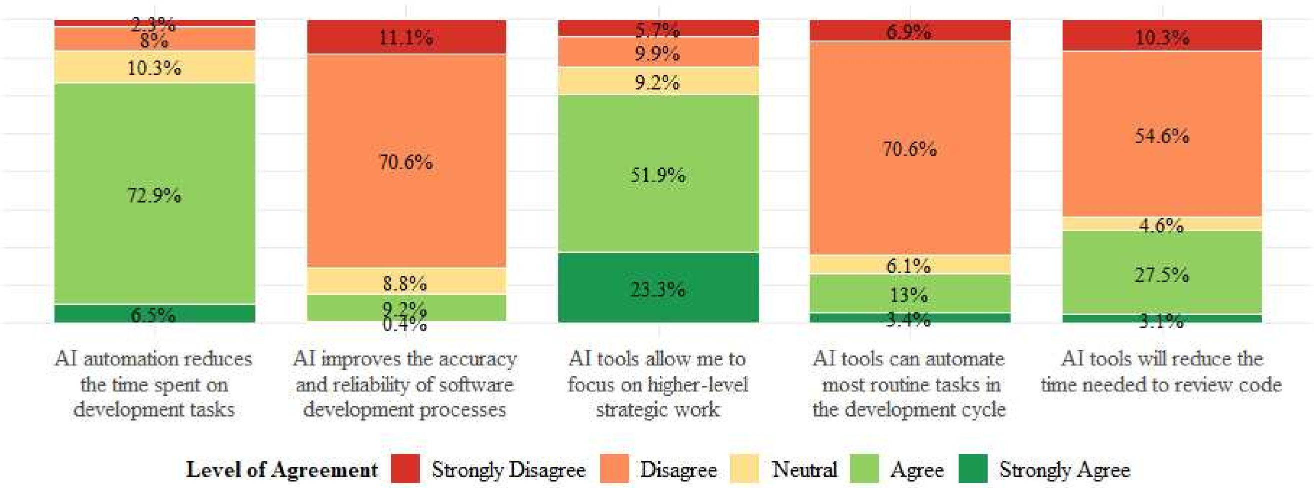
Additionally, most respondents used AI tools daily (61.1%), followed by weekly users (27.1%), while monthly, rare, and non-users represented 3.05%, 5.73%, and 3.05%, respectively, indicating predominantly frequent usage in software development.

4.2.2. Quantitative Analysis of Primary Factors

Figure 10 shows that, in terms of reducing time spent on development tasks, the majority (72.9%) agreed, with 6.5% strongly agreeing and only 2.3% strongly disagreeing. Regarding the improvement of accuracy and reliability in software development processes, 70.6% disagreed, with only 9.2% agreeing and 0.4% strongly agreeing, indicating significant skepticism. When assessing whether AI tools enabled developers to focus on higher-level strategic tasks, 51.9% agreed and 23.3% strongly agreed, demonstrating a positive reception overall. For automating routine tasks in the development cycle, 70.6% disagreed, reflecting doubt about AI's effectiveness in this area. Finally, on reducing the time needed for code review, 54.6% disagreed, while 27.5% agreed, showing mixed but predominantly skeptical views.

Figure 10

Perceived Efficiency and Automation Potential of AI Tools in Software Development

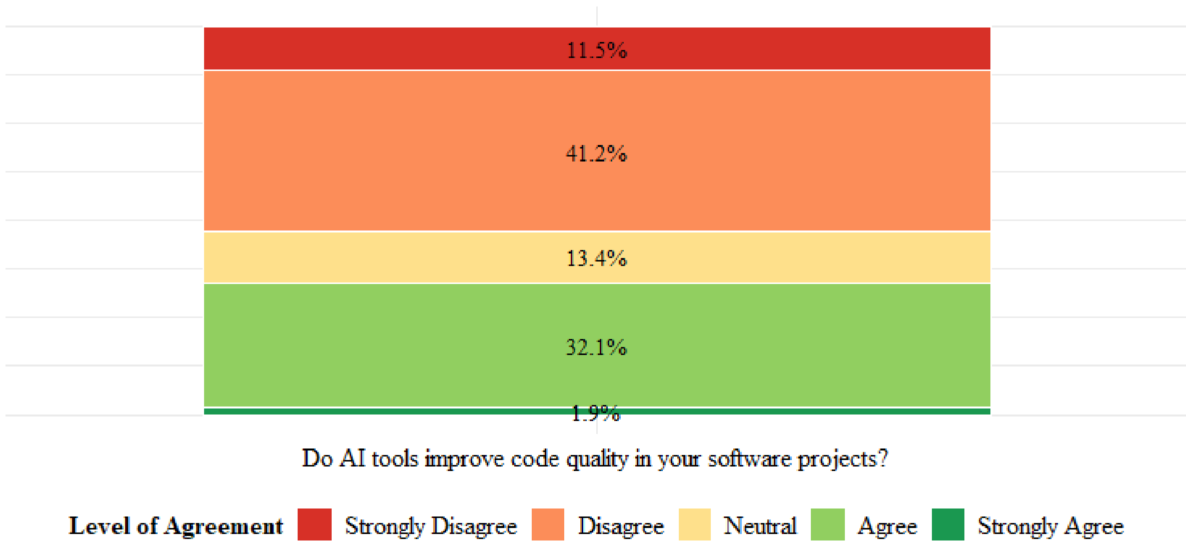


Note. Survey responses from n = 262 participants.

Figure 11 shows that 41.2% of respondents disagreed, with an additional 11.5% strongly disagreeing, indicating a significant level of skepticism about AI tools improving code quality. In contrast, 32.1% agreed and 1.9% strongly agreed, reflecting a minority who viewed AI as beneficial in this area. Meanwhile, 13.4% remained neutral, neither endorsing nor rejecting the idea, suggesting mixed but predominantly negative sentiments regarding AI's impact on code quality.

Figure 11

AI Tools' Impact on Code Quality in Software Development

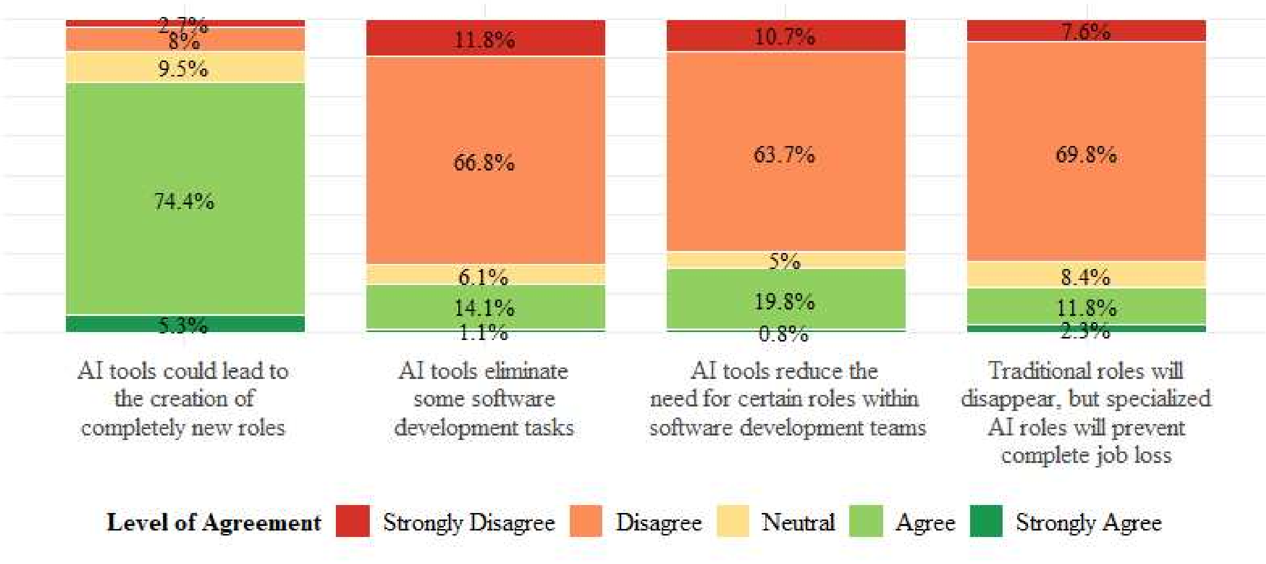


Note. Survey responses from n = 262 participants.

Figure 12 shows that, for creating new roles, 79.7% expressed optimism, while only 2.7% strongly disagreed. Most respondents (78.6%) were skeptical about AI eliminating tasks, with limited agreement (1.1%). Similarly, 77.4% doubted that AI would reduce roles in development teams, and 72.1% rejected the idea that AI roles would replace traditional ones, reflecting widespread skepticism overall.

Figure 12

AI Tools' Impact on Roles and Tasks in Software Development

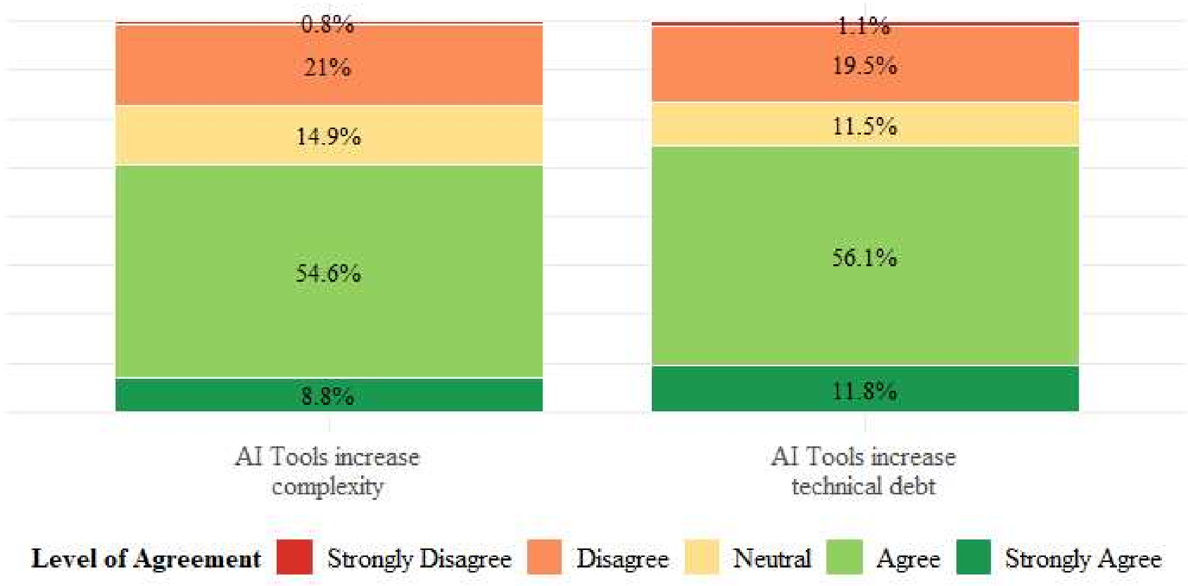


Note. Survey responses from n = 262 participants.

Figure 13 shows that, regarding the increase in complexity due to AI tools, 63.4% agreed, while 21.8% disagreed or strongly disagreed, and 14.9% remained neutral. For the increase in technical debt, 67.9% agreed, 20.6% disagreed or strongly disagreed, and 11.5% were neutral. These results indicate prevalent concerns about AI tools increasing both complexity and technical debt.

Figure 13

Perceptions of AI Tools Increasing Complexity and Technical Debt in Software Development



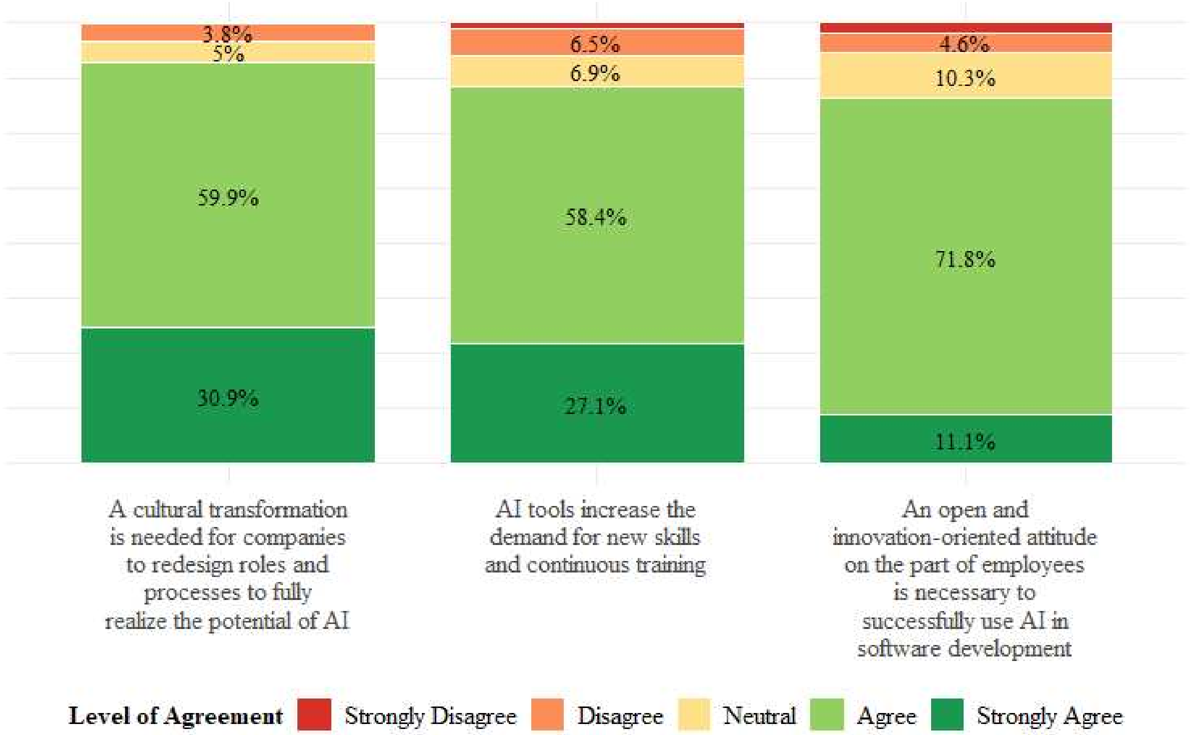
Note. Survey responses from n = 262 participants.

4.2.3. Quantitative Analysis of Secondary Factors

Figure 14 shows regarding the need for cultural transformation to redesign roles and processes, 59.9% agreed, and 30.9% strongly agreed, indicating widespread support. For the increased demand for new skills and continuous training, 58.4% agreed, and 27.1% strongly agreed, reflecting acknowledgment of skills adaptation needs. Lastly, the necessity of an open and innovation-oriented attitude among employees was highly supported, with 71.8% agreeing and 11.1% strongly agreeing, emphasizing the importance of mindset shifts for successful AI integration.

Figure 14

Organizational and Employee Readiness for AI Adoption in Software Development



Note. Survey responses from n = 262 participants.

