Tampereen yliopisto

Vilma Karppinen

LARGE LANGUAGE MODELS – THE BUILDING BLOCKS OF A SOFTWARE PROFESSIONAL

Current Use of Large Language Models in the Finnish Software Industry and Future Predictions

Faculty of Information Technology and Communication Sciences
Master's Thesis
January 2025

ABSTRACT

Vilma Karppinen: LARGE LANGUAGE MODELS – THE BUILDING BLOCKS OF A SOFTWARE PROFESSIONAL: Current Use of Large Language Models in the Finnish Software Industry and Future Predictions
Master's thesis
Tampere University
Computing Sciences
January 2025

This study explores the integration of large language models (LLMs) into the Finnish software industry, focusing on their current use, implications for workforce skills, and future impact on job roles. Through qualitative research involving semi-structured interviews with 12 Finnish IT professionals, the study identifies key trends in LLM adoption. It offers insights for training, organizational strategies, and AI adaptation policies.

Using the Gioia method to determine LLMs' current uses and reflexive thematic analysis to explore future implications, the analysis revealed four primary dimensions of LLM usage: programming-related tasks, non-programming technical activities, self-directed learning, and teamwork. These findings suggest LLMs are transforming workflows, enhancing automation, and reducing the demand for repetitive roles.

Participants highlighted the rapid evolution of LLMs, including improvements in intelligence, multimodality, and capacity. As these models advance, the industry anticipates significant changes in the nature of jobs, with LLMs taking over more complex tasks. While some predict workforce reductions, others foresee the creation of new roles. Software professionals were expected to need a broader set of skills, ranging from technical skills to communication and interdisciplinary knowledge.

These findings underscore the importance of self-directed learning and adaptability in navigating this shift. Recommendations include further research on human-AI collaboration, increased focus on lifelong learning initiatives by businesses, and the development of educational programs tailored to AI-driven roles. These actions are crucial for minimizing cultural lag and ensuring the workforce remains prepared for AI-driven changes.

Keywords: [Large Language Models, Software Industry, Digital Transformation, Cultural Lag]

TIIVISTELMÄ

Vilma Karppinen: LARGE LANGUAGE MODELS – THE BUILDING BLOCKS OF A SOFTWARE PROFESSIONAL: Current Use of Large Language Models in the Finnish Software Industry and Future Predictions
Pro gradu -tutkielma
Tampereen yliopisto
Tietojenkäsittelytieteiden tutkinto-ohjelma
Tammikuu 2025

Tässä tutkimuksessa tarkastellaan suurten kielimallien (LLM) integrointia suomalaiseen ohjelmistoteollisuuteen keskittyen niiden nykyiseen käyttöön, sekä niiden tuomiin vaikutuksiin työvoiman osaamiseen ja työtehtäviin. Kvalitatiivisella tutkimuksella, jossa haastateltiin 12 suomalaista IT-ammattilaista, selvitetään keskeisiä suuntauksia LLM:ien omaksumisessa. Tutkimus tarjoaa näkemyksiä koulutukseen, organisaatiostrategioihin ja tekoälyn mukauttamiskäytäntöihin.

Tutkimus hyödynsi Gioia-menetelmää LLM:ien nykykäyttöjen kartoittamiseen ja refleksiivistä temaattista analyysiä tulevien vaikutusten tutkimiseen. Analyysi paljasti neljä keskeistä ulottuvuutta LLM:ien käytössä: ohjelmointiin liittyvät tehtävät, muut tekniset ei-ohjelmointitoiminnot, itseohjautuva oppiminen ja tiimityö. Tulokset viittaavat siihen, että LLM:t muuttavat työnkulkuja, lisäävät automaatiota ja vähentävät toistuvien tehtävien tarvetta.

Tulokset korostivat LLM:ien nopeaa kehitystä, mukaan lukien älykkyyden, multimodaalisuuden ja kapasiteetin parantuminen. Mallien kehittyessä alalla odotetaan merkittäviä muutoksia työn luonteessa, kun LLM:t ottavat kontilleen monimutkaisempia tehtäviä. Toiset ennustavat työvoiman vähentämistä, kun taas toiset ennakoivat uusien roolien luomista. Ohjelmistoammattilaisten odotettiin tarvitsevan laajempaa osaamista teknisistä taidoista kommunikaatioon ja monitieteelliseen osaamiseen.

Kaiken kaikkiaan tulokset korostavat itseohjautuvan oppimisen ja mukautumiskyvyn merkitystä teknologisen muutoksen navigoimisessa. Suosituksia ovat lisätutkimus ihmisen ja tekoälyn välisestä yhteistyöstä sekä LLM:ien vaikutuksista muihin sektoreihin, yritysten elinikäisen oppimisen aloitteiden lisääminen sekä tekoälyohjattuihin rooleihin räätälöityjen koulutusohjelmien kehittäminen. Nämä toimet ovat ratkaisevan tärkeitä kulttuurisen viiveen minimoimiseksi ja sen varmistamiseksi, että työvoima on valmis tekoälyn ohjaamiin muutoksiin.

Avainsanat: Suuret Kielimallit, Ohjelmistoala, Digitaalinen Transformaatio, Kulttuurinen Viive

Table of Contents

1 Introduction

The field of information and communications technology (ICT) is undergoing a rapid transformation due to advancements in artificial intelligence (AI), particularly in large language models (LLMs) such as OpenAI's GPT series and Google's Gemini (Brown et al., 2020; Raffel et al., 2020). These models, capable of generating human-like text and automating complex tasks, have become powerful tools in various professional domains, including the software industry. As LLM technology evolves, critical questions arise about how software professionals integrate such models into their daily work and how future developments may reshape the industry. However, adapting to technological change is not instantaneous. Organizations and individuals in the software industry often face a lag between the availability of advanced technologies and their integration into professional practices (Ogburn, 1957; Maruping & Matook, 2020). Understanding how LLMs are currently used and their future implications is crucial for minimizing this cultural lag and ensuring a smooth technological transition. Nevertheless, existing research predominantly focuses on the technical capabilities of LLMs, leaving a significant gap in understanding their socio-economic impact on the IT labor market.

Bridging this research gap is essential. As LLMs continue to advance, they have the potential to redefine job roles, from automating routine coding tasks to assisting in creative problem-solving and software design. Addressing these changes requires a comprehensive examination of how professionals currently use LLMs and how they anticipate future developments will affect their roles. This understanding is key to informing educational programs, workforce training initiatives, and policy-making efforts aimed at fostering sustainable employment in the Finnish IT sector. Accordingly, the following research questions are addressed:

RQ1: How are software professionals currently using LLMs to assist them in their work?

RQ2: What are the possible future implications of developments in LLM technology on workforce needs and skills in the Finnish software industry?

In-depth interviews were conducted with IT professionals actively using LLMs to address these questions. The analyzed results were connected to existing theories, such as cultural lag theory (Ogburn, 1957) and AI adoption frameworks (Sauvola et al., 2024), which were applied to understand how the adoption of LLMs influences professional practices, skillsets, and workforce dynamics within the software industry. Through this approach, contributions are made to the ongoing discussion about the future of work in the context of AI-driven technological change.

The remainder of this paper is structured as follows: First, the relevant literature on the ICT and software industry in Finland, along with LLM technology and digital transformation, is reviewed. Next, the research methodology is outlined. Findings from the empirical analysis are then presented and discussed, together with implications for IT professionals, companies, educational institutions, and policymakers. Finally, recommendations for future research and conclusions are provided.

2 Theory

2.1 ITC industry in Finland and what software personnel do

The ICT sector in Finland has become a key contributor to the national economy, acting as a driver of innovation and growth. Characterized as a rapidly expanding field, the sector plays a crucial role in shaping the country's technological and economic development (Lukkarinen & Tuomaala, 2024; Simonen et al., 2016). The rise of digitalization has boosted the economic significance of ICT, attracting considerable investment, particularly in Finnish startups, which have secured substantial venture capital in recent years (Aalto & Mikkola, 2023). Moreover, innovations from the ICT sector are affecting traditional industries through software solutions and automation developments, which are being integrated into manufacturing and other industrial processes (Lukkarinen & Tuomaala, 2024). Finnish companies within the ICT sector are recognized as pioneers in technological innovation, with the country boasting one of the highest proportions of ICT employment in Europe (European Commission et al., 2023). In 2022, the sector employed 6.9% of the private sector workforce, with a 15.3% increase in employment since 2018 (Suomen virallinen tilasto (SVT), 2024).

Software development has emerged as a particularly critical component within this growing sector, accounting for a large portion of the industry's overall value. The software sector is a key pillar of Finland's ICT landscape, with over 80,000 people employed directly in software-related roles within pure IT companies alone (Suomen virallinen tilasto (SVT), 2024). Software design and production represent one of the fastest-growing areas of the Finnish economy and a fundamental driver of technological progress across various industries (Lukkarinen & Tuomaala, 2024). Many other sectors also depend on software professionals who handle software transitions, integrations and maintenance of business-critical programs. Understanding the role of software developers within this context and how they contribute to the broader Finnish economy is essential for assessing the current state and future trends of the ICT industry in Finland. When examining the role of the software industry in Finland, it becomes evident that the work of software developers extends beyond technical expertise. Their responsibilities cover a broad range of skills, tasks, and duties essential to software systems' production and maintenance. This complex nature of software development, which demands a combination of hard and soft skills, is also documented in the literature. A study found that the main activities of developers were programming (21% of the time), reading and writing emails (14% of the time), and doing work-related web browsing (11.4% of the time) (Meyer et al., 2017). Other activities included planning meetings and work items, reading and writing documentation, and having meetings (Meyer et al., 2017). According to Begel and Simon (2008), the duties of software workers can be categorized into eight distinct areas: programming, debugging, testing, project management, documentation, specifications, tools, and communication. Ordered by frequency of actions, each category represents essential aspects of a developer's workflow.

According to Begel and Simon (2008), the first category, programming, involves reading, producing, and reviewing code. This is a foundational aspect of a software developer's role, as it pertains directly to creating functional software. The second category, working on bugs, focuses on identifying and resolving errors in the code through reproducing, reporting, and debugging. Testing, the third category, involves writing and running tests to ensure that the software operates as expected. It is closely tied to programming and bug-fixing tasks, as testing the code often unearths hidden bugs. Project management, the fourth category, consists of version control tasks like checking out code for development and checking in completed work so that updates are tracked and integrated smoothly within the team. Documentation, the fifth category, pertains to searching, reading, and writing project-related documentation, ensuring that all aspects of the development process are clear and accessible to other team members. Specifications, the sixth category, involves the design process, including reading and writing technical specifications for the software to ensure that the product meets its functional requirements. Tools, the seventh category, refers to the discovery, installation, and use of software tools that facilitate development tasks. This could involve using integrated development environments, version control systems, and debugging tools. Finally, the eighth category of communication encompasses all forms of interaction between developers, team members, managers, and other stakeholders. Effective communication is crucial for coordinating efforts, resolving issues, and ensuring alignment between various teams and departments involved in a project.

Capretz and Ahmed (2010) build on these categories by diving into the specific tasks that software developers engage in, such as analyzing business requirements, preparing detailed programming specifications, and implementing modular programs. They state that these technical tasks also connect to and require robust soft skills. For example, the ability to analyze business requirements and prepare system specifications requires not only technical proficiency but also attention to detail, strong analytical abilities, and problem-solving skills (Capretz & Ahmed, 2010). These soft skills are indispensable for translating complex business needs into functional software solutions. Capretz and Ahmed (2010) identify several critical soft skills that software developers must possess. These include communication and interpersonal skills essential for collaborating with team members and stakeholders. Developers must also be open and adaptable to change, as the fast-paced nature of the software industry often requires them to pivot and adjust to new tools, methodologies, or project goals. Organizational skills are crucial for managing multiple tasks and meeting deadlines, while the ability to work independently is necessary for completing tasks without constant supervision. Finally, being a team player is vital for fostering collaboration and ensuring that all development team members are aligned toward a common goal. Overall, software developers must balance technical acumen with a variety of soft skills to excel at translating technical knowledge into practical solutions that align with business requirements.

These findings resonate with Lethbridge's (2000) study, which evaluated the gaps between formal education and on-the-job learning for software professionals. Lethbridge's research highlights the disconnect between the skills emphasized in educational programs and those required in real-world software development. The study found that while programming languages and data structures were extensively covered in education, critical skills such as software management, people skills, and business knowledge were underrepresented. In contrast, these soft skills are essential for developers, particularly in project management, negotiation, human-computer interaction, and leadership, where significant gaps exist between educational preparation and professional practice. Lethbridge (2000) points to the importance of learning on the job, especially in configuration and release management, project management, and quality assurance, which are areas often neglected in formal education. These insights underscore the need for universities to address the educational gaps in software processes and soft skills, which are vital for translating technical knowledge into effective software solutions that meet business needs.

Multiple thesis papers from students at Finnish universities of technology were reviewed to gain a deeper understanding of how these technical and soft skills are visible within the Finnish software landscape. While these papers may sometimes lack the depth or rigor of professional studies, this approach offers valuable insights into real-world experiences and perspectives of emerging professionals in the field. A key advantage of using student theses is that they often include current case studies and practical examples of working ICT workers. The roles within software development, as demonstrated by the works of Seppälä (2024), Mäkinen (2021), Turunen (2022), Porkka (2018), and Matinpuro (2021), highlight a range of responsibilities across different levels of expertise and specialization. Full stack developers, like Seppälä (2024), work with frontend and backend development using technologies such as React and Java Spring Boot while managing databases, performing software testing, and engaging in problem-solving and version control. Similarly, programming interns like Mäkinen (2021) focus on frontend development with HTML and JavaScript while learning additional tools their employer uses. More experienced developers, like Turunen (2022) and Porkka (2018), described their experiences in backend development with C# and Java, debugging systems based on submitted tickets, and interacting with testers to ensure software quality (Porkka, 2018; Turunen, 2022). Effective communication, both in requesting help from coworkers and facilitating team collaboration, is a recurring theme in all the theses. Its importance is especially evident in roles requiring constant interaction with colleagues, testers, and project managers (Matinpuro, 2021; Porkka, 2018; Turunen, 2022). Moreover, all thesis writers reported the need for continuous learning, whether it involved mastering new programming languages or tools, highlighting the dynamic nature of software development as well as the varying needs of employers. Each author worked independently on their assigned tasks within the broader framework of team-based contributions to the final product, reinforcing the sense of balance between autonomy and collaboration in modern software development workflows (Mäkinen, 2021; Matinpuro, 2021; Porkka, 2018; Seppälä, 2024; Turunen, 2022).

The findings from both Scaffidi (2018) and Exter (2014) align closely with the observations gathered from the thesis works of Seppälä (2024), Mäkinen (2021), Turunen (2022), Porkka (2018), and Matinpuro (2021). They also emphasize the importance of combining technical and soft skills in software development. Scaffidi's (2018) research identifies key technical areas such as web development, relational databases, and source code management, which are mirrored in the thesis papers, where developers commonly utilized technologies like JavaScript, React, and version control tools such as Git (Mäkinen, 2021; Porkka, 2018; Seppälä, 2024). Agile methodologies were frequently applied in both the research findings and the thesis works, showcasing the essential role of collaboration and communication within teams. This is especially evident in the experiences of Matinpuro (2021) and Porkka (2018), who emphasized the necessity of teamwork and effective communication to ensure the success of software development projects in a fast-paced environment. Scaffidi's (2018) emphasis on soft skills, such as handling ambiguity, learning quickly, and solving complex problems, also aligns with the developers' experiences. In particular, the importance of problem-solving and seeking coworker advice were recurring themes in the thesis papers, where collaboration and peer support played a critical role in overcoming technical challenges (Mäkinen, 2021; Turunen, 2022). Exter's (2014) findings further reinforce these insights by underscoring the significance of practical skills and foundational knowledge in the workplace. They argue that while programming languages and technical proficiencies are necessary, understanding core concepts such as problem-solving and critical thinking is paramount. This notion is echoed in the thesis works, where authors like Seppälä (2024) and Mäkinen (2021) focused more on problem-solving, continuous learning, and improving their skills. Exter (2014) also highlights the necessity of self-directed learning. This sentiment resonates in the thesis works, where developers had to frequently teach themselves new skills and technologies they had not been taught in their respective institutions (Porkka, 2018; Turunen, 2022). Additionally, both Exter (2014) and the thesis authors stress the importance of teamwork and communication in modern software development environments. As Exter (2014) notes, professionals must not only perform multiple roles but also communicate effectively across disciplines, a theme echoed in the thesis papers, where collaboration with colleagues and asking for assistance were essential elements of daily work (Matinpuro, 2021; Turunen, 2022).

2.2 Change in the nature of work and digital transformation

Digital technologies have also significantly impacted labor markets, particularly across the Organization for Economic Co-operation and Development (OECD) countries, where automation and technological advancements have transformed job roles and industries. Berger and Frey (2016) claim that, ever since the computer revolution, technology has substantially increased the demand for cognitive skills while reducing the need for workers to perform routine tasks. Automation has contributed to declining manufacturing employment and the reallocation of labor to service industries. This shift mirrors the findings of Cascio and Montealegre (2016), who describe how technology has reshaped work processes by altering nonrelational roles, tasks, and interactions within work systems. For example, while many traditional jobs—such as bank tellers—have seen significant changes due to technologies like ATMs, new tasks like customer relationship management have become essential. Thus, even though automation has replaced certain occupations, it has created new ones, including software engineers and database administrators (Berger & Frey, 2016; Cascio & Montealegre, 2016). While automation has eased more laborious roles, the addition of technology has altered the way workers interact with both technology and each other, calling for more well-refined soft skills in the workplace (Capretz & Ahmed, 2010; Cascio & Montealegre, 2016; Exter, 2014). This digital transformation described in both studies expands in all aspects of business. Digital transformation extends beyond the mere adoption of new technologies; it encompasses significant shifts in how organizations operate, driven by advancements in digital technologies such as machine learning, deep learning, blockchain, and cloud computing (Akter et al., 2022; Brock & von Wangenheim, 2019; Gorenek & Kohont, 2019). This process includes a transition from traditional business models to new, digitally enhanced approaches that leverage technologies like mobile, social, and emerging digital platforms to drive innovation and efficiency. In essence, digital transformation is not limited to just technical upgrades; it requires changes in leadership, management practices, and organizational culture.