A significant majority, 58.8%, agreed, and 36.3% strongly agreed, indicating that nearly 95% of respondents believed AI tools posed security risks. Only a small percentage were neutral (3.05%), while disagreement was minimal, with 1.53% disagreeing and just 0.38% strongly disagreeing. This highlighted a strong consensus around the perception of security risks associated with AI tools in software development.

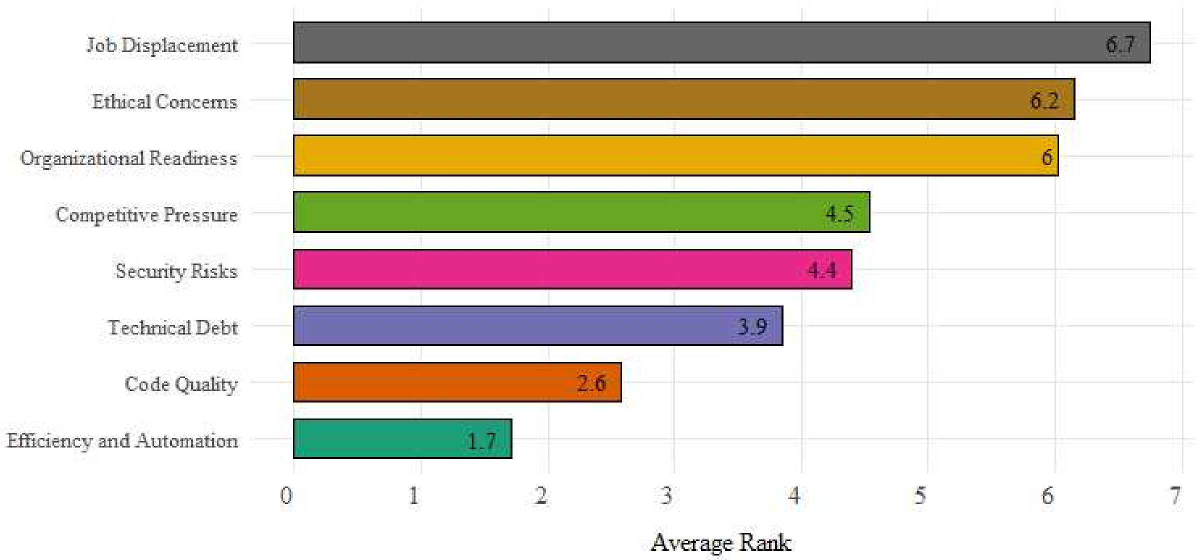
The sample reflected respondents' beliefs about whether AI tools in software development raised ethical concerns. A significant majority, 42% agreed and 18.3% strongly agreed, indicating that 60.3% expressed ethical concerns about AI's role. In contrast, 20.6% disagreed and 7.25% strongly disagreed, with 27.85% indicating skepticism or dismissal of ethical issues. Additionally, 11.8% remained neutral, suggesting a small portion of respondents were undecided. Overall, the majority perceived AI's integration into software development as ethically challenging.

The sample reflected perceptions of increasing competitive pressure in the software market due to AI adoption. A significant majority, 52.7% agreed and 23.3% strongly agreed, indicating widespread acknowledgment of rising competition. Meanwhile, 9.16% remained neutral, suggesting some ambivalence. A smaller proportion, 12.2%, disagreed and 2.67% strongly disagreed, highlighting minimal opposition to the idea. Overall, the results suggest that most participants perceived AI as a significant factor intensifying competition in the software development market.

Figure 15 shows "Efficiency and Automation" was rated the most significant (average rank: 1.7 of 8), followed by "Code Quality" (2.6), highlighting productivity and performance as top concerns. In contrast, "Job Displacement" (6.7) and "Ethical Concerns" (6.2) ranked as the least significant, suggesting lower perceived urgency. Mid-level factors included "Organizational Readiness" (6), "Competitive Pressure" (4.5), "Security Risks" (4.4), and "Technical Debt" (3.9), reflecting varying degrees of importance tied to organizational and technical challenges. The results emphasize a focus on immediate operational benefits over broader systemic issues.

Figure 15

Ranking of Factors Influencing AI Tools' Impact on Software Development



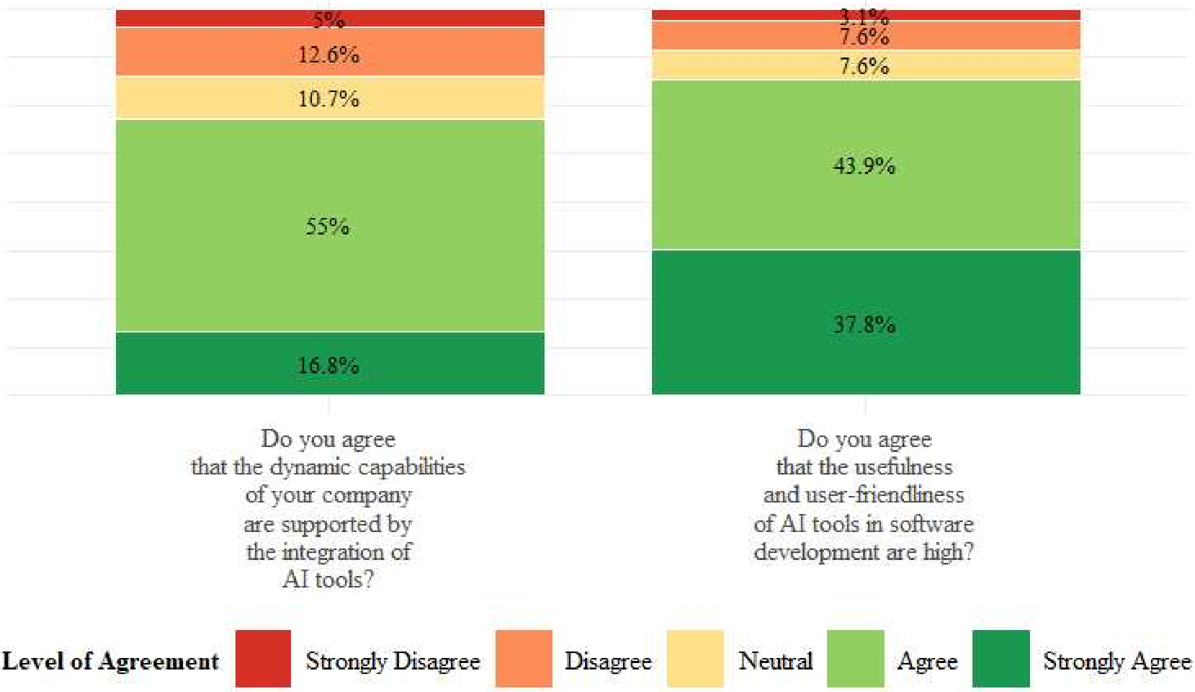
Note. Survey responses from n = 262 participants.

4.2.4. Perspectives on Management Theories

Figure 16 presents management perspectives on AI tools in software development. For their usefulness and user-friendliness, 43.9% agreed and 37.8% strongly agreed, showing broad support, while only 10.7% were neutral or negative. On whether AI integration supported company capabilities, 55% agreed and 16.8% strongly agreed, indicating significant affirmation, though with more neutral (10.7%) and negative (17.6%) responses compared to the first measure. Overall, there was strong support for AI tools, though concerns about organizational readiness persisted.

Figure 16

Management Perspectives on AI Tools in Software Development

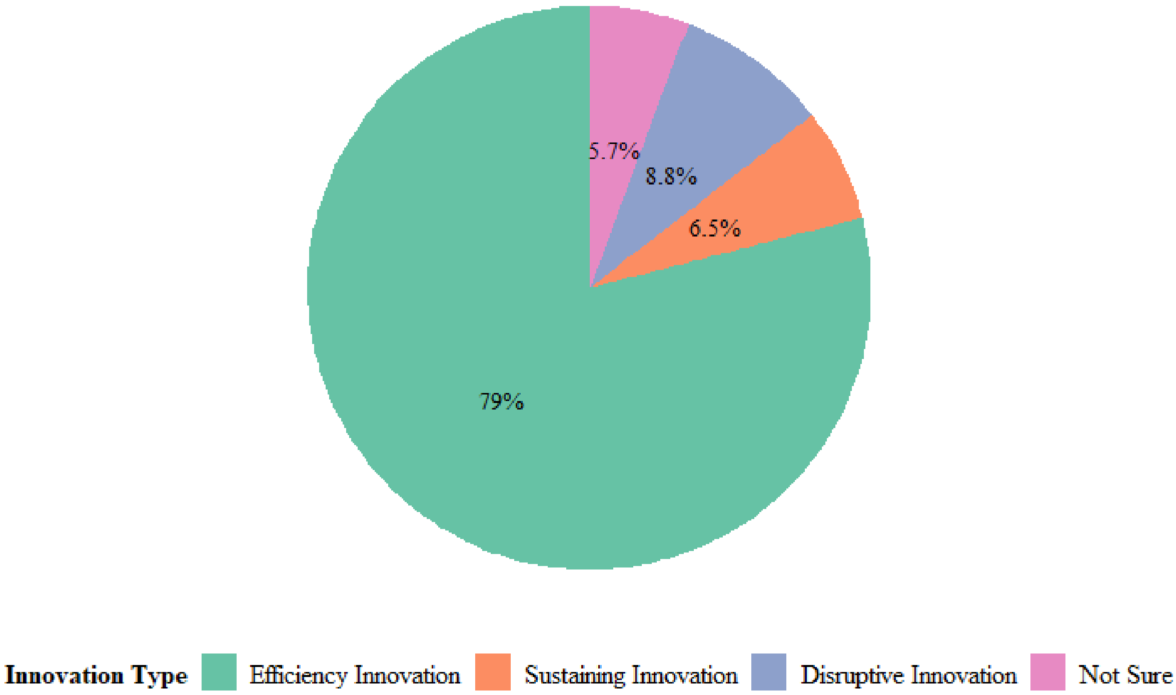


Note. Survey responses from n = 262 participants.

Figure 17 shows that 79% identified AI tools in software development as representing efficiency innovation, while 8.8% classified them as disruptive innovation and 6.5% as sustaining innovation. Only 5.7% were uncertain, indicating a strong perception of AI tools as an efficiency innovation.

Figure 17

Perspectives on Type of Innovation of AI Tools in Software Development



Note. Survey responses from n = 262 participants.

The sample revealed perceptions regarding whether AI tools would cause disruption in software development in the future. A significant majority (60.7%) agreed, and an additional 14.1% strongly agreed, reflecting widespread belief in the transformative potential of AI tools. Meanwhile, 13.7% disagreed and 1.91% strongly disagreed, indicating minimal skepticism. A smaller proportion (9.54%) remained neutral, suggesting some uncertainty or lack of a strong opinion. Overall, the data highlighted a dominant expectation of disruption driven by AI tools in software development.

4.2.5. Impact of AI Usage and Integration on Efficiency and Job Displacement

To further investigate the relationship between AI tools and respondents' perceptions, two ordinal logistic regression models were developed: the first examined the influence of AI usage frequency on perceptions of increased efficiency in development tasks, and the second explored the impact of AI integration level, with company size as an interaction term, on perceptions of reduced demand for developers. Ordinal logistic regression (OLR) was deemed appropriate for analyzing Likert-scale data as it accounts for the ordinal nature of responses and ensures the proportional odds assumption holds for ordered categorical variables (Olsson, 2022; Brant, 1990).

Figure 18

Kendall's Tau Rank Correlation Test

```
        Kendall's rank correlation tau

data:  efficiency_numeric and frequency_numeric
z = -8.3458, p-value < 2.2e-16
alternative hypothesis: true tau is not equal to 0
sample estimates:
       tau
-0.4715107
```

First, a Kendall's tau rank correlation test was performed to assess whether there was a relationship between the frequency of AI tool usage in software development tasks and respondents' perceptions of increased task efficiency due to AI. The test shown in Figure 18 revealed a significant negative association between AI usage frequency and perceived task efficiency indicating that higher frequency of AI usage corresponds to greater perceived efficiency improvements.

Building on this, the OLR analysis shown in Figure 19 further explored this relationship. It revealed significant inverse associations between the frequency of AI usage and the likelihood of perceiving improvements in task efficiency.

Figure 19

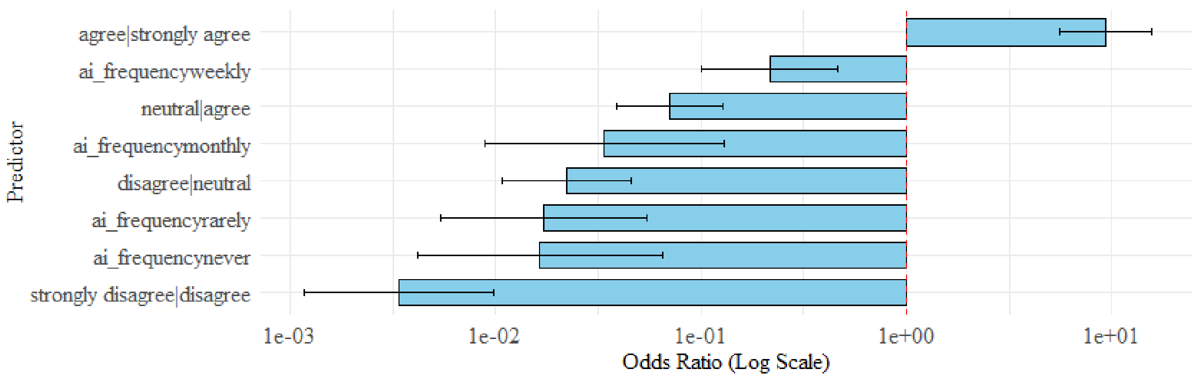
Summary of Regression Results AI Usage Frequency on Perceived Efficiency Gains

	Predictor	OddsRatio	LowerCI	UpperCI	pValue
ai_frequencyweekly	ai_frequencyweekly	0.215924870	0.101183401	0.460782590	7.385063e-05
ai_frequencymonthly	ai_frequencymonthly	0.033920913	0.008885645	0.129492946	7.389972e-07
ai_frequencyrarely	ai_frequencyrarely	0.017126846	0.005365289	0.054671586	6.516118e-12
ai_frequencynever	ai_frequencynever	0.016430055	0.004149340	0.065057751	4.864740e-09
strongly disagree|disagree	strongly disagree|disagree	0.003359738	0.001162510	0.009709883	7.038608e-26
disagree|neutral	disagree|neutral	0.022019787	0.010691385	0.045351564	4.119615e-25
neutral|agree	neutral|agree	0.070457893	0.038854233	0.127767664	2.425219e-18
agree|strongly agree	agree|strongly agree	9.352492438	5.629440510	15.537798941	6.031585e-18

Respondents who used AI tools less frequently, such as weekly, monthly, rarely, or never, were significantly less likely to report that AI tools enhanced task efficiency compared to those who used AI tools daily. Odds ratios for the categories "weekly," "monthly," "rarely," and "never" compared to "daily" usage were 0.22, 0.03, 0.02, and 0.02, respectively, highlighting that more frequent use of AI tools is associated with a higher likelihood of agreeing to perceiving increased efficiency. For example, respondents who used AI tools weekly were significantly less likely to agree to perceived task efficiency improvements compared to daily users, with an odds ratio of 0.22, indicating they were 78% less likely to report perceived efficiency gains. The Wald test produced highly significant p-values for all frequency levels ($p<0.001$), reinforcing the robustness of these findings. Additionally, the proportional odds assumption was upheld ($p>0.80$), confirming the validity of the model using the Brant test. These results underscore the importance of consistent AI tool usage in improving perceived task efficiency, with diminishing returns for less frequent users. These findings are illustrated in Figure 20, where the odds ratios on a logarithmic scale highlight the steep decline in perceived task efficiency gains with decreasing AI tool usage frequency.