The shift toward digital transformation also affects the nature of software development and design work. Recent studies in software engineering have shown how emerging technologies are continuously altering software development paradigms, changing the composition of software and the methodologies through which it is created (Wiedemann et al., 2020). The demand for expertise in cutting-edge technological roles is also visible in the job market, where these skill requirements were identified in an increasing number of job listings (Gurcan & Cagiltay, 2019). Consequently, the software industry is constantly changing, driven by ongoing technological advancements and innovations, reshaping the processes of software development and the roles required within the industry (Maruping & Matook, 2020). This dynamic nature of the industry highlights the importance of keeping pace with technological megatrends to anticipate their far-reaching implications, both within the software industry and beyond (Pappas et al., 2018). The software industry thus acts as both a key contributor to and a beneficiary of digital transformation processes (Vial, 2019), emphasizing the reciprocal relationship between technology development and industrial change.

The increasing influence of digital transformation in the software industry has paved the way for significant technological advancements. One of the most profound developments has been the rise of large language models such as Gemini by Google and GPT-4 by OpenAI. These generative AI models, capable of understanding and producing human-like text, have emerged as powerful tools for automating tasks like code generation, debugging, and content creation. Their integration into the software development process also represents a continuation of digital transformation, where automation and AI technologies are transforming how work is done and the very nature of roles within the industry. Since the release of GPT-3 in 2021 and GPT-4 in 2023, the use of LLMs in companies has accelerated, enabling companies to streamline operations, increase productivity, and shift skilled workers towards more complex and strategic tasks (Ebert & Louridas, 2023). Tools like GitHub Copilot and AWS CodeWhisperer exemplify how AI provides real-time code suggestions and automates repetitive tasks, freeing developers to focus on more complex and creative aspects of their work. Multiple studies show that these tools can improve efficiency and flow by optimizing time and minimizing disruptions, leading to higher developer satisfaction and better performance (Coutinho et al., 2024; Cui et al., 2024; Hamdi & Kim, 2023; Jaffe et al., 2024; Li et al., 2023; S. Peng et al., 2023). A series of studies funded by Microsoft also found that the users of AI-based tools found their work more enjoyable and had a lighter cognitive load when doing familiar tasks. However, in some cases, using AI tools comes with a trade-off of lower-quality output when employee productivity increases (Cambon et al., 2023). In software professionals, however, workplace satisfaction was not affected by these tools, but the usage of tools resulted in 36%-time savings when performing familiar tasks (Jaffe et al., 2024).

Since the successful release of GPT-3 by OpenAI, the market has exploded, with multiple large operators like Google, Meta, and Amazon entering the market with their own LLM-based AI products (Hadi et al., 2023). This fast-paced development of AI models has also inspired predictive research on what LLMs might be able to do in the future. Sauvola et al. (2024) outline various possible future scenarios of AI involvement in the software development process, ranging from traditional human-led operations to scenarios where AI plays a dominant role in managing development tasks. These scenarios illustrate a trajectory where AI gradually assumes more responsibilities, moving from task automation to complete development process management, with human oversight primarily focused on complex problem-solving and quality assurance. While LLMs have already shown promise in creating functionally correct code (Liu et al., 2024) and process automation (X. Peng et al., 2019; Saklamaeva & Pavlič, 2024), there seems to be an increasing demand for automatization among software developers. When asked, software developers have reported wanting to use AI-based tools for automating routine tasks, like writing documentation and generating unit tests, thus improving the developer experience (Khemka & Houck, 2024). Some also reported wanting tools to use as a partner for finding defects in the code while programming (Khemka & Houck, 2024). These findings demonstrate the need for AI-based tools in the software industry and the willingness of employees to adapt them to their work, giving them more time away from tedious, repetitive tasks.

While technology and LLM-based solutions and applications may continue to develop rapidly, the institutions and law will lag behind. This kind of cultural lag happens when material culture, such as scientific progress or technological advancements, develops faster than the values and beliefs of society (Ogburn, 1957). Along with ideologies and morals, these beliefs and values are part of the non-material culture, which, according to the cultural lag theory, will continuously develop slower than material culture and must, in time, meet the changes set by the material culture (Ogburn, 1957; Volti, 2004). The lag theory's basis is that institutions, such as the educational and legal systems and governments, are often slow to change. Society might also take time to accept changes and adapt to new things due to innate beliefs or old practices. On the other hand, companies are slow to adapt their methods and respond to changed demands in skills (i.e., hiring new people) (Ogburn, 1957). Because the advancements in LLMs are relatively recent, we can expect the non-material culture to adapt to LLM technology sometime in the future. However, while slow, the non-material culture has already shown signs of trying to catch up with the evolving LLM technology; the EU has come up with laws and regulations concerning data privacy in training LLMs (European Data Protection Supervisor, 2023), and Finnish universities- such as Tampere University, Helsinki University and Aalto University of Technology- already provide several courses focused on building or applying AI for different solutions. For example, the bachelor’s program in computer science at the University of Helsinki includes mandatory and optional LLM-related courses. While the mandatory courses are mainly introductory, the optional courses include studies on AI and society, building AI, and the ethics of AI. Tampere University, on the other hand, has courses on human-centered aspects of human-AI interaction, such as how to design for a conversational AI model and consider the societal perspectives of LLMs(Tampere University, 2024b, 2024a), as well as how to utilize AI models for information search and translation (Tampere University, 2024c).

3 Methodology

This study employed a qualitative research approach to investigate the current utilization of LLMs among IT professionals in Finland and explore the potential implications of LLMs on the IT industry's methodologies and workforce needs. The methodology consisted of expert interviews conducted through a semi-structured interview technique. The interviews were recorded, transcribed, and analyzed using the Gioia method. The coming sections will explain in more detail why interviews were chosen as the primary data-gathering method, how participants were recruited for the study, how the interviews were conducted, and how the data was transcribed and analyzed.

3.1 Reasoning behind the data gathering method

The choice of semi-structured interviews as the primary data-gathering method for this research was influenced by several key factors. Firstly, interviews were more straightforward to organize than other methods, such as observational and diary studies, which require extensive planning, coordination, and participant commitment. Interviews could be arranged by email, asking for the respondent's availability, and could be rearranged at a moment's notice if needed. This flexibility was particularly advantageous given the time constraints many potential interviewees face due to their busy project schedules.

Secondly, the interview structure could be designed to include questions about both research questions. Exploring both questions with one method streamlined the data-gathering process. Moreover, interviews allowed for a natural exploration of the participants' experiences and perspectives, and the semi-structured format provided the opportunity to ask follow-up questions and clarify participants' responses.

Thirdly, alternative methods such as focus groups or the Delphi method were considered impractical due to potential participants' varying schedules, making coordinating a convenient time for all experts involved challenging. Similarly, a diary method was ruled out as it would have been difficult to recruit participants willing to maintain a diary over an extended period, and the method would have only addressed the first research question.

During the recruitment process, it became apparent that many potential interviewees faced time constraints due to their busy project schedules. Some also voiced concerns about sharing detailed information about their work and disclosing company secrets. Interviews offered flexibility in length and pacing, allowing for adjustments based on the participants' availability and comfort levels. The semi-structured format also encouraged participants to discuss sensitive topics at their discretion, ensuring a balance between gathering valuable insights and respecting participants' boundaries.

Lastly, the author's familiarity and experience conducting interviews played a role in selecting this method, allowing for a more natural and conversational interaction with the participants. In summary, using interviews as the primary data-gathering method was driven by practical considerations, familiarity with the method, and flexibility in accommodating participants' schedules and preferences.

3.2 The interview frame

The construction of the interview frame was an iterative process. The goal of the interview was to make the participants comfortable in the interview setting and encourage speculation and critical thinking. The initial draft of the interview question was formulated based on the author's prior knowledge and initial conceptions. These questions were then divided into more specific and refined questions, leading to 17 interview questions. The frame was then constructed by organizing the questions by topic, starting the interview with specifications about the participants and their company's policy with LLMs, and moving on to their personal experiences with LLMs. Lastly, the interview questions guided the interviewee in speculating about the future of LLMs and their effect on the IT industry. The interview concluded with a chance to append or extend their answers and ask questions. The interview frame can be found in Table 3 (Appendix A).

The interview questions were piloted with two individuals from the author's academic network. This phase served the dual purpose of refining the interview protocol and confirming its suitability for subsequent use. After the interviews, the pilot participants were asked for their feedback on the research questions and to describe the interview experience. Feedback from the pilot phase led to the incorporation of one additional question.

3.3 Sampling technique

To ensure that the interviewed participants had the experience and expertise to provide insight into the research questions, a set of criteria was developed for participant sampling. First, the participants had to work in the software industry for at least half a year. Second, the participants had to work either in a Finnish company or their company had to have an active office in Finland. Third, the participants had to be familiar with LLMs. The last criterion was formulated to ensure that the participants would understand how LLMs worked and how they could be used.

After the pilot phase, the snowball sampling technique was followed to recruit participants for the study. This procedure started by considering possible personal connections that would fill the sampling criteria or know someone who would. These potential participants were then contacted and invited to a one-on-one interview with the study's author. After each interview, participants were invited to recommend colleagues or acquaintances who might offer valuable insights, thus initiating a snowball effect in participant recruitment. These actions fostered the snowball effect, which allowed a wide range of views and ideas to be captured. The snowball method's iterative nature also facilitated the sample size's organic expansion until saturation. Saturation was estimated to be reached when two consecutive participants failed to give more than two new perspectives (or codes as used in the analysis) on the questions.

3.4 Participants

Because the two pilot interviews fulfilled the sampling criteria, and their interview structure remained similar to the final form of the interview protocol, their interview data was included in the final dataset.

The study included 12 participants aged 27 to 52 (Table 1). All but one of the participants presented as male. The sample of participants included diverse roles in the software industry, ranging from software developers to project managers, with varying experience levels in the field. One participant did not work in the software industry but was nonetheless included in the analysis due to the limited number of participants. Detailed information regarding each participant, including their job title and years of IT experience at the time of the interviews, can be found in Table 1. The descriptions of their respective employers are provided in Table 2. All but one of the employers were relatively large companies with over 1000 employees. The one smaller company (Company B) still employed several hundred employees at the time of the interviews.

Table 1. Participant information.

	Occupation	**Work experience**	**Company description**
Participant 1	Junior Designer	8 months	B
Participant 2	Software Engineer	12 years	A
Participant 3	cybersecurity specialist	2 years	C
participant 4	Dev. Ops team lead	25 years	A
Participant 5	Lead Software Engineer	25 years	A
Participant 6	Software Architect	9 years	D
Participant 7	Software Developer	6 years	A
Participant 8	Director of Product Development	20 years	E
Participant 9	Director of data analytics and integrations	20 years	B
Participant 10	Research Assistant	7 months	F

Participant 11	Back-end Software Developer	5 years	G
Participant 12	Software Developer	3 years	G

Table 2. Company Descriptions.

Company	Description
A	An international enterprise that offers a multitude of digital services, specializing in digitalization and information management
B	A software house that offers cybersecurity, data, and integration services to companies and institutions. It is smaller than the other included companies.
C	A technology company that offers software, cybersecurity, and automation services on an industrial level
D	A company in the building and real estate industry, specializing in improving productivity, energy efficiency, sustainability, and digitalization
E	An international company providing automation services on a multitude of retail processes.
F	A University
G	Multinational information technology company,

3.5 Interview process

The interview process was conducted with careful attention to ethical considerations and participant confidentiality. The interviews were conducted in Finnish or English, depending on the interviewee's language skills. Depending on the participants ' preferences, they took place in a university meeting room or via Microsoft Teams.

Before the interview, participants were provided with a privacy statement detailing the collection, storage, and usage of their information for research purposes. This privacy statement aimed to ensure transparency and informed consent regarding handling participants' data throughout the research process. The interview sessions started with a brief introductory speech outlining the purpose and structure of the interview and informing participants about their rights as research participants. The participants were then informed that the audio recording would begin, and their consent for audio recording was explicitly obtained at the beginning of the recording session.

Participants were given the opportunity to ask questions or seek clarification on any aspect of the research process both before and after the recording session. The interviews lasted from 10 to 40 minutes, depending on the depth of discussion and the participants' insights on the topics at hand.

After the interviews, the interview recordings were transcribed. During the transcribing process, only the segments relevant to the research questions were retained for analysis. This ensured that the data analyzed were directly aligned with the study's objectives and that no sensitive information continued its way to analysis. The interviews lasted from 10 to 40 minutes, depending on the depth of discussion and the participants' insights on the topics at hand.

3.6 Data Analysis

The data analysis was divided into two sections due to the nature of the two research questions. The Gioia method (Gioia et al., 2013) was deemed appropriate for exploring the current trends in the usage of LLMs, but the complexity of the second research question called for a more nuanced method of analysis and a narrative that the Reflexive Thematic Analysis (Braun & Clarke, 2019) could offer. In the following subsections, the rationale of selecting each method is better explained, accompanied by a detailed explanation of the step-by-step processes of each analytical approach.

3.6.1 RQ1: The Gioia Method

The data analysis process for the first research questions was guided by the Gioia method (Gioia et al., 2013). While acknowledging the challenge of researcher subjectivity inherent in the theorizing process (Mees-Buss et al., 2022), the Gioia method provides a systematic and transparent approach to theory development, offering guidelines for maintaining qualitative rigor and minimizing the influence of individual biases. It was chosen for its systematic approach to coding data and visualization of the analysis process.

The analysis proceeded through four stages, adhering to established qualitative coding methodologies.

Phase 1: Open coding. The initial analysis phase involved open coding (Strauss & Corbin, 1998), where each transcript was carefully read to identify initial codes reflecting participants' discussions and possible underlying meanings. In the case of this study, the codes were answers to the question, "*How does this participant use LLMs in their work*?" The answers, thus the codes, were direct use cases derived from the quotes. For example, based on the quote, "I most often go on ChatGPT or Bard and ask specific questions about certain technologies that I am not familiar with" (P2), a code "Learning about unfamiliar technologies" was created. Table 4 (Appendix B) records these codes and their respective quotes. Each instance of information related to the research questions was picked from transcribed text and placed as a note on a Miro board with the participant's ID number.

Phase 2: Axial coding. In the next phase, axial coding (Strauss & Corbin, 1998) was done to identify similarities in the open. Combining codes similar in content or with logical connections reduced the number of overall codes, and broader categories were formed. The codes were combined into groups based on similarity. Each group was then reviewed and given an overarching theme. This process was straightforward as the classifications were derived from the first phase codes.

Phase 3: Selective coding. In the third phase of the analysis, selective coding (Strauss & Corbin, 1998) was used to find the overarching themes in the data. The second-order categories formed in the previous phase were examined for underlying connections. This is how the more abstract aggregate dimensions were found from the data. Considering the first research question, these categories were thought to be linked to different responsibilities and qualities of IT professionals in their work, as described by previous research.

Phase 4: Comparative Analysis. Finally, the analysis's results were compared with relevant literature to explore relationships and identify any new concepts that emerged from the data. This study weighed the findings against studies about IT workers' skills and job descriptions, digitalization, and cultural lag.

3.6.2 RQ2: Reflexive thematic analysis

Reflexive thematic analysis (Braun & Clarke, 2019) was particularly well-suited for analyzing the results of the second research question, which required a combined understanding of how LLMs would develop in the future and how these developments would affect work in the IT industry. Reflexive thematic analysis offered a flexible and iterative approach that allowed for exploring this complex phenomenon while providing a framework for systematically and transparently examining qualitative data.

The analysis consisted of six phases defined by Braun & Clarke (2012, 2019), though the process was not linear but iterative.

Phase 1: Familiarization with the Data. The analysis began with a thorough familiarization with the qualitative data. This was done by re-listening the interviews and re-reading the transcripts to better understand the content and context of the interview data.

Phase 2: Initial Coding. Initial codes were created based on patterns, themes, and key concepts identified within the data. These codes were then extracted to the Miro-board with relevant quotes and participant information.

Phase 3: Generating Themes. Following the initial coding phase, codes were visually organized into preliminary themes that captured the main patterns and concepts within the data. Themes were developed iteratively after each re-visited interview by constantly comparing coded segments and reflecting on their relevance and coherence.

Phase 4: Review and Revision of the Potential Themes. The identified themes were reviewed and refined through a process of iterative analysis. By making different combinations of the initial codes, the themes were revised, merged, or split as necessary to ensure they accurately reflected the content and context of the data. The review of the themes included the following questions, as suggested by Braun & Clarke (2012):

- Is this a theme (or is it just a code)?
- If it is a theme, what is the quality of this theme (does it tell me something valuable about the data set and my research question)?
- Are there enough (meaningful) data to support this theme (is the theme thin or thick)?

- Are the data too diverse and wide-ranging (does the theme lack coherence)?