Figure 20

Odds Ratios of Predictors on Perceived Task Efficiency Gains



The second OLR model assessed whether company size moderates the relationship between the level of AI integration in software development tasks and respondents' perceptions of reduced demand for developers. The model incorporated an interaction term between AI integration and company size to evaluate this potential moderating effect.

Figure 21

Odds Ratios of Predictors on Perceptions of Reduced Developer Demand

	Predictor	Odds Ratio	Lower CI	Upper CI	p-value
ai_integration	ai_integration	1.0282	1.0114	1.0453	0.0010
companysize200+ employees	companysize200+ employees	1.3567	0.6075	3.0301	0.4568
ai_integration:companysize200+ employees	ai_integration:companysize200+ employees	1.0027	0.9802	1.0257	0.8183

As shown in Figure 21, the analysis revealed that AI integration levels were significantly associated with perceptions of reduced demand for developers, with an odds ratio of 1.03 (p=0.001), indicating that higher AI integration slightly increases the likelihood of agreeing that AI reduces developer demand. However, the main effect of company size (p=0.457) and the interaction between AI integration and company size (p=0.818) were not statistically significant, suggesting that perceptions of job displacement are largely unaffected by company size or its interaction with AI integration. The proportional odds assumption was validated (p =0.24) as confirmed by the Brant test, supporting the model's robustness. These results highlight that while AI integration levels have a modest but significant influence on perceptions of developer demand, company size does not play a moderating role in this relationship.

5. Conclusion

The final chapter summarizes the findings, addresses limitations, gives theoretical and practical implications, and outlines directions for future research.

5.1. Main Findings

The findings from previous chapters, triangulating literature, expert interviews, and survey data, provide the basis for assessing the disruptive potential of AI tools in Germany's software development sector.

5.1.1. Adoption Trends of AI in Software Development

Survey data highlights that the integration of AI tools among German SMEs is at an early stage, with usage primarily focused on routine tasks like debugging and code refactoring, while strategic adoption remains limited (Hicks, Lee, & Foster-Marks, 2023; Rajbhoj et al., 2024). This was corroborated by expert interviews, which revealed that AI tools such as ChatGPT are effective in optimizing documentation and testing processes but are seldom integrated into strategic software development workflows (Daigle, 2024; Weber et al., 2024). Supporting this, the literature emphasizes AI's potential to enhance productivity and reduce cognitive load, though adoption is hindered by organizational readiness and infrastructure challenges (Russo, 2024; Saleem et al., 2023).

SMEs face distinct challenges compared to LEs, as highlighted in both expert interviews and survey findings. Experts noted that SMEs often operate with constrained resources and limited technical expertise, which hampers their ability to adopt and integrate custom AI solutions effectively (Holmström & Carroll, 2024; Kanbur et al., 2023). However, SMEs were found to leverage their agility for faster implementation of off-the-shelf tools, offering niche innovations (Chui et al., 2023). Conversely, LEs benefit from robust infrastructure, dedicated R&D budgets, and strategic planning capabilities, enabling them to deploy AI at scale, a point echoed in the literature (Teece et al., 1997; Christensen et al., 2015).

While the survey pointed to skepticism about AI’s ability to enhance strategic decision-making, expert interviews revealed optimism regarding its potential to democratize coding and innovation (Hicks, Lee, & Foster-Marks, 2023). The literature emphasized AI’s transformative capacity but highlighted systemic barriers such as ethical concerns and technical debt that disproportionately affect SMEs (Yetiştiren et al., 2023; Ebert & Louridas, 2023). Claims of AI bridging skill gaps and democratizing software development were supported by experts but not universally acknowledged in survey data, indicating this optimism is not yet reflected in widespread adoption patterns (Filippi et al., 2023; Johnson & Smith, 2021).

5.1.2. Drivers of Adoption

The adoption of AI in software development is primarily driven by efficiency gains and competitive pressures, supported by insights from literature, expert interviews, survey data, and regression analysis. AI tools such as ChatGPT and GitHub Copilot are consistently recognized for their capacity to automate routine coding tasks, debugging, and documentation (Weber et al., 2024; Rajbhoj et al., 2024). Survey results reveal that 72.9% of respondents identify efficiency improvements as a critical motivator for adoption, further supported by its top ranking in both the survey and expert assessments. The regression analysis revealed that consistent usage of AI tools is critical for improving perceived task efficiency, with diminishing returns observed for less frequent users. Experts emphasized these tools' ability to streamline workflows, reduce cognitive load, and enable developers to focus on strategic, high-value tasks (Daigle, 2024). The literature further aligns, highlighting that AI minimizes manual coding and enhances productivity by accelerating development cycles and improving operational accuracy (Ebert & Louridas, 2023).

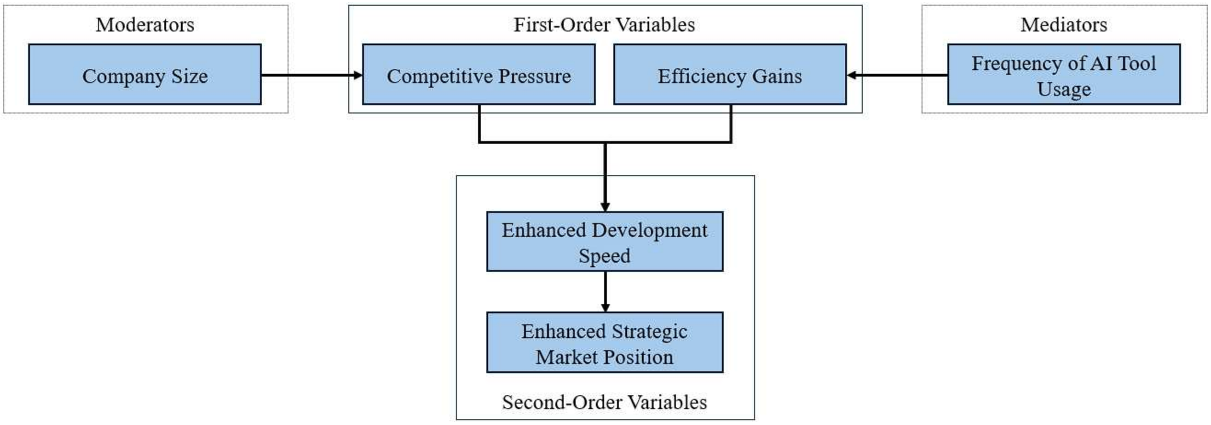
Competitive pressure emerged as another key driver of AI adoption, with survey data indicating that respondents view AI as essential for maintaining relevance in a rapidly evolving market. Expert interviews highlighted AI's strategic necessity, particularly for SMEs, which leverage it to overcome resource constraints and compete with larger enterprises (Chui et al., 2023). Larger organizations, meanwhile, adopt AI to enhance scalability, improve speed, and sustain market leadership by delivering higher-quality solutions (Bhavsar et al., 2019). The literature further explored AI's democratizing potential, enabling smaller firms to access advanced development capabilities and compete effectively within competitive markets (Schneider et al., 2024).

The triangulated insights consistently validate efficiency gains and competitive pressures as primary drivers of AI adoption. Regression findings provided quantitative weight to these dynamics, substantiating the direct impact of AI integration on productivity and market positioning. Collectively, these findings underscored the critical role of AI in enhancing productivity, tailoring adoption strategies to company size and usage frequency, and sustaining competitiveness in the software development sector (Christensen & Raynor, 2003; Teece et al., 1997).

A framework was developed, as shown in Figure 22. A first-order variable is a construct measured directly by its observed indicators, while a second-order variable is a higher-level construct measured indirectly through its first-order variables (Polites et al., 2012). Moderators influence the strength or direction of an independent variable's effect on a dependent variable, while mediators explain the process linking the two (Baron & Kenny, 1986).

Figure 22

Framework of AI Adoption Drivers in Software Development



5.1.3. Impediments to AI Adoption

Job displacement posed a challenge to AI adoption in software development, emerging as a transitional risk rather than an absolute barrier. Literature, expert opinions, and survey data converged on the notion that AI reshapes roles, replacing routine tasks with new positions like AI system managers and strategists while preserving human-centric functions such as debugging (Babashahi et al., 2024; Meesters et al., 2022). However, views on the extent of displacement diverged as some experts emphasize role resilience (Filippi et al., 2023), while others and long-term AI users acknowledge restructuring risks (Naveen et al., 2023). Survey data reinforces skepticism about AI eliminating jobs (72.1% disagreement) while highlighting optimism for role creation (79.7% agreement). Although regressions revealed higher integration levels of AI tools in software development are associated with greater agreement that AI tools decrease demand for developers, company size and its interaction with AI integration do not significantly influence these perceptions, suggesting a consistent impact across companies of varying sizes. Adaptability and skill transformation, particularly in higher-order cognitive abilities and multidisciplinary expertise, are consistently identified as essential to mitigating displacement risks (Chen and Tajdini, 2024; Babashahi et al., 2024).

The adoption of AI tools in software development faced challenges in realizing their full potential for improving code quality. Literature and expert opinions consistently highlighted AI's ability to produce cleaner, more consistent, and maintainable code, enhancing readability and adherence to conventions (Dantas et al., 2023; Liu et al., 2024). However, limitations emerged with complex tasks, where semantic inaccuracies, debugging difficulties, and the accumulation of technical debt necessitate substantial human oversight (Ebert and Louridas, 2023). Experts further critiqued the opacity of AI-generated outputs, complicating long-term maintenance (Yetiştiren et al., 2023). Survey data reinforced these concerns, as over half of respondents' express skepticism regarding AI's impact on code quality. Notably, consistent and experienced use of AI tools correlated with reduced technical debt concerns and improved efficiency, underscoring the importance of robust workflows and expertise to address these limitations effectively.

Technical debt significantly impeded the adoption of AI tools in software development by increasing cognitive complexity, complicating long-term maintenance, and introducing scalability and security challenges. Literature, expert insights, and survey results converged on the need for continuous human oversight to mitigate risks stemming from poorly documented, inflexible, or improperly validated AI-generated code (Jaspan & Green, 2023; Russo, 2024). However, divergences highlight the role of consistent AI integration and organizational context, with regular usage and smaller, agile teams reporting reduced concerns about technical debt (Ebert and Louridas, 2023). While AI boosts efficiency and productivity, its inability to address non-standard requirements and its lack of explainability exacerbated maintenance challenges (Yetiştiren et al., 2023).

Security risks were a significant impediment to the adoption of AI tools in software development, as vulnerabilities such as data leakage, adversarial prompting, and biases in training data undermined trust in AI-driven systems. Literature and expert opinions emphasized the complexity and opacity of AI-generated outputs, necessitating governance frameworks and human oversight to address these risks (Chui et al., 2023; Russo, 2024). Survey results indicated a strong consensus, with nearly 95% of respondents agreeing that AI tools introduced security risks, reflecting widespread recognition of these challenges. Advanced proficiency and organizational readiness mitigate some concerns, while inconsistent usage amplifies vulnerabilities (Humphreys et al., 2024; Nguyen-Duc et al., 2023).

Ethical concerns posed a significant barrier to the adoption of AI tools in software development, with literature and expert opinions highlighting critical issues such as privacy, intellectual property, and bias (Thomson and Schmoldt, 2001; Gogoll et al., 2021). Robust governance frameworks, including internal ethics roles and external regulation, are widely advocated to address these challenges (Al-Kfairy et al., 2024). Privacy risks and ambiguities in ownership of AI-generated content underscored the need for clear policies (Baddi et al., 2024), while systemic biases raise fairness concerns that some experts argue require more than technical solutions (Johnson and Smith, 2021). Survey findings indicated that ethical concerns are broadly recognized, with 60.3% of respondents acknowledging them, reflecting a general apprehension about AI's ethical implications.

5.1.4. Organizational Dynamics

Organizational readiness is a critical determinant of AI adoption in software development, encompassing workforce upskilling, cultural transformation, and role redefinition. Saleem et al. (2023) and Russo (2024) emphasize the necessity of structured training programs to address skill gaps, particularly for non-technical roles, and to foster an innovation-oriented mindset essential for AI integration. Survey results revealed strong support for cultural transformation (82.9%) and continuous training (85.5%), highlighting widespread recognition of these needs. Developers often adapt independently to AI tools, but non-technical staff face greater challenges, necessitating targeted interventions (Holmström and Carroll, 2024). Larger organizations reported higher readiness, reflecting the role of resources and scale in mitigating adoption barriers (Russo, 2024).

5.1.5. Disruptiveness of AI Tools in German SME Software Development Sector

Applying disruptive innovation theory to AI tools in the German SME software development sector highlighted that their impact is predominantly aligned with efficiency innovations rather than disruptive ones. Christensen, Raynor, and McDonald (2015) argued that disruptive innovations emerge in underserved markets, initially underperforming established solutions but gradually reshaping industries. In contrast, survey respondents and experts identified AI tools primarily as efficiency innovations, enhancing workflows through task automation and improved code quality (Weber et al., 2024; Daigle, 2024). Expert interviews and survey data consistently emphasized that these tools sustain competitiveness by optimizing established practices without displacing traditional methodologies or creating new market segments

(Christensen & van Bever, 2014). Although some experts and scholars such as Li (2015) point to the potential for long-term disruption as AI becomes more accessible and integrated, the current consensus, supported by triangulated findings, was clear: AI tools act as refinements to existing processes, fostering incremental improvements rather than driving transformative change in the SME software development sector.

5.1.6. Theoretical and Practical Implications

The adoption of AI tools in software development reflects a complex interplay between theoretical frameworks and practical realities, balancing opportunities and constraints. The dynamic capabilities framework (Teece, 1997; Eisenhardt & Martin, 2000) highlights adaptability and resource reconfiguration, yet challenges like organizational inertia and technical debt impede progress (Collis & Anand, 2019). The results located AI within the construct of efficiency innovation articulated by Christensen et al. (2015) and emphasized the automation of tasks and short-term gains, similar to the literature and expert interviews, which show only limited disruptive innovations. Barreto (2010) emphasized multidimensional capabilities, including sensing opportunities and mitigating threats, aligning with findings that SMEs face technical resource and readiness limitations.