Phase 5: Defining and Naming the Themes. The formed themes were reviewed, and the key defining points for each theme were formulated in a few short sentences. The themes were defined by the criteria given by Braun & Clarke (2012): (1) The themes should have a singular focus, (2) they should be related but not overlap, and (3) the themes should relate directly to the given research question. Once defined, the themes and subthemes were given names. For this study, the themes were directly related to the different parts of the research question and were named as such. This phase also included choosing extracts from the interview transcripts to present when writing up the themes. The most coherent and argumentative extracts were chosen from the selection of extracts. Table 5 (Appendix C) shows the quotes associated with each theme.

Phase 6: Producing the Report. The final phase addressed the continuous reporting of changes via informal memos and notes on the Miro board. The narrative of the results was also established in this phase by choosing the order in which to report the defined themes. In this study, the reporting order was partially guided by the second research question: reporting the potential developments of LLMs before reporting the impacts of the LLMs on the future workforce needs and skills of the IT industry.

4 Findings on Research Question 1

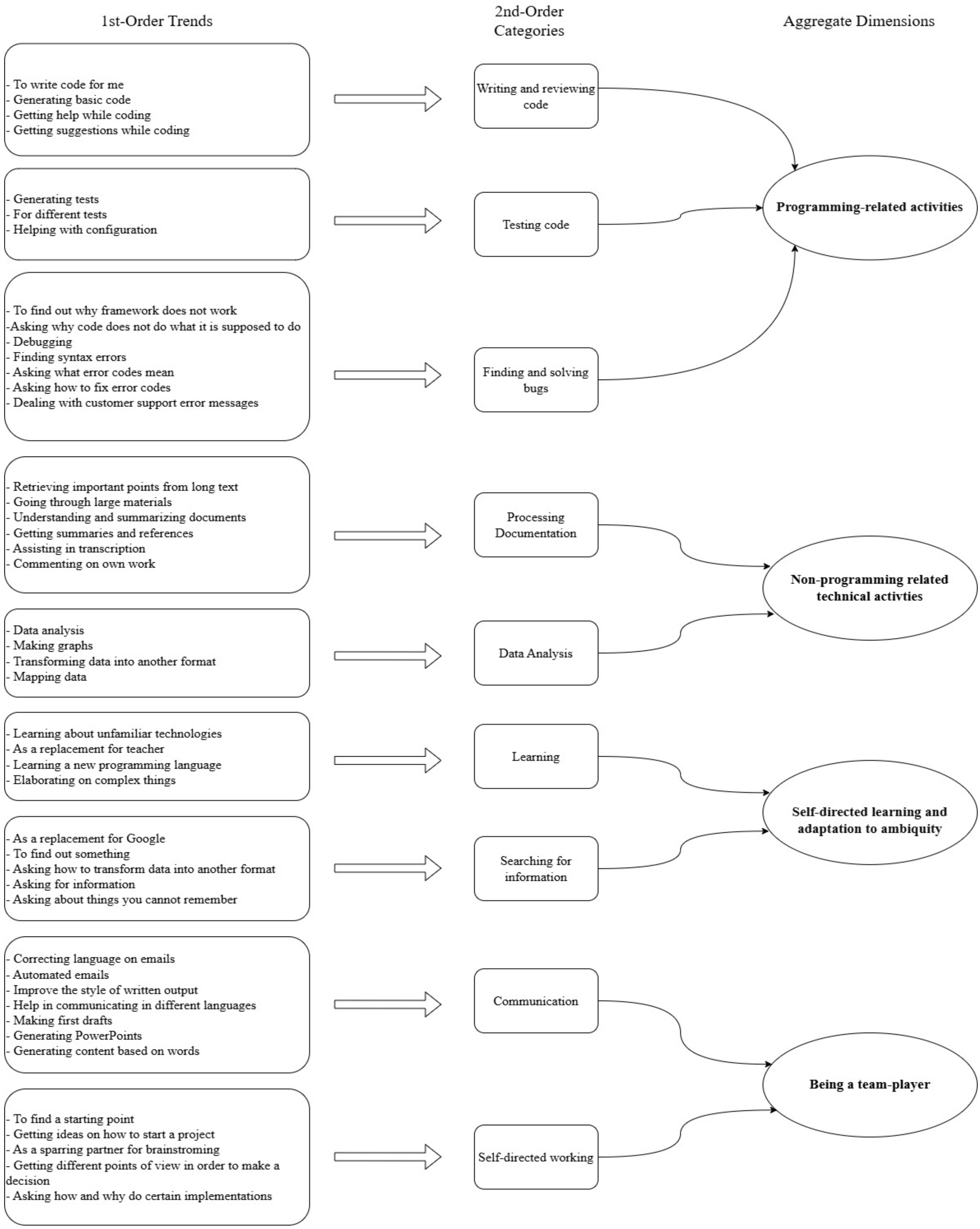

Figure 1. The results of the data analysis for RQ1.

To address Research Question 1, the analysis focused on identifying patterns and themes related to using LLMs among IT professionals in Finland. Four aggregate dimensions were identified as a result of the analysis: Programming-related activities, non-programming-related technical activities, self-directed learning, adaptation to ambiguity, and being a team player. The formation of these aggregate dimensions is showcased in Figure 1.

4.1 Programming-related activities

This subtheme explores using LLMs to streamline programming-related tasks and improve efficiency. Several participants highlighted the utility of LLMs for code generation and automating repetitive tasks, such as generating test cases. Nearly all participants reported using LLMs for coding-related tasks, including producing code, identifying and fixing errors, and generating documentation. Participant 7, who works as a software developer, described LLMs as a tool for accelerating the coding process:

"I last used [LLMs] for basic coding, just to make the work faster [...] Small snippets of code can be obtained using the language model faster than otherwise." (P7)

Specialized tools like GitHub Copilot were mentioned by Participants 2 and 6, who both have several years of experience as software professionals. Their statements highlighted the role of LLMs in assisting programming efforts:

"[When asked about most useful use cases for LLMs] *Code assistance, I'm currently using Copilot to assist in coding, or do the work for me*." (P2)

"In the coding phase, CoPilot helps by giving suggestions on coding." (P6)

LLMs were further reported as valuable for error-solving and testing. Participant 11, who works as a backend software developer, feels that now is the best time to be utilizing LLMs while also discussing the potential downfalls to relying on LLMs:

"I feel that now is the most optimal time to use language models, as a middle model developer, for example, for problem-solving. […] *For example, when I'm looking for frames for certain types of problems that are not already familiar. Admittedly, this is partly a bad way, because verifying a problem fix is difficult, because blindly trusting a fix has its own risks, also in terms of one's own development.* " (P11)

This use case of finding and fixing problems within the code was also reported by Participant 10, a fresh face in the IT industry, and Participant 2, a seasoned professional:

"[When describing most useful use cases for LLMs] *It's really good for debugging, and for finding syntax errors."* (P10)

"When a program error occurs, asking LLMs how to fix specific errors that are not user friendly, but also to diagnose a program." (P2)

While colleagues with over 20 years of IT experience, Participants 2, 4 and 5 had different uses for LLMs while working. Participant 4 describes using the LLM tools for producing test cases and automatizing their work processes, while Participant 5 seeks help in feature implementations:

"[When listing use cases for LLMs] …*even up to generating test cases for your code, and just throwing in some ideas like how to start a project is a common way to use it."* (P4)

"[When describing most useful use cases for LLMs] *When we need to implement a feature in some configuration, help is sought from language models to make the software compatible."* (P5)

Overall, participants found LLMs helpful for tasks related to Task automation and enhancement of work tasks, such as coding, problem-solving and testing.