The integration of TAM and DOI frameworks (Davis, 1985; Rogers, 1983) underscored the role of perceived usefulness and compatibility in adoption, but real-world insights highlight gaps in trust and systemic readiness, particularly in SMEs. Rapid technological change, talent shortages, and upskilling needs exacerbate barriers, with the structural talent gap in SMEs reflecting disparities compared to larger enterprises. While upskilling aligns with continuous learning (Russo, 2024; Hicks et al., 2023), Barreto's (2010) emphasis on timely decision-making and resource flexibility stresses the need for workforce agility. Ethical and security concerns, including transparency and bias, further complicate adoption (Al-Kfairy et al., 2024; Chui et al., 2023). These findings reveal a misalignment between AI's theoretical potential and practical adoption, highlighting the need for strategies to mitigate technical debt, enhance readiness, and fully realize AI's transformative impact.

5.2. Limitations

5.2.1. Survey

The survey's limitations revealed methodological constraints affecting the generalizability and validity of its findings. Only 262 of 375 respondents (69.87%) fully completed the survey, raising concerns about nonresponse bias. The sample was largely WEIRD (Western, educated, industrialized, rich, and democratic), a demographically distinct group (Henrich et al., 2010). Homogeneity in age (52% aged 25 - 34) and education (43.1% with master's degrees) limited insights into how diverse demographics might influence attitudes toward AI in software development, potentially skewing results and reducing applicability to broader populations. The use of Likert scales, prone to acquiescence bias, may have inflated perceptions of AI's effectiveness (Krosnick, 1999). Although quantitative techniques aimed to reduce bias, reliance on qualitative data adds subjectivity, potentially compromising objectivity.

5.2.2. Expert Interviews

The study faced several limitations in its semi-structured interviews. The inductively developed interview framework, informed by literature, provided focus but may have introduced bias through the researcher's judgment. Participant selection, constrained by time and resources, prioritized availability over diversity, while recruitment via personal networks risked selection bias. Participants, though consistently highly experienced, varied in professional roles; some were not coding actively but contributed valuable insights from software-related positions. Differences in engagement with AI tools, especially in professional contexts, might have influenced the depth of insights. Variability in interview modes (virtual, face-to-face, email) affected response depth, with email interviews limiting probing opportunities. Conducting interviews in German and translating transcripts to English risked losing nuance in translation. The structured deductive coding approach may have missed emergent themes, and the rapid evolution of AI tools limits the findings' temporal relevance and generalizability.

5.3. Future Research

Future research on the adoption of AI tools in software development, particularly in Germany's SME sector, is critical to addressing gaps and challenges identified in this study. Future research should explore the interplay between organizational readiness and technical innovation, focusing on how SMEs can integrate AI sustainably with limited resources. AI-as-a-Service could level the playing field between SMEs and large enterprises, warranting investigation. Ethical implications, including data privacy, bias mitigation, and intellectual property, warrant granular analysis, given their potential to shape adoption trajectories. AI's impacts on workforce dynamics and job roles merit exploration to understand skill transformations and hybrid roles. Comparative studies across industries or geographic contexts could reveal how organizational ecosystems influence AI adoption and disruptions, enabling tailored strategies for sustainable integration.

6. Bibliography

Alcor. (n.d.). 10 key roles in a software development team: Who is responsible for what? Alcor. Retrieved from https://alcor-bpo.com/10-key-roles-in-a-software-development-team-who-is-responsible-for-what/

Ali, S. J., Christin, A., Smart, A., & Katila, R. (2023). Walking the walk of AI ethics in technology companies. HAI Policy & Society, 1–10. https://doi.org/10.1145/3593013.3593990

Al-Kfairy, M., Mustafa, D., Kshetri, N., Insiew, M., & Alfandi, O. (2024). Ethical challenges and solutions of generative AI: An interdisciplinary perspective. Informatics, 11(58), 1-29. https://doi.org/10.3390/informatics11030058

Al-Saqqa, S., Sawalha, S., & AbdelNabi, H. (2020). Agile software development: Methodologies and trends. International Journal of Interactive Mobile Technologies, 14(11), 244-270. https://doi.org/10.3991/ijim.v14i11.13269

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., ... Stoica, I. (2010). A view of cloud computing. Communications of the ACM, 53(4), 50–58. https://doi.org/10.1145/1721654.1721672

Armyanova, M., & Aleksandrova, Y. (2023). Using artificial intelligence in software development. Notices of the Union of Scientists - Varna, Economic Sciences Series, 12(1), 167-174.

Babashahi, L., Barbosa, C. E., Lima, Y., Lyra, A., Salazar, H., Argôlo, M. A., & de Souza, J. M. (2024). AI in the workplace: A systematic review of skill transformation in the industry. Administrative Sciences, 14(127). https://doi.org/10.3390/admsci14060127

Baddi, Y., Almaiah, M. A., Almomani, O., & Maleh, Y. (Eds.). (2024). The art of cyber defense: From risk assessment to threat intelligence (1st ed.). CRC Press. https://doi.org/10.1201/9781032714806

Bader, J., Kim, S. S., Luan, F. S., Chandra, S., & Meijer, E. (2021). AI in software engineering at Facebook. IEEE Software, 58-61. https://doi.org/10.1109/MS.2021.3061664

Bain & UiPath. (2023). AI and automation accelerating rapid, large-scale business change. UiPath. Retrieved from https://www.uipath.com/newsroom/ai-and-automation-accelerating-business-change-finds-new-bain-uipath-report

Balaji, S., & Murugaiyan, M. S. (2012). Waterfall vs. V-Model vs. Agile: A comparative study on SDLC. International Journal of Information Technology and Business Management, 2(1), 26-30.

Bano, M., Hoda, R., Zowghi, D. et al. (2024). Large language models for qualitative research in software engineering: Exploring opportunities and challenges. Automated Software Engineering, 31(8). https://doi.org/10.1007/s10515-023-00407-8

Baron, R. M., & Kenny, D. A. (1986). The moderator-mediator variable distinction in social psychological research: Conceptual, strategic, and statistical considerations. *Journal of Personality and Social Psychology, 51*(6), 1173–1182. https://doi.org/10.1037/0022-3514.51.6.1173

Barreto, I. (2010). Dynamic capabilities: A review of past research and an agenda for the future. Journal of Management, 36(1), 256–280. https://doi.org/10.1177/0149206309350776

Bartlett, J. E., Kotrlik, J. W., & Higgins, C. C. (2001). Organizational research: Determining appropriate sample size in survey research. *Information technology, learning, and per- formance journal*, *19*(1), 43–50.

Bayer, F. (n.d.). ITK-Arbeitsmarkt und -Unternehmen. Bitkom e. V. Retrieved October 1, 2024, from https://www.bitkom.org/Themen/Marktdaten/ITK-Konjunktur/Anzahl-ITK-Unternehmen

Bayer, F. (n.d.). ITK-Märkte. Bitkom e.V. Retrieved October 1, 2024, from https://www.bitkom.org/Marktdaten/ITK-Konjunktur/ITK-Markt-Deutschland

Bhavsar, K., Shah, V., & Gopalan, S. (2019). Business Process Reengineering: A Scope of Automation in Software Project Management using Artificial Intelligence. International Journal of Engineering and Advanced Technology, 9(2), 3589-3595.

Bitkom. (2023). Rekord-Fachkräftemangel: In Deutschland sind 149.000 IT-Jobs unbesetzt. Bitkom e. V. Retrieved October 1, 2024, from https://www.bitkom.org/Presse/Presseinformation/Rekord-Fachkraeftemangel-Deutschland-IT-Jobs-unbesetzt

Bitkom. (2024). IT-Fachkräfte 2040: Wo steht die deutsche Wirtschaft? Bitkom e. V. Retrieved October 1, 2024, from https://www.bitkom.org/sites/main/files/2024-04/240411Bitkom-Charts-IT-Fachkraftemangel-2040final.pdf

Brant, R. (1990). Assessing proportionality in the proportional odds model for ordinal logistic regression. Biometrics, 46(4), 1171–1178. https://doi.org/10.2307/2532457

Brooks, F. P. Jr. (1986). No silver bullet: Essence and accident in software engineering. Proceedings of the IFIP Tenth World Computing Conference. Reprinted in The Mythical Man-Month: Essays on Software Engineering (Anniversary ed.). Addison-Wesley.

Bryman, A., & Bell, E. (2011). *Business research methods* (3rd ed.). Oxford University Press.

Chen, J. (2024). The use of AI in software engineering: A synthetic knowledge synthesis of the recent research literature. Springer.

Chen, J., & Tajdini, S. (2024). A moderated model of artificial intelligence adoption in firms and its effects on their performance. Information Technology and Management. https://doi.org/10.1007/s10799-024-00422-5

Christensen, C. M., & Raynor, M. E. (2003). The innovator's solution: Creating and sustaining successful growth. Harvard Business School Press.

Christensen, C. M., & van Bever, D. (2014). The capitalist's dilemma. *Harvard Business Review*. Retrieved from https://hbr.org/2014/06/the-capitalists-dilemma

Christensen, C. M., Raynor, M., & McDonald, R. (2015). What is disruptive innovation? Harvard Business Review, (December 2015), 44–53.

Chui, M., Yee, L., Hall, B., Singla, A., & Sukharevsky, A. (2023). The state of AI in 2023: Generative AI's breakout year. McKinsey & Company. Retrieved from

Cocco, L., Mannaro, K., Concas, G., & Marchesi, M. (2011). Simulating Kanban and Scrum vs. Waterfall with system dynamics. In Agile Processes in Software Engineering and Extreme Programming: 12th International Conference, XP 2011, Madrid, Spain, May 10-13, 2011. Proceedings 12 (pp. 117-131). Springer Berlin Heidelberg.

Collins, B. (2024, February 26). *Nvidia CEO predicts the death of coding — Jensen Huang says AI will do the work, so kids don't need to learn*. TechRadar. https://www.techradar.com/pro/nvidia-ceo-predicts-the-death-of-coding-jensen-huang-says-ai-will-do-the-work-so-kids-dont-need-to-learn

Collis, D. J., & Anand, B. N. (2019). The limitations of dynamic capabilities. Harvard Business School.

Coutinho, M., Marques, L., Santos, A., Dahia, M., França, C., & de Souza Santos, R. (2024, July). The role of generative AI in software development productivity: A pilot case study. In Proceedings of the 1st ACM International Conference on AI-Powered Software (pp. 131-138). arXiv. https://arxiv.org/abs/2406.00560

Crevier, D. (1993). Expert systems as design aids for artificial vision systems: A survey. In Intelligent Robots and Computer Vision XII: Algorithms and Techniques (Proc. SPIE, Vol. 2055). SPIE. https://doi.org/10.1117/12.150131

Cusumano, M. A. (2010). Cloud computing and SaaS as new computing platforms. Communications of the ACM, 53(4), 27–29. https://doi.org/10.1145/1721654.1721667

Daigle, K. (2024, September 3). Survey: The AI wave continues to grow on software development teams. GitHub. https://github.blog/news-insights/research/survey-ai-wave-grows/

Dantas, C. E. C., Rocha, A. M., & Maia, M. A. (2023). Assessing the readability of ChatGPT code snippet recommendations: A comparative study. In Proceedings of the XXXVII Brazilian Symposium on Software Engineering (SBES 2023) (pp. 283-292). ACM. https://doi.org/10.1145/3613372.3613413

Davis, F. D. (1985). A technology acceptance model for empirically testing new end-user information systems: Theory and results. Massachusetts Institute of Technology.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (pp. 4171–4186). https://doi.org/10.18653/v1/N19-1423

Ebert, C., & Louridas, P. (2023). Generative AI for software practitioners. IEEE Software, 40(4), 30-37. https://doi.org/10.1109/MS.2023.3265877

Eisenhardt, K. M., & Martin, J. A. (2000). Dynamic capabilities: What are they? Strategic Management Journal, 21(10/11), 1105-1121. https://doi.org/10.1002/smj.

Filippi, E., Bannò, M., & Trento, S. (2023). Automation technologies and their impact on employment: A review, synthesis and future research agenda. Technological Forecasting and Social Change, 191, 122448. https://doi.org/10.1016/j.techfore.2023.122448

Fowler, M., & Beck, K. (2019). *Refactoring: Improving the design of existing code*. Addison-Wesley.

Germany Trade & Invest. (n.d.). Software in the digital economy. GTAI. Retrieved October 1, 2024, from https://www.gtai.de/en/invest/industries/digital-economy/software#75634

Goertzel, B. (2014). Artificial general intelligence: Concept, state of the art, and future prospects. Journal of Artificial General Intelligence, 5(1), 1–48. https://doi.org/10.2478/jagi-2014-0001

Gogoll, J., Zuber, N., Kacianka, S., Greger, T., Pretschner, A., & Nida-Rümelin, J. (2021). Ethics in the software development process: From codes of conduct to ethical deliberation. Philosophy & Technology, 34(3), 1085-1108. https://doi.org/10.1007/s13347-021-00451-w

Google Cloud. (n.d.). What are AI applications? Cloud.google.com. Retrieved October 1, 2024, from https://cloud.google.com/discover/ai-applications?hl=en

Guest, G., Bunce, A., & Johnson, L. (2006). How many interviews are enough? an experiment with data saturation and variability. *Field Methods*, *18*(1), 59–82.

Harguess, J., & Ward, C. M. (2022). Is the next winter coming for AI? Elements of making secure and robust AI. In 2022 IEEE Applied Imagery Pattern Recognition Workshop (AIPR) (pp. 1-7). IEEE. https://doi.org/10.1109/AIPR57179.2022.10092230

Henrich, J., Heine, S. J., & Norenzayan, A. (2010). The weirdest people in the world? Behavioral and Brain Sciences, 33, 61–135. https://doi.org/10.1017/S0140525X0999152X

Hicks, C. M., Lee, C. S., & Foster-Marks, K. L. (2023). The new developer: AI skill threat, identity change & developer thriving in the transition to AI-assisted software development. The Developer Success Lab. Retrieved from https://www.pluralsight.com/resource-center/guides/new-developer-research-paper

Highsmith, J. (2009). Agile project management: Creating innovative products (2nd ed.). Addison-Wesley.

Holmström, J., & Carroll, N. (2024). How organizations can innovate with generative AI. *Business Horizons*. https://doi.org/10.1016/j.bushor.2024.02.010

Humphreys, D., Koay, A., Desmond, D., & Mealy, E. (2024). AI hype as a cyber security risk: The moral responsibility of implementing generative AI in business. AI and Ethics, 4(4), 791–804. https://doi.org/10.1007/s43681-024-00443-4

IBM. (n.d.). AI vs. machine learning vs. deep learning vs. neural networks: What's the difference? IBM.com. Retrieved October 1, 2024, from https://www.ibm.com/think/topics/ai-vs-machine-learning-vs-deep-learning-vs-neural-networks

IE. (2023). Survey reveals that 68% of Europeans want government restrictions on AI. Center for the Governance of Change. Retrieved from https://www.ie.edu/cgc/news-and-events/news/european-tech-insights-2023/

Jaspan, C., & Green, C. (2023). Defining, measuring, and managing technical debt. *IEEE Software*, 40(3), 15–19. https://doi.org/10.1109/MS.2023.3242137

Johnson, B., & Smith, J. (2021). Towards ethical data-driven software: Filling the gaps in ethics research & practice. Journal of Software Engineering Ethics, 1–10. https://doi.org/10.1000/s13347-021-00451-w

Kanbur, M., Om Prakash, C., & Kulkarni, P. (2023). Creative AI in software project management. Proceedings of the 2023 2nd International Conference on Futuristic Technologies (INCOFT), 1-6. https://doi.org/10.1109/INCOFT60753.2023.10425234

Kissinger, H. A., Schmidt, E., & Huttenlocher, D. (2021). The age of A.I. and our human future. John Murray (Publishers).