4.2 Non-programming-related technical activities

The second aggregate dimension focused on other technical activities in which IT workers utilized LLMs. The work activities covered by this aggregate dimension are data analysis and processing documentation, which require technical knowledge. Three participants mentioned using LLMs directly for data analysis, while two others referred to knowing people who used LLMs for this task. Participant 10, who handles lots of data in his work as a research assistant, highlighted the use of LLMs for reformatting data:

"If the data is in one format and you want it in another format, the language model will do it for you." (P10)

Participant 9, who works as a director of data analytics and integration, was thrown away by the effectiveness of using LLMs for data analytics, particularly in terms of visual representation:

"The analytics function has been a WOW factor. For example, you can ask GPT to make a graph and analyse results." (P9)

Participants discussed the potential usefulness of LLMs for navigating and comprehending large texts or documentation. While listing their use cases for LLMs, some participants, like Participants 1 and 3, both rather recent additions to the IT field, mentioned using LLMs for processing larger documents:

"[When listing uses for LLMs] *For retrieving important points from a long text. "*(P1)

"[When listing uses for LLMs] *For reviewing of large materials."* (P3)

In addition to summarizing documents using LLMs, Participant 6, a software architect of several years, reported putting their documents in the language model:

"[When listing cases in which LLMs are useful] *To understand and summarize documents.* [...] *Putting your own documents in the language model."* (P6)

4.3 Self-directed learning and adaptation to ambiguity

The third dimension explored using LLMs for information retrieval, learning, and managing large amounts of data. Four participants mentioned using LLMs for learning, while five reported using them to process and summarize long texts. Participant 1, who is still completing their bachelor's while working, highlighted using LLMs as a replacement for a teacher:

"I also use LLMs for learning. I can ask it questions and clarifications instead of a teacher. To understand better." (P1)

Participant 6, on the other hand, had realized the potential of LLMs for self-learning. They talked of a new way of learning and how fast it can be done with LLMs:

"I've noticed that if a new instruction notification language comes along, you don't need to read anything, you can start from a blank slate and talk to that language model, and you can get things done surprisingly quickly. What should have been learned earlier in the course or through articles. Now all you have to do is read what the tool is and where you download it and start talking with it." (P6)

Participants 4 and 5, seasoned colleagues from a large IT house, also brought up the advantages of using LLMs over traditional search engines for more direct and concise information retrieval:

"Much better than Google, as it puts [the information] *to the point. If you have very complex things to elaborate, it can guide you the way."* (P4)

"If something needs to be found out, then ChatGPT is surprisingly quick at producing answers." (P5)

Participant 2, a lower-level employee of the same company, also reported using LLM tools to search for information, especially about new technologies.

"I most often go on ChatGPT or Bard and ask specific questions about certain technologies that I am not familiar with." (P2)

These citations together paint a picture of convenience, using tools to make information searches faster and more to the point. While not all participants mentioned searching for information with LLMs, none mentioned receiving bad or false information from them.

4.4 Being a team player

This aggregate dimension explores the use of LLMs in activities related to communication tasks and self-directed working. Several participants reported using LLMs to ease their communication processes and assist with generating written content. Participant 1, in their work as a junior designer, produces PowerPoint presentations and uses LLMs to help with them:

"[When listing most useful use cases for LLMs] *PowerPoint generators for Office."* (P1)

Participant 9, who works in an executive-level position, on the other hand, used LLMs to transcribe project meetings:

"[When asked about the last project in which LLMs were used to aid the work] *Artificial intelligence helped with transcription during the project. "*(P9)

Another important use case for LLMs was helping with communication, particularly in improving written output. Participant 3, whose work as a cyber security specialist involves discussing specifics with the clients, for instance, noted that LLMs helped handle communication across languages, especially when interacting with a diverse customer base:

"If there are a lot of customers and you have to communicate in different languages, check the dialects. Just in daily communication." (P3)

Participants 8 and 12 echoed Participant 3's sentiment as they also spoke about using LLMs for written communication, especially in crafting professional emails. Participant 8 noted that LLMs help speed up the process and improve the quality of written messages:

"*I might correct the language in emails in different languages. I speed up and improve the style of the written output. For example., I want to say these things in this style."* (P8)

Several participants emphasized the role of LLMs in supporting brainstorming and idea generation, particularly in the early stages of projects. Participant 8 referred to LLMs as a “sparring partner” for brainstorming and exploring different viewpoints to make decisions:

“It's a good sparring partner for brainstorming or to get different points of view to help make a decision.” (P8)

Participant 9 also used LLMs for ideation, mainly related to marketing ideas:

“[When describing most useful use cases for LLMs] *Searching for ideas, e.g., related to marketing.”* (P9)

Furthermore, several participants noted using LLMs to help generate and refine project ideas. Participant 10 referred to using LLMs for reflective thinking when starting new tasks or projects:

“For reflection, etc., for starting things and work.” (P10)

LLMs were also used for refining ideas. Participant 6 described using LLMs to discuss and evaluate various ideas, particularly when considering technical design patterns:

“You can discuss ideas with it. In the conversational model, I ask, for example, how I would make a repo pattern and discuss why or why you wouldn't use another way.” (P6)

5 Findings on Research Question 2

To address the second research question, the analysis focused on the three themes present in the question: the development of LLMs and the effect of LLMs on workforce needs and skills in the future. The data revealed subthemes for some of the themes, which are discussed in the coming chapters.

5.1 Smarter, better LLMs

While some participants hesitated to predict the future of LLMs, all agreed that the use of LLMs would increase when asked about it. Most also stated that the advancement of LLM technology would continue rapidly. Participant 8, speaking with over 20 years of IT experience, expressed how many aspects of the LLMs will be improved:

"As the language models get bigger and better, and their intelligence/reasoning ability improves. In addition, the so-called context window, i.e., can enter more text and words into the language model." (P8)

Participant 9's equally experienced view on the advancements of LLMs matched that of P8's, stating that:

"The capacity that is utilized will expand. Reasoning ability, quality, and speed will improve." (P9)

Both participants also worked in executive-level positions at their respective firms. Participant 11, who possesses a quarter of the experience compared to Participants 8 and 9, on the other hand, had difficulties predicting LLM growth due to the multiple factors influencing the development of LLMs. They still predicted their ability to learn new things instead of applying trained knowledge:

"[When asked which direction the development of LLMs is headed to] *It probably goes towards language models knowing how to learn something new. I can't really say in which direction, since there are a lot of ethical and regulatory problems associated with it."* (P11)

The hesitancy due to the dynamic and fast-paced nature of LLM development was also reflected in the responses of Participant 5, who has 20 years of experience in the IT field, and Participant 12, who has 3.

"[When asked which direction the development of LLMs is headed to] *It's difficult to see very far. Every month there is something new." (P5)*

"[When asked which direction the development of LLMs is headed to] *I can't really take a stance, but yes, they will definitely get better all the time. Beneath the surface, there's a lot of potential available."* (P12)

Participant 6, who is working as a software architect with over 9 years of experience, did not voice opinions of how LLMs were going to improve but was sure that there was still a long way to go in terms of development:

"Some think that 99% of the development of language models has been achieved, in the minds of others it is 1%. I don't have enough understanding myself, but I think we are around one percent. We will catch up the rest of it and leave it at that." (P6)

In addition to improved Intelligence quotient (IQ) and processing power, some participants predicted LLMs with new features, such as multimodality. This was mentioned in the responses of Participants 10 and 8, who, despite their differences both in occupation and level of experience, saw multimodality as a future feature of LLM tools:

"Maybe at some point we'll see that technology will reach a point where it's harder to develop anymore. multimodality will be added there." (P10)

"Then the language models could become more multimodal or "generalist" models that could interpret images and videos and audio." (P8)

5.2 Effect of LLMs on Workforce Needs

5.2.1 Subtheme 1: Changing nature of work

The first subtheme identified the transformation of work and job descriptions due to future advancements in LLMs. This would also affect the job requirements when hiring

IT workers. Participant 6 mentioned the possibility of lowering job requirements due to employees being able to quickly solve problems and learn new skills, such as emerging programming languages, using LLMs. They highlighted the ability to bypass traditional learning methods through direct interaction with LLMs, simplifying tasks like code creation and documentation. Participant 6, a software architect with nine years of experience in the IT field, was sure that future LLMs would help automate tasks in his workflow:

"It will be that "tedious" i.e. boring, repetitive, and simple tasks will be left out. You make a documentation of your code, press enter and then everything happens exactly as you said. Those tasks are automated with a language model." (P6)

Participants 2 and 10 also predicted a shift in the nature of the software engineering role. Despite their differences in work roles and experience, both predicted programming automation. Instead of creating new code, engineers might focus on revising and managing automatically generated material.

"In the software industry, the methodological practices will most likely be replaced by automated practices. The role of software engineer will change from creating stuff to maybe just revising stuff." (P2)

"Coders will be more code readers because artificial intelligence could make the simple block codes." (P10)

Participant 8, as a director of product development, brought his view of an executive-level role and observed that this shift might reduce certain outsourced tasks while changing the nature of managerial positions:

"In management-type positions, the tasks have decreased radically, but in some places, they have also increased. The nature of jobs will change. For example, bulk tasks that have been subcontracted from India, their amount could be reduced considerably." (P8)

Participants 4 and 7, colleagues from company A, further discussed this transformation, predicting that repetitive tasks will decrease and developers may shift towards a supervisory role overseeing automated processes. Their views were echoed by Participant 12 despite their comparable lack of experience in the IT world.

"I believe my job -or any software developer's- will change in direction of just giving an instruction and there will be a mesh of agents, that will do the task, better than any human, and at some point, you will have to intervene and redirect them." (P4)

"At some point language models can be used to generate code. One language model can code and another one can test." (P7)

"I believe that the nature of the work will be different, manual things, things that have to be repeated, will radically decrease." (P12)

Overall, these insights indicate a significant transformation in the roles and responsibilities of IT professionals due to the advancements in LLMs.

5.2.2 Subtheme 2: Change in the amount of work

The interviews with IT professionals revealed various perspectives on how the amount of work may evolve due to the growing influence of LLMs. While some participants emphasized that no immediate changes have occurred, others predicted significant shifts in the quantity of work and the nature of job roles, particularly as LLMs continue to advance.

Several participants suggested that LLMs might reduce the overall workload by automating tasks, potentially decreasing the number of employees required for certain jobs. Participant 7, working as a software developer, had lots to say about the future of employment in the IT field. They emphasized that the same tasks might require fewer people in the future, stating, "*The same job requires less people*" when asked about the future workforce needs. They expanded this sentiment with the topic of AI-driven agents:

"*Agents made out of different language models can be used to create a team that can replace workers*." (P7)

A colleague from the same company, Participant 2, despite having 14 years more IT experience, had almost identical expectations with Participant 7:

"*Lots of software engineers like me will be losing or changing their jobs to something different. LLMs are going to be optimized and automated to replace people trained to do certain things."* (P2)

This view aligns with Participant 6's and 9's observations that less workforce may be necessary in the future.

"[When asked how LLMs will affect workforce needs] *Maybe on the lower-level jobs or simple jobs, less workforce is needed*" (P6)

"*We are going to go down in terms of labor needs. I believe that there is enough work, but the nature of the work will change*." (P9)

Again, both participants shared an over a decade-long gap in experience in the IT field but still shared the same vision of replaced jobs.

On the other hand, some participants underscored the potential for new opportunities emerging alongside these reductions. Participant 10 did not believe in a reduction in employment. When asked how workforce needs may change in the future, the research assistant of seven months stated: "*I don't think that employees will be replaced"* and later discussed the increase in the number of IT companies due to the rise of LLMs:

"*I think that there will be more companies in the IT field, for example individual software solutions for people's problems will be developed. It's hard to say if this means more jobs in the field"* (P10)

The sentiment of this prediction was shared by Participant 12, whose three years of experience as a software developer shaped his predictions of the future:

"*There will certainly also be new jobs. Simple work is decreasing in the IT sector, but there will also be more jobs."* (P12)

Participants 7 and 4, on the other hand, pointed out that there will likely be a need for individuals with specialized skills. Both participants had vastly different backgrounds in terms of experience and job positions but were able to identify a fundamental change in the IT industry:

"Probably a larger part of the employees in the IT field will work for companies whose business is to offer optimized language models." (P7)

"There will still be coders in the future, but they will just be different. I believe there won't be shortage of work, as there will be more time for developers and coders to do other things with help of LLMs." (P4)

Some interviewees also expressed caution about immediate changes. Participant 5, despite having over 25 years of experience in the IT field, remained vague in his response:

"*Right now, nothing changes. But the situation may be different in the near future*." (P5)

Participant 6, having worked nine years in the IT industry, offered an even more cautious view, noting that future developments in LLM capabilities will ultimately determine the extent of change:

"[Whether there is change in workforce needs in the IT industry] it *depends on the development of LLMs*." (P6)

Finally, a transitional phase was brought up by Participant 7, who pondered that initially, there will still be a need for people to teach others how to effectively use LLMs:

"*In this initial phase, people will still be needed to teach how to use language models*." (P7)

5.3 Effect of LLMs on Workforce Skills

Besides knowing the technical requirements, the software and IT workers of the future must master an array of other skills, according to the interview results. This is also stated by Participant 2, a software engineer with 12 years of experience in the IT field:

"[When asked which skills would be needed from IT workers in the future] *Having knowledge on many different things, but also be specialized in something."* (P2)

This sentiment was echoed by Participant 12, a software developer, who in turn emphasized the significance of familiarity with technology, legality, and foundational concepts:

"*Understanding about technology, legality, concepts related to language models such as how they work, etc., basic technological and software development understanding." (P12)*

In addition to technical knowledge, participants identified the need for interdisciplinary skills. Participant 11, a back-end developer with five years of IT experience, highlighted the relevance of psychological and philosophical proficiencies, citing the manipulation of language models and the philosophical foundation of logic as useful knowledge:

"Probably psychology to some extent, when language patterns can be, so to say, "manipulated" to give better and more accurate answers. Probably also philosophical skills as logic is based on philosophy." (P11)

Participant 3, the only cyber security expert among the participants, mentioned the importance of security solutions and social skills in the future:

"[When asked about what skills will be needed from IT workers] *Information security solutions and know-how will be highlighted. The importance of social skills has increased."* (P3)

Communication with LLM's emerged as a critical skill set, with several participants emphasizing its role in the future. Participant 4, a dev. Ops team lead with over 25 years of experience emphasized the importance of effective communication in conveying instructions to LLMs:

"Communication skills will be very important; how do you express things to LLM. And the ability to review what the outsourced LLMs create." (P4)

A similar thing was said by Participant 2, a colleague of Participant 4 with half the experience in the field:

"Companies will want employees who know how to work with LLMs. […] Like, knowing general things about LLMs and feeling confident about using them." (P2)

The need for understanding how to communicate with LLMs currently and in the future was discussed by Participant 6, who highlighted the desire of employers to obtain individuals with specialized skillset:

“It will be required that there is a lot of experience with language models. If we think about a developer who has built, for example, this [LLM application] or other applications with language models, that is already a plus. It's already visible now that it's a trending word, that companies will hire you if you have some experience, even if it's not used there.” (P6)

6 Discussion

6.1 How do IT professionals in Finland currently use LLMs to assist in their work?

The analysis of the interviews uncovered four higher-order themes on how LLMs are used in the software industry to assist in everyday work. These four themes were programming-related activities, non-programming-related technical activities, self-directed learning and adaptation to ambiguity, and being a team player. In the following sections, each aggregate theme will be explored in terms of subthemes and connections to the literature. After, the themes will be summarized and discussed as a whole.

6.1.1 Programming-related activities

The first aggregate dimension identified in the analysis is programming-related activities, which include writing and reviewing code, testing code, and identifying and fixing bugs. Nearly all participants commented on this theme, accentuating the central role that programming plays in their daily work.

A key finding is that most participants reported using LLMs for programming-related assistance, indicating that these tools are not only integrated into their workflows but also enhance the productivity of programming tasks. The ability to quickly generate code snippets, as described by Participant 7, and tools like GitHub Copilot, as mentioned by Participant 6, demonstrate how IT professionals have adopted LLMs to automate and assist in once time-consuming tasks. Some participants also highlighted their use of LLMs to generate test cases, reflecting a broader desire to streamline the more repetitive aspects of their work. Lastly, participants of all experience levels also highlighted the role of LLMs in debugging and error resolution. However, while LLMs were considered valid for finding the bugs and diagnosing and solving the error, some voiced caution about the limitations in the capabilities of LLMs. Participant 11's statement reiterated this ambivalence, pointing out that despite the excellence of LLMs as problem-solving tools, reliance on them without proper verification can introduce risks in the debugging process.

This first aggregate dimension revealed that IT professionals utilize LLMs for programming and other programming-related tasks, such as testing and finding, diagnosing and fixing errors in the code. These findings strongly support previous research that has identified programming as a core technical skill among software professionals (Begel & Simon, 2008; Exter, 2014; Lethbridge, 2000; Scaffidi, 2018), as well as that programming-related tasks occupy a significant portion of their working hours (Meyer et al., 2017). The findings reinforce these earlier conclusions while echoing the work of Khemka and Houck (2024), who documented the desire among software professionals to leverage LLM tools for automating more straightforward, more labor-intensive tasks. Overall, the first aggregate dimension demonstrates that IT professionals have already integrated LLM-based tools into their workflows to assist with programming, solve errors, and automate repetitive tasks. While these tools are seen as helpful assistants, participants generally refrained from relying on them for more complex programming challenges. Strengthening the findings that LLMs are already capable of error-solving and programming (Ebert & Louridas, 2023; Liu et al., 2024; X. Peng et al., 2019). The ability of LLMs to generate code and suggest solutions has proven valuable in increasing the speed and efficiency of development tasks.

6.1.2 Non-programming-related technical activities

The second aggregate dimension includes other non-programming-related technical activities, such as data analysis and documentation. The participants who reported using LLMs for data analysis either mentioned it in passing or described using them for a particular purpose, such as generating visualizations, transforming or mapping their data. For enthusiastic Participant 9, the analytics function in LLMs was a “WOW factor,” praising the diverse capabilities of LLMs to both analyse and visualize data. However, while participants praised LLMs’ data-related capabilities, the absence of detailed technical descriptions indicates that LLMs are likely being used primarily for straightforward or supplementary tasks rather than complex data analysis workflows, highlighting the current boundary of LLM adoption in advanced technical roles.

Participants also identified LLMs as valuable tools for processing large volumes of technical documentation. Early-career professionals (Participants 1 and 3) and more experienced specialists (Participant 6) mentioned using LLMs for tasks such as summarizing, extracting key points, and reviewing lengthy materials. This demonstrates LLMs' ability to reduce cognitive load and enhance productivity by simplifying information retrieval and comprehension. Participant 6's use of LLMs to process personal work-related documents suggests a deeper level of adoption, potentially reflecting trust in the technology's ability to handle proprietary information. However, such usage also raises concerns about data privacy and confidentiality, which is highlighted in the LLM policies of many companies.