Kokol, P. (2024). The use of AI in software engineering: A synthetic knowledge synthesis of the recent research literature. Information, 15(6), 354. https://doi.org/10.3390/info15060354

Krejcie, R. V., & Morgan, D. W. (1970). Determining sample size for research activities. *Educational and Psychological Measurement*, 30(3), 607–610. https://doi.org/10.1177/001316447003000308

Krosnick, J. A. (1999). Survey research. Annual Review of Psychology, 50(1), 537–567.

Kumar, R., Naveen, V., Illa, P. K., Pachar, S., & Patil, P. (2023). The current state of software engineering employing methods derived from artificial intelligence and outstanding challenges. In 2023 1st International Conference on Innovations in High Speed Communication and Signal Processing (IHCSP) (pp. 105–108). IEEE. https://doi.org/10.1109/IHCSP56702.2023.10127112

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521, 436–444. https://doi.org/10.1038/nature14539

Lennox, J. C. (2020). 2084: Artificial intelligence and the future of humanity. Zondervan.

Li, F.-F. (2015). *How we're teaching computers to understand pictures* [Video]. TED. https://www.ted.com/talks/fei_fei_li_how_we_re_teaching_computers_to_understand_pictures/transcript?lng=de&geo=de

Lins, S., Pandl, K. D., Teigeler, H., Thiebes, S., Bayer, C., & Sunyaev, A. (2021). Artificial Intelligence as a Service: Classification and Research Directions. Business & Information Systems Engineering, 63(4), 441–456. https://doi.org/10.1007/s12599-021-00708-w

Liu, Y., Le-Cong, T., Widyasari, R., Tantithamthavorn, C., Li, L., Le, X. D., & Lo, D. (2024). Refining ChatGPT-generated code: Characterizing and mitigating code quality issues. ACM Transactions on Software Engineering and Methodology, 33(5), Article 116. https://doi.org/10.1145/3643674

Lou, A. T. F., & Li, E. Y. (2017). Integrating innovation diffusion theory and the technology acceptance model: The adoption of blockchain technology from business managers' perspective. Proceedings of the 17th International Conference on Electronic Business, 293-296.

Markides, C. (2006). Disruptive Innovation: In Need of Better Theory. Journal of Product Innovation Management, 23, 19-25.

McCarthy, J. (2007). What is artificial intelligence? Stanford University. http://www-formal.stanford.edu/jmc/whatisai.pdf

Meesters, M., Heck, P., & Serebrenik, A. (2022). What is an AI engineer? An empirical analysis of job ads in the Netherlands. Proceedings of the 1st International Conference on AI Engineering, 136-144. https://doi.org/10.1145/3522664.3528594

Mell, P., & Grance, T. (2011). The NIST definition of cloud computing. National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-145

Miles, M. B., Huberman, A. M., & Saldaña, J. (2014). Qualitative data analysis: A methods sourcebook (3rd ed.). Sage Publications.

Naveen, V., Kumar, R., Illa, P. K., & Pachar, S. (2023). The current state of software engineering employing methods derived from artificial intelligence and outstanding challenges. Proceedings of the 1st IEEE International Conference on Innovations in High-Speed Communication and Signal Processing, 105-108. https://doi.org/10.1109/IHCSP56702.2023.10127112

Nguyen-Duc, A., Cabrero-Daniel, B., Przybylek, A., Arora, C., Khanna, D., Herda, T., Rafiq, U., Melegati, J., Guerra, E., Kemell, K.-K., Saari, M., Zhang, Z., Le, H., Quan, T., & Abrahamsson, P. (2023). Generative artificial intelligence for software engineering: A research agenda. arXiv. https://doi.org/10.48550/arXiv.2310.18648

Nilsson, N. J. (2009). The quest for artificial intelligence. Cambridge University Press.

OECD. (2024). OECD AI review: Germany. OECD AI Policy Observatory. Retrieved from https://oecd.ai/en/wonk/review-germany-2024

Olsson, U. (2022). Power properties of ordinal regression models for Likert type data. Practical Assessment, Research & Evaluation, 27(6). https://doi.org/10.7275/er28-5992

O'Reilly, C. A., & Tushman, M. L. (2008). Ambidexterity as a dynamic capability: Resolving the innovator's dilemma. Research in Organizational Behavior, 28, 185-206. https://doi.org/10.1016/j.riob.2008.06.002.

Peng, S., Kalliamvakou, E., Cihon, P., & Demirer, M. (2023). The impact of AI on developer productivity: Evidence from GitHub Copilot. arXiv. https://arxiv.org/abs/2302.06590

Polites, G. L., Roberts, N., & Thatcher, J. (2012). Conceptualizing models using multidimensional constructs: A review and guidelines for their use. *European Journal of Information Systems, 21*(1), 22–48. https://doi.org/10.1057/ejis.2011.10

Qu, S., & Dumay, J. (1995). The qualitative research interview. Qualitative research in accounting & management.

Radack, S. (2009). The system development life cycle (SDLC). ITL Bulletin. National Institute of Standards and Technology, Gaithersburg, MD. https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=902622 (Accessed September 30, 2024).

Rajbhoj, A., Somase, A., Kulkarni, P., & Kulkarni, V. (2024). Accelerating software development using generative AI: ChatGPT case study. Proceedings of the 17th Innovations in Software Engineering Conference (ISEC 2024), 1-11. https://doi.org/10.1145/3641399.3641403

Rogers, E. M. (1983). Diffusion of innovations (3rd ed.). The Free Press.

Rowley, J. (2012). Conducting research interviews. *Management Research Review, 35*(3/4), 260–271. https://doi.org/10.1108/01409171211210154

Royce, W. W. (1987, March). Managing the development of large software systems: Concepts and techniques. In Proceedings of the 9th International Conference on Software Engineering (pp. 328–338).

Rumelhart, D., Hinton, G., & Williams, R. (1986). Learning representations by back-propagating errors. Nature, 323, 533–536. https://doi.org/10.1038/323533a0

Russell, S., & Norvig, P. (2021). Artificial intelligence: A modern approach (4th ed.). Pearson.

Russo, D. (2024). Navigating the complexity of generative AI adoption in software engineering. ACM Transactions on Software Engineering and Methodology, 33(5), 135. https://doi.org/10.1145/3652154.

Saleem, I., Hoque, S. M. S., Tashfeen, R., & Weller, M. (2023). The interplay of AI adoption, IoT edge, and adaptive resilience to explain digital innovation: Evidence from German family-owned SMEs. Journal of Theoretical and Applied Electronic Commerce Research, 18(3), 1419-1430. https://doi.org/10.3390/jtaer18030071

Saravanos, A., & Curinga, M. X. (2023). Simulating the software development lifecycle: The waterfall model. Applied System Innovation, 6(108), 1-19. https://doi.org/10.3390/asi6060108

Saunders, M., Lewis, P., & Thornhill, A. (2007). *Research methods for business students* (4th ed.). Pearson Education.

Schmidt, C. (2016). Agile software development. In Agile software development teams. Progress in IS. Springer, Cham. https://doi.org/10.1007/978-3-319-26057-0_2

Schneider, J., Shah, T., & Abraham, J. C. (2024). Navigating the generative AI disruption in software. McKinsey & Company.

Schumpeter, J. A. (2006). Capitalism, socialism and democracy. Routledge. (Original work published 1942)

Silicon Saxony. (2023). Bitkom: Shortage of IT specialists threatens to worsen dramatically. Silicon Saxony. Retrieved October 1, 2024, from https://silicon-saxony.de/en/bitkom-shortage-of-it-specialists-threatens-to-worsen-dramatically/

Solans-Domènech, M., Pons, J. M. V., Adam, P., Grau, J., & Aymerich, M. (2019). Development and validation of a questionnaire to measure research impact. *Research Evaluation, 28*(3), 253–262. https://doi.org/10.1093/reseval/rvz007

Sommerville, I. (2011). Software engineering (9th ed.). Addison-Wesley.

Souza, J. M. de, & Magalhães, J. P. (2024). Artificial intelligence in the workplace: A systematic review of skill transformation in the industry. Administrative Sciences, 14(6), 127. https://doi.org/10.3390/admsci14060127

Stack Overflow. (2024). Stack Overflow Developer Survey 2024. Retrieved from https://survey.stackoverflow.co/2024/

Strode, D., Dingsøyr, T., & Lindsjorn, Y. (2022). A teamwork effectiveness model for agile software development. Empirical Software Engineering, 27(56). https://doi.org/10.1007/s10664-021-10115-0

Teece, D. J. (2007). Explicating dynamic capabilities: The nature and microfoundations of (sustainable) enterprise performance. *Strategic Management Journal, 28*(13), 1319-1350. https://doi.org/10.1002/smj.640

Teece, D. J., Pisano, G., & Shuen, A. (1997). Dynamic capabilities and strategic management. *Strategic Management Journal*, *18*(7), 509–533. http://www.jstor.org/stable/3088148

Thomson, A. J., & Schmoldt, D. L. (2001). Ethics in computer software design and development. Computers and Electronics in Agriculture, 30(1), 85–102. https://doi.org/10.1016/S0168-1699(00)00158-7

Tom, E., Aurum, A., & Vidgen, R. (2013). An exploration of technical debt. *The Journal of Systems and Software, 86*(6), 1498–1516. https://doi.org/10.1016/j.jss.2012.12.052

Training Industry. (2023). Generative AI's impact on jobs and the urgency of upskilling. Training Industry. Retrieved from https://trainingindustry.com/articles/strategy-alignment-and-planning/generative-ais-impact-on-jobs-and-the-urgency-of-upskilling/

Turing, A. M. (1950). Computing machinery and intelligence. Mind, 59(236), 433–460. https://doi.org/10.1093/mind/LIX.236.433

Venkatesh, V., Morris, M. G., Davis, G. B., & Davis, F. D. (2003). User acceptance of information technology: Toward a unified view. MIS Quarterly, 27(3), 425-478.

Weber, T., Brandmaier, M., Schmidt, A., & Mayer, S. (2024). Significant productivity gains through programming with large language models. Proceedings of the ACM on Human-Computer Interaction, 8(EICS), Article 256, 1-29. https://doi.org/10.1145/3661145

Woiceshyn, J., & Daellenbach, U. S. (2018). Evaluating inductive versus deductive research in management studies: Implications for authors, editors, and reviewers. *Qualitative Research in Organizations and Management: An International Journal, 13*(2), 183-195. https://doi.org/10.1108/QROM-06-2017-1538

World Economic Forum. (2023). The future of jobs report 2023. World Economic Forum. Retrieved from https://www.weforum.org/reports/the-future-ofjobs-report-2023/

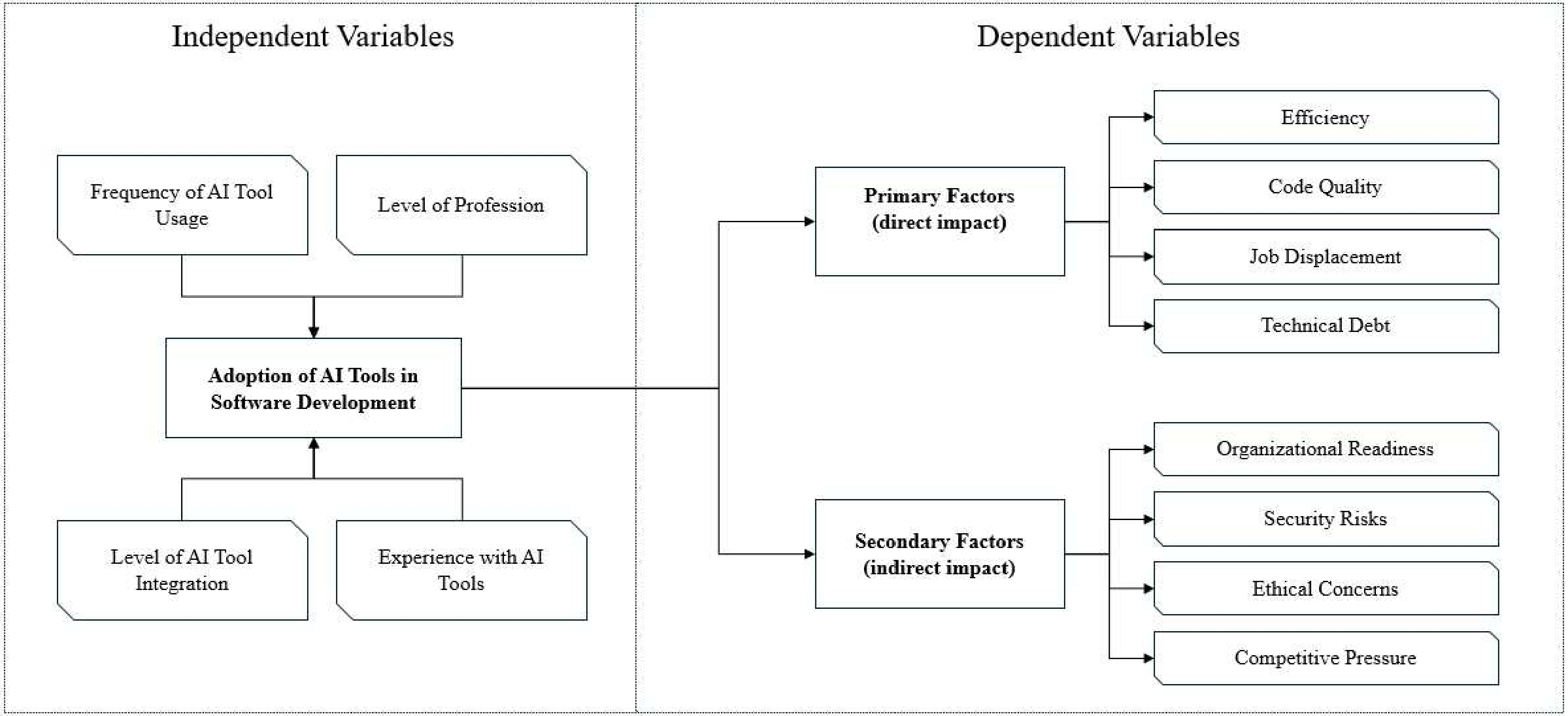
Yetiştiren, B., Özsoy, I., Ayerdem, M., & Tüzün, E. (2023). Evaluating the code quality of AI-assisted code generation tools: An empirical study on GitHub Copilot, Amazon CodeWhisperer, and ChatGPT. arXiv preprint arXiv:2304.10778v2. https://doi.org/10.48550/arXiv.2304.10778v2

7. Appendices

7.1. Appendix A – Conceptual Model

Figure 23

Conceptual Model



7.2. Appendix B – Interview Guide

Table 2

Interview Guide

#	**Question**
1.	Could you please briefly introduce yourself and provide details of your background and current role at [COMPANY]?
2.	AI tools such as ChatGPT and GitHub Copilot are becoming increasingly important in various areas of software development. What experience have you had with AI tools in your role so far? Follow-up questions: • Name the tools that you use most frequently. • How long have you been using these AI tools in your work? • How often do you use them? • Please estimate: In what percentage of your tasks do you use AI tools to assist you?
3.	AI tools can help to make routine tasks in software development more efficient and automated. Which tasks in software development do you think will benefit most from this? Follow-up question: • How much do you estimate the efficiency gain and the degree of automation through the use of AI tools in software development?
4.	Technical debt arises when short-term, quick solutions are chosen in software development that lead to additional costs and extra work in the long term. To what extent do you see the risk of technical debt when using AI tools in software development?
5.	The use of AI tools in software programming entails potential security risks. What security risks do you see when using AI tools in software programming?
6.	How has the use of AI tools affected the quality of code in your software development processes and projects?
7.	The use of AI tools could change the roles of employees in software development, and it is often discussed whether this could make certain positions redundant. Do you think AI will require new skills or roles in your field of work and could potentially replace traditional positions? Follow-up question: • In your opinion, what new roles and skills could arise and be required in the future through the use of AI tools in software development?
8.	Ethics in AI is a constantly debated topic where experts often have different views on what is right and wrong to do. In your opinion, how important are ethical principles when using AI tools in software development and what needs to be considered?
9.	Do you think that companies that are reluctant to integrate AI tools into their development processes will come under increased competitive pressure and could

	possibly lose relevance in the coming years? If so, what do you see as the reasons for this?
10.	How do you rate the importance of organizational adaptation for the successful introduction of AI tools in software development? Follow-up question: • What specific measures, such as structured training programs and the promotion of an innovation-oriented corporate culture, do you consider crucial to prepare employees for the necessary changes?
11.	**Sustaining Innovation**: Improvements to existing software solutions that optimize the development process for current users. **Efficiency Innovation**: Process optimizations in software development that reduce time and costs and increase efficiency. **Disruptive innovation**: New approaches or tools in software development that change the traditional development process or create new opportunities. How would you assess the role of AI tools in software development: do you see them more as "Sustaining Innovation", "Efficiency Innovation" or "Disruptive Innovation", and why?
12.	How do you rate the usefulness and usability of AI tools in your software development practice, and what factors do you think influence the adoption of these technologies in your team?
13.	Dynamic capabilities are a company's ability to adapt to changing market conditions, take advantage of new opportunities and reorganize existing resources effectively. Do you believe that these capabilities of your company are supported by the integration of AI tools?
14.	In your opinion, what factors currently have the greatest influence - whether positive or negative - on software development through the use of AI tools? Please rank the factors from 1 to 8, with 1 being the strongest influence and 8 being the weakest influence. Factors: • Efficiency • Code quality • Job displacement • Technical debt • Organizational readiness • Security risks • Ethical concerns • Competitive pressure
15.	How do you assess the impact of AI on software development in small to medium-sized enterprises (SMEs) compared to large enterprises (LEs)? Follow-up question: • Do you see any significant differences in the effects, and if so, what specific factors do you think play a role?
16.	There is much discussion about the fact that AI could significantly change software development and replace traditional processes. Do you believe that AI tools will really bring about a fundamental **disruption** in software development?

7.1. Appendix C – Deductive Coding Structure

Table 3

Deductive Coding Structure

Code	Keywords
Efficiency: Confirm	Automation, productivity, time-saving, optimization, streamlined processes
Efficiency: Refute	Lack of productivity improvements, ineffective automation, time-consuming processes, over-reliance
Code Quality: Confirm	Clean code, readability, standardization, reduced errors, robust testing
Code Quality: Refute	Code errors, bugs, inconsistencies, code smells, poor documentation, over-reliance, human intervention
Job Displacement: Confirm	Workforce transformation, automation of tasks, redundancy, role shifts
Job Displacement: Refute	Workforce stability, manual execution of tasks, job retention, role consistency
Technical Debt: Confirm	Neglect of maintenance, increased debugging, inability to scale, design flaws, over-reliance
Technical Debt: Refute	Sustainable design, scalable solutions, minimized maintenance costs
Organizational Readiness: Confirm	Training programs, skill development, adaptability, innovation mindset
Organizational Readiness: Refute	Resistance to change, lack of training, cultural inertia, unpreparedness
Security: Confirm	Cybersecurity, data protection, encryption, risk management
Security: Refute	Minimal vulnerabilities, robust safeguards, minimal oversight
Ethics: Confirm	Privacy concerns, intellectual property, algorithmic bias, accountability
Ethics: Refute	Minimal privacy risks, no significant bias

Competitive Pressure: Confirm	Market competition, early adoption, productivity advantages, competitive edge
Competitive Pressure: Refute	Minimal industry disruption, stable market dynamics, limited adoption impact
Technology Acceptance Model: Perceived Ease of Use	Intuitive interfaces, user-friendliness, simplicity, usability
Technology Acceptance Model: Perceived Usefulness	Efficiency benefits, problem-solving, practical applications
Type of Innovation: Disruptive Innovation	Market transformation, new technology adoption, paradigm shifts
Type of Innovation: Efficiency Innovation	Cost reduction, productivity optimization, streamlined workflows
Type of Innovation: Sustaining Innovation	Incremental improvements, feature enhancements, existing market focus
Diffusion of Innovation: Compatibility	Alignment with existing processes, integration challenges, system compatibility
Diffusion of Innovation: Early Adopter	Innovation leaders, trendsetters, first-mover advantage
Diffusion of Innovation: Relative Advantage	Clear benefits, cost-effectiveness, performance superiority
Diffusion of Innovation: Trialability	Experimentation, pilot projects, phased implementation
Dynamic Capabilities: Adaptability	Resource reconfiguration, flexibility, market responsiveness
Dynamic Capabilities: Ambidexterity	Balancing exploration and exploitation, dual strategies, resource management

7.1. Appendix D – Frequency of Codes

Figure 24

Frequency of Codes Applied

		Project Managers			Developers										
Codes	Sum	PM1	PM2	PM3	DEV1	DEV2	DEV3	DEV4	DEV5	DEV6	DEV7	DEV8	DEV9	DEV10	DEV11
Efficiency: Confirm	150	15	11	18	3	13	10	11	16	15	18	5	11	4	12
Security: Confirm	145	9	14	18	11	14	10	19	17	12	6	4	10	1	11
Job Displacement: Confirm	103	19	11	13	5	13	9	13	7	3	6	1	2	1	4
Organizational Readiness: Confirm	94	18	13	12	9	7	3	5	12	2	3	3	4	3	3
Technical Debt: Confirm	90	5	8	6	10	12	12	10	8	6	3	5	4	1	1
Ethics: Confirm	77	3	9	9	6	3	6	8	7	9	7	5	5	0	5
Technology Acceptance Model: Perceived Usefulness: Confirm	76	6	5	12	11	3	1	6	4	8	9	1	9	1	1
Technology Acceptance Model: Perceived Ease of Use: Confirm	70	10	5	9	9	4	2	2	4	10	8	1	5	1	1
Code Quality: Refute	59	0	8	8	3	14	1	6	5	2	3	4	3	2	2
Effciency: Refute	50	0	7	4	3	10	3	7	4	1	4	6	1	0	3
Dynamic Capabilities: Adaptability	46	10	3	5	2	2	1	4	3	6	5	2	1	2	1
Code Quality: Confirm	38	8	3	6	3	1	2	2	2	7	1	0	2	1	3
Job Displacement: Refute	34	1	2	0	5	3	0	2	3	3	2	6	4	3	1
Competitive Pressure: Confirm	33	1	3	3	4	2	2	0	4	4	2	1	6	1	2
Type of Innovation: Disruptive Innovation	33	3	5	1	0	1	2	3	5	0	3	1	6	3	2
Type of Innovation: Efficiency Innovation	30	2	2	2	2	2	2	2	2	5	5	2	1	1	1
Organizational Readiness: Refute	28	1	0	1	4	5	0	0	0	6	7	3	1	0	0
Dynamic Capabilities: Ambidexterity	24	3	4	6	1	4	0	0	0	0	3	1	1	1	1
Diffusion of Innovation: Compatibility	22	6	0	1	4	5	2	2	1	0	0	0	0	1	1
Technical Debt: Refute	17	0	0	3	0	2	0	2	4	1	4	0	0	1	4
Competitive Pressure: Refute	14	1	0	2	2	1	2	3	0	1	0	2	0	0	0
Diffusion of Innovation: Trialability	11	4	1	3	0	0	1	1	0	0	1	0	0	0	0
Ethics: Refute	11	0	0	0	1	1	0	1	2	0	4	0	1	1	0
Diffusion of Innovation: Early Adopter	8	3	1	2	0	0	1	0	0	0	0	0	1	0	0
Type of Innovation: Sustaining Innovation	6	0	0	0	0	0	1	2	1	1	0	0	1	0	0
Diffusion of Innovation: Relative Advantage	5	1	1	1	0	1	0	0	0	0	0	0	1	0	1
Security: Refute	3	0	0	0	1	0	0	1	1	0	0	0	0	0	1

7.2. Appendix E – Survey Design

Table 4

Survey Design

#	Question	Type	Answer options
Demographics			
1	How old are you?	Multiple Choice	18 - 24 years old; 25 - 34 years old; 35 - 44 years old; 45 - 54 years old; 55 - 64 years old; 65 years and older
2	How do you describe your gender identity?	Multiple Choice	Male; Female; Non-binary/Third gender; Prefer not to say
3	What is the highest level of education you have completed or the highest degree you have attained?	Multiple Choice	Less than high school degree; High school graduate; Vocational training; Bachelor's degree (or equivalent); Master's degree (or equivalent); Doctoral degree
4	What is your annual gross income range?	Multiple Choice	Less than 40,000 Euro; 40,000 Euro - 59,999 Euro; 60,000 Euro - 79,999 Euro; 80,000 Euro - 99,999 Euro; 100,000 Euro and more
5	What is/are your role/s in software development?	Multiple Choice	Product Manager; Project Manager; Product Owner; Business Analyst; Development Manager; Software Architect; Software Developer; UX/UI Designer; Quality Assurance Engineer (QA); Scrum Master; System Administrator; Tester; Technical Director; Other (please specify)
6	How many years of experience do you have in software development?	Multiple Choice	0-2 years; 3-5 years; 6-10 years; 11-15 years; 16+ years

7	What is the size of the company you currently work at?	Multiple Choice	1-10 employees; 11-50 employees; 51-200 employees; 201-500 employees; 501+ employees
AI Tool Usage and Integration			
8	Which AI tools do you use in software development?	Multiple Choice	None; GitHub Copilot; ChatGPT; Google Gemini; Bing AI; Other (please specify)
9	How often do you use AI tools in your work?	Multiple Choice	Daily; Weekly; Monthly; Rarely; Never
10	What percentage of your software tasks are assisted or automated by AI tools?	Sliding Scale	0% (Not at all) to 100% (Fully automated)
11	How proficient are you with using AI tools?	Multiple Choice	1 (Beginner); 2 (Novice); 3 (Intermediate); 4 (Advanced); 5 (Expert)
12	How long have you been using AI tools in software development?	Multiple Choice	Less than 1 year; 1-2 years; 3-4 years; 5+ years
13	Please indicate your level of agreement with the following statements: “AI tools allow me to focus on higher-level strategic work.”	5-Point Likert Scale	Strongly Disagree; Disagree; Neutral; Agree; Strongly Agree
14	What aspects of AI do you find most beneficial in software development?	Multiple Choice	Writing code; Debugging and getting help; Testing code; Code documentation; Other (please specify)
Primary Factors			
15	Please imagine this scenario: *“Your team is instructed to integrate AI tools into their development processes to increase **efficiency** and the degree of automation.”* Please indicate your level of agreement with the following statements: "AI tools will reduce the time needed to review code.” "AI tools can automate most routine tasks in the development cycle." "AI automation reduces the time spent on development tasks." "AI improves the accuracy and reliability of software development processes."	5-Point Likert Scale	Strongly Disagree; Disagree; Neutral; Agree; Strongly Agree
16	Do AI tools improve **code quality** in your software projects?	5-Point Likert Scale	Strongly Disagree; Disagree; Neutral; Agree; Strongly Agree

17	Please imagine this scenario: *"Your team adopts AI-driven tools, resulting in* ***changes to roles*** *or even reductions within the development team."* Please indicate your level of agreement with the following statements: "AI tools reduce the need for certain roles within software development teams." "AI tools eliminate some software development tasks." "AI tools could lead to the creation of completely new roles." "Traditional roles will disappear, but specialized AI roles will prevent complete job loss."	5-Point Likert Scale	Strongly Disagree; Disagree; Neutral; Agree; Strongly Agree
18	Please imagine this scenario: *"Your manager wants you to use AI tools to review code. You are concerned about the* ***complexity*** *and the* ***technical debt*** *for the project moving forward."* Please indicate your level of agreement with the following statements: "The complexity of the project will increase." "The technical debt will increase."	5-Point Likert Scale	Strongly Disagree; Disagree; Neutral; Agree; Strongly Agree
Secondary Factors			
19	Please imagine this scenario: *"For the successful introduction of AI tools to automate testing procedures,* ***organizational readiness*** *is important, including the restructuring of roles and responsibilities."* Please indicate your agreement with the following statements "AI tools increase the demand for new skills and continuous training." "A cultural transformation is needed for companies to redesign roles and processes to fully realize the potential of AI." "An open and innovation-oriented attitude on the part of employees is necessary to	5-Point Likert Scale	Strongly Disagree; Disagree; Neutral; Agree; Strongly Agree

	successfully use AI in software development.”		
20	Do you believe that adopting AI tools in software development introduces **security risks**?	5-Point Likert Scale	Strongly Disagree; Disagree; Neutral; Agree; Strongly Agree
21	Do you believe that the use of AI tools in software development processes raises **ethical concerns**?	5-Point Likert Scale	Strongly Disagree; Disagree; Neutral; Agree; Strongly Agree
22	Would you agree that **competitive pressure** in the software market is increasing due to the adoption of AI tools in development processes?	5-Point Likert Scale	Strongly Disagree; Disagree; Neutral; Agree; Strongly Agree
23	Do you agree that AI tools lead to **better decisions** in project management?	5-Point Likert Scale	Strongly Disagree; Disagree; Neutral; Agree; Strongly Agree
Attention Check			
24	Please select "Strongly Agree" to show that you are paying attention	5-Point Likert Scale	Strongly Disagree; Disagree; Neutral; Agree; Strongly Agree
Ranking Factors by Level of Significance			
25	Please rank the following factors in order of significance for the impact of AI tools on software development, with 1 being the most significant and 8 being the least significant. Factors: - Efficiency - Code Quality - Job Displacement - Technical Debt - Organizational Readiness - Security Risks - Ethical Concerns - Competitive Pressure	Ranking Order	1 being the most significant and 8 being the least significant
Management Theories			
26	How do you rate the role of AI tools in software development? Do you see them as: **Sustaining innovation**: improvements to existing solutions. **Efficiency innovation**: process optimizations to increase efficiency. **Disruptive innovation**: approaches that change the traditional process.	Multiple Choice	Sustaining Innovation; Efficiency Innovation; Disruptive Innovation; Not sure
27	Do you agree with the statement that the **usefulness** and **user-friendliness** of AI tools in software development are high?	5-Point Likert Scale	Strongly Disagree; Disagree; Neutral; Agree; Strongly Agree

28	Do you agree with the statement that the **capabilities** of your company – that is, its ability **dynamically** to adapt to change, identify and seize opportunities in a timely manner, and to be competitive – are supported by the integration of AI tools?	5-Point Likert Scale	Strongly Disagree; Disagree; Neutral; Agree; Strongly Agree
Perception of AI's Future in Software Development			
29	Disruption refers to a fundamental change in a market or industry that **replaces existing technologies**, products or processes **with new and innovative solutions**. Do you think that AI tools will cause a disruption in software development in the future?	5-Point Likert Scale	Strongly Disagree; Disagree; Neutral; Agree; Strongly Agree

7.3. Appendix F – Interview Summaries

Table 5

Interview Summary of PM1

Code: PM1	**Mode:** Virtual	**Date:** 23.10.2024
Time: 7:00 PM	**Duration:** 54 minutes	
Question 2		
Extensive experience with AI tools like ChatGPT, using them for 70% of tasks such as code generation, troubleshooting, and workflow automation.		
Question 3		
Believes AI tools significantly improve efficiency, code quality, and idea generation, especially in coding and structuring tasks.		
Question 4		
Highlights risks of technical debt as reliance on AI may reduce understanding of code structures, causing inefficiencies over time.		
Question 5		
Notes significant security risks, especially regarding data privacy and GDPR compliance when using non-EU AI servers.		
Question 6		
Observes that AI enhances code quality through error detection, optimization, and leaner solutions, acting as a "sparring partner."		
Question 7		
Anticipates AI will require new skills like problem-centered thinking, with roles evolving and emphasizing strategic AI use.		
Question 8		
Stresses ethical principles, urging verification of AI outputs and transparency in training data to mitigate biases and ensure fairness.		
Question 9		
Believes companies delaying AI adoption risk losing relevance as AI enhances productivity and offers a competitive edge.		
Question 10		
Emphasizes structured training and fostering an innovation culture to ensure effective organizational adaptation to AI tools.		
Question 11		
Views AI tools as "Efficiency Innovation" that streamlines processes, while also offering potential for "Disruptive Innovation."		
Question 12		
Rates AI as user-friendly, noting trust and ease of use drive adoption, with efficiency benefits increasing acceptance.		
Question 13		
Reports AI enhances adaptability and innovation by fostering knowledge transfer, but notes some structural limits to flexibility.		
Question 14		
Ranks factors: Efficiency; Code quality; Competitive pressure; Security risks; Ethical concerns; Technical debt; Job displacement; Organizational readiness.		
Question 15		

Observes large enterprises benefit from resources but face slower adoption, while SMEs adapt faster but lack expertise and funding.
Question 16
Believes AI will transform software development, enhancing efficiency and automating processes, evolving rather than replacing traditional methods.

Table 6

Interview Summary of PM2

Code: PM2	**Mode:** Virtual	**Date:** 25.10.2024
Time: 9:00 PM	**Duration:** 1 hour 10 minutes	
Question 2		
Uses ChatGPT and in-house AI chatbot "DirectChat" daily for 30-40% of tasks, aiding efficiency across projects.		
Question 3		
Believes AI improves efficiency in coding tasks by automating foundational work, though human oversight is needed.		
Question 4		
Notes technical debt risks from AI-generated code's opacity, which complicates debugging and long-term maintenance.		
Question 5		
Highlights security risks like data leakage, stressing in-house systems and compliance to address privacy concerns.		
Question 6		
Sees mixed impact on code quality; AI aids less experienced developers but lacks precision for complex projects.		
Question 7		
Predicts AI tools will shift roles toward debugging AI-generated code, with skills in AI oversight becoming crucial.		
Question 8		
Emphasizes ethics, stressing transparency, control, and avoiding biases, especially in diverse cultural contexts.		
Question 9		
Believes delayed AI adoption leads to competitive disadvantages, as AI improves efficiency and democratizes development.		
Question 10		
Advocates for structured training and an innovation-driven culture to support AI integration and empower employees.		
Question 11		
Views AI as disruptive innovation, enabling non-experts to contribute but raising challenges in code quality and security.		
Question 12		
Highlights AI's ease of use and efficiency benefits, with organizational support and training fostering adoption.		
Question 13		
Notes AI strengthens adaptability and innovation by enabling employees to create practical tools independently.		

Question 14
Ranks factors: Efficiency; Code quality; Technical debt; Competitive pressure; Security risks; Ethical concerns; Organizational readiness; Job displacement.
Question 15
Notes SMEs gain agility with AI but face resource and data challenges, while LEs benefit from greater external resources.
Question 16
Believes AI disrupts development by reducing reliance on traditional roles, requiring new solutions for quality and security.

Table 7

Interview Summary of PM3

Code: PM3	**Mode:** Face-To-Face	**Date:** 06.11.2024
Time: 01:00 PM	**Duration:** 1 hour 2 minutes	
Question 2		
Uses ChatGPT, DALL-E, and Copilot for 50% of tasks, with senior developers relying less on AI tools.		
Question 3		
AI is beneficial for routine coding and documentation but requires human oversight for debugging and complex tasks.		
Question 4		
Warns of technical debt risks if best practices are neglected, especially for new developers over-relying on AI.		
Question 5		
Cites data residency and dependency risks, stressing strict data protection and limiting over-reliance on AI tools.		
Question 6		
Notes AI improves initial code quality but stresses that overuse without oversight can reduce long-term standards.		
Question 7		
Predicts AI will automate simpler tasks while creating roles in data engineering and ML operations.		
Question 8		
Advocates for global ethical standards, balancing compliance with flexibility across diverse cultural contexts.		
Question 9		
Believes competitive pressure depends on company size, with larger firms maintaining competitiveness without AI.		
Question 10		
Highlights structured training and a lifelong learning culture as key for overcoming resistance to AI integration.		
Question 11		
Views AI as primarily an efficiency enhancer, optimizing processes rather than disrupting software development.		
Question 12		

Finds AI tools useful and user-friendly, with adoption driven by integration into familiar platforms and workflows.
Question 13
Stresses cultural adaptability among developers to leverage AI, though rapid change presents challenges.
Question 14
Ranks factors: Efficiency; Code quality; Technical debt; Security risks; Organizational readiness; Ethical concerns; Competitive pressure; Job displacement.
Question 15
Notes SMEs face resource challenges with AI, while LEs benefit from custom AI projects and better scalability.
Question 16
Sees AI as an efficiency driver for large firms, with potential disruption for SMEs relying heavily on AI.

Table 8

Interview Summary of DEV1

<table>
<tr><td colspan="2">Code: DEV1</td><td colspan="2">Mode: Face-To-Face</td><td colspan="2">Date: 29.10.2024</td></tr>
<tr><td colspan="3">Time: 16:00 PM</td><td colspan="3">Duration: 47 minutes</td></tr>
<tr><td colspan="6">Question 2</td></tr>
<tr><td colspan="6">Uses ChatGPT for 10% of tasks, primarily for standard routines, with limited reliance on other AI tools.</td></tr>
<tr><td colspan="6">Question 3</td></tr>
<tr><td colspan="6">Finds AI beneficial for routine coding tasks, enhancing efficiency but requiring human oversight for complex work.</td></tr>
<tr><td colspan="6">Question 4</td></tr>
<tr><td colspan="6">Warns of technical debt if developers lack expertise to assess AI-generated outputs, stressing alignment with standards.</td></tr>
<tr><td colspan="6">Question 5</td></tr>
<tr><td colspan="6">Highlights data protection risks and the need to review AI-generated code to prevent malicious inclusions.</td></tr>
<tr><td colspan="6">Question 6</td></tr>
<tr><td colspan="6">Reports AI has limited impact on code quality, with final output depending on critical review and personal programming style.</td></tr>
<tr><td colspan="6">Question 7</td></tr>
<tr><td colspan="6">Believes AI will not displace roles in small teams but may shift standardized tasks to AI in larger organizations.</td></tr>
<tr><td colspan="6">Question 8</td></tr>
<tr><td colspan="6">Stresses ethical oversight to prevent biases in AI training data, particularly for impactful decisions like recruitment.</td></tr>
<tr><td colspan="6">Question 9</td></tr>
<tr><td colspan="6">Notes competitive pressure for slow adopters, as AI enhances productivity and reduces costs in standardized industries.</td></tr>
<tr><td colspan="6">Question 10</td></tr>
<tr><td colspan="6">Emphasizes innovation-driven culture for AI integration, with developers exploring technologies independently.</td></tr>
</table>

Question 11
Views AI as "Efficiency Innovation," optimizing processes and reducing costs without fundamentally disrupting development.
Question 12
Finds AI tools useful for repetitive tasks, with adoption influenced by ease of integration and experimentation.
Question 13
Notes AI supports dynamic capabilities by automating routine tasks, enabling focus on strategic initiatives.
Question 14
Ranks factors: Efficiency; Organizational readiness; Code quality; Technical debt; Security risks; Competitive pressure; Job displacement; Ethical concerns.
Question 15
Sees SMEs benefiting from faster adaptation, while LEs gain from extensive resources and infrastructure for AI integration.
Question 16
Believes AI enhances efficiency but supplements rather than disrupts current development practices, with future potential.

Table 9

Interview Summary of DEV2

<table>
<tr><td>Code: DEV2</td><td colspan="2">Mode: Virtual Interview</td><td>Date: 31.10.2024</td></tr>
<tr><td colspan="2">Time: 15:00 PM</td><td colspan="2">Duration: 1 hour 34 minutes</td></tr>
<tr><td colspan="4">Question 2</td></tr>
<tr><td colspan="4">Uses ChatGPT weekly for self-education and 10-20% of tasks, avoiding GitHub Copilot due to irrelevant suggestions.</td></tr>
<tr><td colspan="4">Question 3</td></tr>
<tr><td colspan="4">Sees AI benefiting testing and documentation, though its role in core development tasks is limited to secondary activities.</td></tr>
<tr><td colspan="4">Question 4</td></tr>
<tr><td colspan="4">Warns of technical debt and security risks with AI-generated code, emphasizing experienced oversight to ensure quality..</td></tr>
<tr><td colspan="4">Question 5</td></tr>
<tr><td colspan="4">Highlights vulnerabilities like SQL injection risks and data protection concerns when sharing sensitive code with AI.</td></tr>
<tr><td colspan="4">Question 6</td></tr>
<tr><td colspan="4">Reports AI offers useful ideas but requires substantial corrections, sometimes degrading code quality if misused.</td></tr>
<tr><td colspan="4">Question 7</td></tr>
<tr><td colspan="4">Predicts new roles in prompt engineering and oversight, with job shifts being less significant in smaller companies.</td></tr>
<tr><td colspan="4">Question 8</td></tr>
<tr><td colspan="4">Emphasizes ethical principles, warning of biases in training data and challenges in creating universal ethical standards.</td></tr>
<tr><td colspan="4">Question 9</td></tr>
</table>

Believes slow adopters face pressure, as AI boosts efficiency and lowers costs, with greater impact on larger firms.
Question 10
Views adaptation as vital, driven by proactive developers, with organizational support enhancing innovation adoption.
Question 11
Classifies AI as "Efficiency Innovation," streamlining repetitive tasks without fundamentally transforming development.
Question 12
Finds AI tools useful and user-friendly, with adoption driven by ease of use and teams' technical curiosity.
Question 13
Notes AI enhances dynamic capabilities but cautions high implementation costs limit practicality for smaller firms.
Question 14
Ranks factors: Competitive pressure; Security risks; Ethical concerns; Job displacement; Efficiency; Code quality; Technical debt; Organizational readiness.
Question 15
Sees AI's greater impact in LEs due to resources, while SMEs adopt flexibly but are limited by costs and scale.
Question 16
Views AI as overhyped, useful for efficiency but insufficient to replace traditional processes or deterministic methods.

Table 10

Interview Summary of DEV3

Code: DEV3	**Mode:** Face-To-Face	**Date:** 02.11.2024
Time: 11:00 AM	**Duration:** 1 hour	
Question 2		
Uses AI monthly, primarily for non-development tasks and SQL, employing AI in 5% of work, especially for simple tasks.		
Question 3		
Finds AI beneficial for code writing, documentation, and SQL, streamlining tasks with background knowledge.		
Question 4		
Warns of technical debt if oversight is lacking, emphasizing the need for experienced developers to ensure quality.		
Question 5		
Highlights security risks from AI-generated errors, cautioning against over-reliance that reduces developer independence.		
Question 6		
Reports limited impact on code quality as AI tools are sparingly used, requiring validation to meet internal standards.		
Question 7		
Suggests AI may replace some roles but foresees new positions for validating and managing AI-generated outputs.		

Question 8
Stresses ethical principles, urging user responsibility for oversight to ensure outputs meet ethical and operational standards.
Question 9
Notes non-adopters may lose competitiveness in certain industries, with political factors influencing tool adoption.
Question 10
Emphasizes structured AI integration protocols and continuous training to keep pace with evolving technologies.
Question 11
Views AI as enhancing efficiency without disrupting practices but anticipates future disruptive potential in development.
Question 12
Finds AI useful and user-friendly, with adoption driven by perceived benefits, ease of use, and accessible training.
Question 13
Confirms AI improves dynamic capabilities, streamlining workflows and adapting to changing market demands.
Question 14
Ranks factors: Security Risks; Efficiency; Competitive Pressure; Ethical Concerns; Code Quality; Job Displacement; Technical Debt; Organizational readiness.
Question 15
Notes LEs face greater job displacement but have resources for specialized expertise, while SMEs benefit from agility.
Question 16
Sees potential for AI to disrupt development by automating repetitive tasks, though timelines remain uncertain.

Table 11

Interview Summary of DEV4

<table>
<tr><td>Code: DEV4</td><td>Mode: Virtual</td><td>Date: 08.11.2024</td></tr>
<tr><td colspan="3">Time: 10:00 AM   Duration: 1 hour 13 minutes</td></tr>
<tr><td colspan="3">Question 2</td></tr>
<tr><td colspan="3">Limited personal use of AI, experimenting with ChatGPT and observing colleagues. Skepticism due to mixed results.</td></tr>
<tr><td colspan="3">Question 3</td></tr>
<tr><td colspan="3">AI could streamline tasks like automating class generation but may impact code consistency in creative projects.</td></tr>
<tr><td colspan="3">Question 4</td></tr>
<tr><td colspan="3">Warns of technical debt if AI is used without understanding, as it may provide flawed solutions without warnings.</td></tr>
<tr><td colspan="3">Question 5</td></tr>
<tr><td colspan="3">Cites risks of sharing sensitive data with AI models and emphasizes the importance of protecting proprietary information.</td></tr>
<tr><td colspan="3">Question 6</td></tr>
</table>

Notes minor benefits for code quality but observes positive outcomes in specific scenarios like language updates.
Question 7
Believes AI may eliminate repetitive roles but will require new roles for managing AI outputs and configurations.
Question 8
Stresses ethical oversight, noting companies often prioritize profit over ethics, leading to insufficient AI monitoring.
Question 9
Believes AI is not essential for competitiveness, though reliance on it without caution can harm quality and reputation.
Question 10
Emphasizes structured training and expert-led introductions to AI, advocating cautious adoption with continuous learning.
Question 11
Views AI as "Efficiency Innovation," enhancing processes without fundamentally disrupting software development.
Question 12
Finds AI moderately useful, with better outcomes from improved prompting. Trust in results drives broader adoption.
Question 13
Notes situational benefits of AI for dynamic capabilities, such as transitioning software between programming languages.
Question 14
Ranks factors: Efficiency; Technical debt; Code quality; Organizational readiness; Security risks; Job displacement; Competitive pressure; Ethical concerns.
Question 15
SMEs adopt AI faster due to flexibility, while LEs face delays from complex approvals and strict data protection.
Question 16
Sees long-term potential for AI to disrupt software development, depending on advances in modular programming tools.

Table 12

Interview Summary of DEV5

<table>
<tr><td>Code: DEV5</td><td>Mode: Virtual</td><td>Date: 11.11.2024</td></tr>
<tr><td colspan="2">Time: 10:00 AM</td><td>Duration: 1 hour 15 minutes</td></tr>
<tr><td colspan="3">Question 2</td></tr>
<tr><td colspan="3">Extensively uses AI tools like ChatGPT and Azure OpenAI for 50% of daily tasks, benefiting prototyping and automation.</td></tr>
<tr><td colspan="3">Question 3</td></tr>
<tr><td colspan="3">AI boosts efficiency in prototyping and routine tasks, with up to 100% automation for simple processes but limits in complexity.</td></tr>
<tr><td colspan="3">Question 4</td></tr>
</table>

Highlights technical debt risks from quick AI fixes, as immediate functionality often overshadows maintainable coding practices.
Question 5
Notes risks like "prompt injection" attacks, insecure APIs, and data exposure, stressing user training to enhance security.
Question 6
AI's impact on code quality is minimal as traditional coding dominates, with AI mostly used for prototyping tasks.
Question 7
Foresees shifts in developer roles to focus on architectural oversight, validation, and managing AI-generated outputs.
Question 8
Stresses the need for "Explainable AI" to ensure transparency and ethical use, balancing functionality with social impacts.
Question 9
Believes non-AI adopters will face competitive pressure, as AI increases efficiency, lowers costs, and boosts market influence.
Question 10
Emphasizes organizational adaptation through training, fostering innovation, and addressing fears of job displacement.
Question 11
Views AI as "Efficiency Innovation," optimizing processes and routine tasks without fundamentally disrupting development.
Question 12
Sees AI as highly useful for prototyping and automation, though concerns remain about security, technical debt, and team readiness.
Question 13
Confirms AI enhances dynamic capabilities, aiding initial analysis but needing internal expertise for strategic decisions.
Question 14
Ranks factors: Competitive pressure; Efficiency; Security risks; Organizational readiness; Job displacement; Code quality; Technical debt; Ethical concerns.
Question 15
SMEs struggle with funding and expertise, while LEs leverage resources for competitive AI adoption, benefiting both sectors.
Question 16
Believes AI will complement rather than replace traditional development, with roles shifting to architecture and oversight.

Table 13

Interview Summary of DEV6

Code: DEV6	**Mode:** Face-To-Face	**Date:** 15.11.2024
Time: 03:30 PM	**Duration:** 52 minutes	
Question 2		
Uses ChatGPT daily for 20-30% of tasks, preferring it over GitHub Copilot for quality but noting limited development integration.		

Question 3
AI enhances efficiency by automating routine tasks like documentation and providing initial structures for complex work.
Question 4
Warns of technical debt risks, emphasizing human oversight to ensure AI remains a tool, not a decision-maker.
Question 5
Cites security risks like vulnerable code and data privacy concerns when sharing sensitive information with AI tools.
Question 6
AI improves code quality by standardizing repetitive tasks and aiding updates, though careful review is necessary.
Question 7
Foresees complementary AI roles like Prompt Engineers but emphasizes that core developer skills remain indispensable.
Question 8
Stresses ethical oversight, ensuring humans remain accountable for decisions and addressing AI's fairness and transparency.
Question 9
Believes non-AI adopters risk inefficiencies and losing competitiveness, though skilled teams may delay adoption temporarily.
Question 10
Suggests fostering interest through demonstrating AI’s capabilities, with personal willingness driving faster adoption.
Question 11
Views AI as "Efficiency Innovation," enhancing processes without disrupting traditional software development workflows.
Question 12
Rates AI tools as useful for routine tasks, with adoption driven by ease of use and the motivating "wow effect."
Question 13
Confirms AI strengthens adaptability, aiding transitions to modern languages and streamlining resource reorganization.
Question 14
Ranks factors: Ethical concerns; Efficiency; Code quality; Organizational readiness; Security risks; Job displacement; Technical debt; Competitive pressure.
Question 15
SMEs adapt faster due to flexibility, while LEs face slower innovation from political issues and complex structures.
Question 16
Believes AI supplements rather than replaces traditional development, requiring skilled interaction for effective use.

Table 14

Interview Summary of DEV7

Code: DEV7	**Mode:** Virtual	**Date:** 18.11.2024
Time: 10:00 AM	**Duration:** 56 minutes	
Question 2		
Uses AI like ChatGPT and Tab9 for 5% of tasks, mainly for simple scripts and boilerplate generation, not complex coding.		
Question 3		
Sees AI boosting efficiency in tasks like unit test generation and static code analysis, but full automation remains aspirational.		
Question 4		
Warns improper AI use can increase technical debt, though proper use can reduce errors and improve code design.		
Question 5		
Highlights risks of vulnerabilities from unchecked AI code but sees AI's potential for security flaw detection.		
Question 6		
Reports limited impact on code quality, as AI is not widely used for critical development, mainly aiding documentation.		
Question 7		
Believes AI will transform roles, with emerging positions in AI integration, particularly in DevOps workflows.		
Question 8		
Stresses ethical responsibility, noting AI may amplify systematic biases but requires careful application to ensure fairness.		
Question 9		
Notes competitive pressure depends on sector, with business logic-heavy industries facing higher adoption demands.		
Question 10		
Views organizational adaptation as rooted in culture, requiring leadership and structured roles to support AI integration.		
Question 11		
Sees AI as disruptive for business logic development but largely efficiency-enhancing for other software processes.		
Question 12		
Finds AI tools context-dependent, with benefits in automation and efficiency varying by IDE integration and workflows.		
Question 13		
Confirms AI enhances dynamic capabilities, such as restructuring code bases and translating programming languages.		
Question 14		
Ranks factors: Efficiency; Code quality; Security risks; Technical debt; Organizational readiness; Job displacement; Ethical concerns; Competitive pressure.		
Question 15		
LEs benefit from scalability for large codebases, while SMEs see relative gains in adaptability and innovation.		
Question 16		

Predicts AI will disrupt development by enhancing efficiency but emphasizes job transformation over displacement.

Table 15

Interview Summary of DEV8

Code: DEV8	**Mode:** Face-To-Face	**Date:** 21.11.2024
Time: 08:30 PM	**Duration:** 37 minutes	
Question 2		
Uses ChatGPT for 15% of tasks, primarily for simple programming, but notes limited applicability in tasks like troubleshooting.		
Question 3		
AI improves efficiency in testing and debugging but offers moderate gains dependent on programming language and platform.		
Question 4		
Warns of technical debt risks if developers rely on AI without understanding generated code or its logic.		
Question 5		
Highlights security risks like data exposure and vulnerabilities from AI prioritizing functionality over secure coding.		
Question 6		
Reports negligible impact on code quality, with 90% of AI-generated code discarded due to required adjustments.		
Question 7		
Believes AI will not replace roles in areas like SAP programming; skills in managing AI outputs may evolve instead.		
Question 8		
Stresses ethical principles, citing risks of bias and lack of empathy, especially in critical applications like hiring.		
Question 9		
Sees limited competitive advantage in AI integration, as pricing and delivery speed outweigh AI benefits in their industry.		
Question 10		
Notes generational differences in AI adoption, advocating structured training and an innovation-driven culture.		
Question 11		
Views AI as "Efficiency Innovation," optimizing routine tasks but requiring human intervention for complex issues.		
Question 12		
Rates AI tools as highly usable but situationally useful, with adoption influenced by task relevance and team openness.		
Question 13		
Skeptical about AI's support for dynamic capabilities, seeing intuition as more influential in strategic decisions.		
Question 14		

Ranks factors: Ethical concerns; Security risks; Efficiency; Technical debt; Organizational readiness; Code quality; Job displacement; Competitive pressure.
Question 15
Notes LEs implement AI extensively due to resources, while SMEs show flexibility but face cost-related constraints.
Question 16
Believes AI will evolve as an assistive tool for routine tasks but cannot fully replace human expertise in complex areas.

Table 16

Interview Summary of DEV9

Code: DEV9	**Mode:** Virtual	**Date:** 23.11.2024
Time: 4:00 PM	**Duration:** 57 minutes	
Question 2		
Uses ChatGPT daily for 60% of tasks, replacing Stack Overflow, aiding function creation, refactoring, and efficiency.		
Question 3		
AI enhances tasks like refactoring and documentation, acting as a "second brain" but requiring manual oversight.		
Question 4		
Warns that uncritical reliance on AI-generated code increases technical debt and long-term complexity.		
Question 5		
Notes security risks like outdated training data and predictable AI-generated code, requiring verification.		
Question 6		
Reports AI improves code structure and readability but has limited impact on functionality without human validation.		
Question 7		
Predicts evolving roles like "AI Engineers," with AI automating coding processes, leaving humans for oversight.		
Question 8		
Emphasizes monitoring for bias, ownership concerns, and ethical oversight to ensure fairness and accountability.		
Question 9		
Believes non-adopters risk losing relevance, as AI boosts efficiency, enabling time for innovation and productivity.		
Question 10		
Advocates structured training to introduce AI tools, tailoring approaches for enthusiasts and hesitant employees.		
Question 11		
Sees AI as "Sustaining and Efficiency Innovation," optimizing coding but not yet disruptive, with future potential.		
Question 12		
Rates AI tools as highly useful, expecting IDE integration to enhance usability and drive broader adoption.		

Question 13
Confirms AI improves dynamic capabilities, enabling faster adaptation to market changes and providing an edge.
Question 14
Ranks factors: Efficiency; Competitive pressure; Ethical concerns; Security risks; Technical debt; Job displacement; Code quality; Organizational readiness.
Question 15
SMEs adapt faster due to agility but face resource limits; LEs leverage resources but encounter slower processes.
Question 16
Believes AI will disrupt development over 10-20 years, with human oversight shaping its transformative potential.

Table 17

Interview Summary of DEV10

Code: DEV10	**Mode:** E-Mail	**Date:** 22.11.2024
Time: 12:00 AM	**Duration:** -	
Question 2		
Uses GitHub Copilot and ChatGPT for 30–40% of tasks, improving efficiency, particularly with Microsoft Dynamics integration.		
Question 3		
AI accelerates tasks like generating tables, offering up to 800% faster automation with tailored prompts and proper use.		
Question 4		
Views technical debt risk as low, attributing errors to conceptual issues rather than AI execution, with QA ensuring quality.		
Question 5		
Emphasizes data protection risks, advising against processing customer data in public AI systems to prevent breaches.		
Question 6		
AI improves code quality by maintaining consistency, though syntax errors require manual review and final oversight.		
Question 7		
Predicts roles like prompt engineering will grow, but traditional positions will remain vital for complex, context-specific tasks.		
Question 8		
Focuses on practical utility over ethical concerns, prioritizing AI tools' ability to simplify and improve workflows.		
Question 9		
Notes competitive pressure arises from missed efficiencies, though reluctance to adopt AI is not commonly observed.		
Question 10		
Highlights infrastructure integration, structured training, and an innovation-driven culture as keys to effective AI adoption.		
Question 11		

Sees AI as "Efficiency Innovation," automating tasks but requiring rework and human input to complete development cycles.
Question 12
Rates AI as highly useful, with adoption driven by integration into existing tools like Microsoft 365 for seamless workflows.
Question 13
Confirms AI enhances dynamic capabilities, supporting adaptation to market changes and optimizing resource allocation.
Question 14
Ranks factors: Efficiency; Organizational readiness; Security risks; Competitive pressure; Ethical concerns; Code quality; Technical debt; Job displacement.
Question 15
Notes AI scales well for SMEs and LEs, with SMEs benefiting from lower complexity and adaptability to tools.
Question 16
Believes AI won't disrupt development yet but expects significant changes in 5–10 years, with human roles remaining essential.

Table 18

Interview Summary of DEV11

Code: DEV11	**Mode:** Virtual Interview	**Date:** 30.11.2024
Time: 11:00 AM	**Duration:** 57 minutes	
Question 2		
Uses GitHub Copilot and ChatGPT daily for 10–20% of work tasks and 80–100% in personal projects, noting security concerns.		
Question 3		
AI aids repetitive tasks like documentation, improving efficiency but struggling with high-complexity functions.		
Question 4		
Warns misuse of AI can increase technical debt, though proper use reduces it by enhancing safeguards and efficiency.		
Question 5		
Highlights risks like sensitive code exposure and hardcoding issues, stressing responsible use to maintain security.		
Question 6		
AI improves code quality by enabling time for optimization and proposing safeguards, but outputs need validation.		
Question 7		
Foresees roles ensuring ethical AI use, compliance, and automation, though job displacement depends on training.		
Question 8		
Stresses ethical oversight to prevent bias and misuse, emphasizing accountability and fairness in AI applications.		
Question 9		
Believes non-adopters risk losing relevance, as AI drives efficiency and enables competitiveness in a fast-changing market.		

Question 10
Highlights the need for training and innovation-driven culture to integrate AI tools effectively and reduce risks.
Question 11
Sees AI as "Sustaining and Efficiency Innovation," enhancing processes without fundamentally disrupting development.
Question 12
Rates AI as useful for automating routine tasks, with adoption influenced by usability, security, and team readiness.
Question 13
Confirms AI supports dynamic capabilities, aiding tasks like code translation and resource reorganization if used wisely.
Question 14
Ranks factors: Competitive pressure; Efficiency; Security risks; Code quality; Ethical concerns; Job displacement; Technical debt; Organizational readiness.
Question 15
Notes SMEs adopt AI more quickly due to flexibility, while LEs face delays from conservative structures and security issues.
Question 16
Believes AI enhances efficiency but won't disrupt development unless true autonomous AI is achieved in the future.