The third aggregate dimension shows how LLMs assist with other technical tasks besides programming. With LLMs assisting in modeling and exploring the data, more time can be allocated to carefully interpreting the results. The use of LLMs to process documentation, however, shows that the arduous task of going through large texts leaves software professionals desiring a more efficient way to handle documentation. This shift from technical to cognitive tasks illustrates the broader trend of the industry toward automation and efficiency.

6.1.3 Self-directed learning and adaptation to ambiguity

The third aggregate dimension, self-directed learning and adaptation to ambiguity, encompasses activities related to acquiring new information through LLM tools, which were found to function as both educators and search engines.

Many participants described LLMs as helpful tools for learning purposes. They were used as a replacement for a teacher when trying to understand new material and praised for putting the information in a straightforward and easy-to-understand format, as highlighted by participants 1, 4, and 6. Additionally, Participant 6's description of learning new programming languages through conversational interaction with LLMs showcases the models' ability to facilitate skill development in specialized technical domains. Their experience illustrates a shift from passive learning through reading to interactive learning through dialogue.

In addition to learning, participants described using LLMs instead of traditional search engines like Google. This preference for LLMs over traditional search engines, as described by Participants 4 and 5, emphasizes the tool's effectiveness in providing practical, task-specific information. However, while participants highlighted convenience, none mentioned concerns about misinformation or incorrect responses. This absence could reflect either limited negative experiences or a lack of critical evaluation due to the perceived reliability of LLM outputs. This raises concerns about potential overreliance on AI-generated content, especially in domains where accuracy is critical.

Overall, the interview analysis reveals that IT professionals use LLMs for various learning purposes, whether to quickly search for information about new technologies or seek guidance on mastering an entirely new programming language. The widespread adoption of LLM technology already reflects the eagerness and motivation of software professionals to learn and adapt to new methodologies. Moreover, using LLMs for personal learning demonstrates creativity and a willingness to explore these tools beyond what might be required of them in a professional capacity. Integrating LLMs into self-directed learning processes highlights a broader shift in how IT professionals approach knowledge acquisition. These tools are transforming learning by enabling professionals to acquire new skills and technologies on the fly without formal, structured instruction. This shift suggests a move away from traditional learning methods toward more flexible, on-demand approaches to acquiring knowledge.

6.1.4 Being a good team player

The last aggregate dimension, being a good team player, consists of activities where LLMs assist in communication tasks and support independent working. Several participants described using LLMs to improve communication with stakeholders, such as clients, teams, or subordinates. For example, Participant 1 mentioned utilizing LLMs to help generate PowerPoint presentations in the complicated MS Office environment, while others described use cases in which LLMs were used to improve emails. This capability allows professionals to focus more on the strategic aspects of their work rather than getting bogged down by email formatting and other routine tasks.

A key advantage identified from the interviews was LLMs' role in facilitating multilingual communication. Participants noted how these tools help manage interactions with clients and colleagues who speak different languages, ensuring more accurate and culturally appropriate communication. LLMs also enhance professional correspondence, with participants highlighting how the tools improve the clarity and style of emails, making them more polished and efficient.

Another significant use case that emerged from the analysis was self-directed working, which included use cases such as brainstorming and finding starting points for projects. Participants described LLMs as valuable “sparring partners” for developing and refining ideas, especially during the early stages of projects. These tools provide different perspectives and suggestions, helping professionals explore different approaches and make informed decisions. This functionality extends beyond general brainstorming to more specific technical contexts, where LLMs can be used to discuss and evaluate different solutions or design patterns. Such interactions help foster deeper individual reflection and critical thinking, enhancing the overall quality of the decision-making process.

The integration of LLMs into communication and independent working practices illustrates their growing role in modern IT practices. Communication also emerged as a critical soft skill in literature, important in coordinating efforts and facilitating collaboration (Begel & Simon, 2008; Capretz & Ahmed, 2010). These tools enable professionals to make better decisions regarding their work tasks by automating and aiding communication and supporting decision-making processes. This transformation not only boosts efficiency but also enhances the quality of work, reflecting the evolving demands of the industry and the continuous need for adaptability and creativity in the field.

6.1.5 Summary

The analysis reveals that IT professionals in Finland use LLMs across four primary dimensions: programming-related tasks, non-programming technical tasks, self-directed learning, and being a good team player. LLMs assist in automating repetitive coding tasks, debugging, and generating test cases, enhancing productivity. They also facilitate data analysis, document processing, and summarization, allowing professionals to manage large information loads efficiently. Moreover, LLMs support self-directed learning by functioning as interactive tutors and search engines, enabling users to acquire new skills on demand. In teamwork, they facilitate independent working by aiding in brainstorming and decision-making and enhance collaboration through automated writing support and multilingual communication tools.

The findings reinforce existing research on digitalization theory, which posits that digital tools reshape work processes and task automation (Gorenek & Kohont, 2019). Integrating LLMs into software development illustrates this transformation, as professionals increasingly rely on LLMs to perform complex tasks with minimal human intervention. This aligns with studies highlighting how digital technologies drive organizational change through increased productivity and efficiency (Maruping & Matook, 2020). Cultural lag theory (Ogburn, 1957) also emerges as a relevant framework, which explains how the developments in material culture, i.e., LLM technology, advance faster than the non-material culture, i.e., social norms, organizational and ethical policies. Contrary to this theory, software development companies have shown relatively quick adaptation of LLM technologies and related policies. Yet, while participants and companies in this study have embraced LLMs, concerns about data privacy and trust in AI outputs suggest that industry standards and cultural norms are still evolving.

Overall, the findings demonstrate that LLMs are reshaping the IT profession by blending technical, cognitive, and interpersonal capabilities. This reflects broader technological and societal shifts, highlighting the need for adaptive organizational policies, lifelong learning approaches, and continued exploration of human-AI collaboration.

6.2 What are the possible future implications of developments in LLM technology on workforce needs and skills in the Finnish IT field?

The thematic analysis of the interview results revealed three main themes: Smarter, better LLMs, the effects of LLMs on work, in which the changing nature and amount of work were discussed, and the effects of LLMs on workforce skills, in which participants discussed the future skillsets required from IT workers. In the next section, these findings and their implications will be discussed in more detail.

6.2.1 Smarter, better LLMs

The interviews revealed a shared belief among participants that LLM development will continue rapidly despite varying degrees of certainty about its future direction. All participants agreed that using LLMs would increase, although their views differed on how and when specific advancements would occur. While some participants focused on technological improvements such as enhanced reasoning, context-handling, and multimodality, others acknowledged challenges related to ethical and regulatory complexities that could influence development speed.

Participants with significant industry experience, such as Participants 8 and 9, emphasized that LLMs would become more capable, particularly in reasoning, processing power, and context window size. Their executive-level positions likely contributed to this perspective, giving them a broader understanding of industry trends and technological trajectories. Participant 8, for example, predicted that "*as the language models get bigger and better, and their intelligence/reasoning ability improves*," suggesting confidence in continuous progress. Multimodality was also discussed as a potential area of development, mentioned by Participants 8 and 10. Participant 8 envisioned future LLMs becoming "more multimodal or 'generalist' models that could interpret images, videos, and audio," suggesting that LLM capabilities might expand beyond text-based applications. This reflects industry goals of integrating various data types into unified models, supporting broader and more complex tasks.

Despite sharing this optimism, not all participants felt confident predicting specific advancements. Participants 5 and 12, for example, expressed hesitation due to the rapid pace of change. Participant 5 stated, "*It's difficult to see very far. Every month there is something new.*" This uncertainty reflects the unpredictable nature of technological innovation, where breakthroughs often emerge unexpectedly, making long-term forecasting challenging. Some participants also recognized potential obstacles to LLM development. Participant 11, for instance, acknowledged that ethical and regulatory concerns could influence LLM development. They noted, "*I can't really say in which* *direction, since there are a lot of ethical and regulatory problems associated with it.*" This perspective aligns with Ogburn's (1957) cultural lag theory, which argues that non-material culture, including legal and societal norms, often lags behind technological advancements.

Overall, participants viewed LLMs as advancing rapidly, although they expressed varying confidence levels regarding future developments. Those with more experience focused on technical improvements, while others highlighted uncertainties stemming from the unpredictable nature of technological progress and potential regulatory barriers. This combination of optimism and caution reflects LLM development's complexities, shaped by technological potential and societal constraints. As LLM technology evolves, balancing technological progress with ethical considerations will likely remain a critical challenge.

6.2.2 Effect of LLMs on Workforce Needs

The findings from the study suggest an impending paradigm shift in the roles and job requirements of IT professionals due to LLMs. Participant 6's observations about the potential for lowering job requirements due to easy access to information and increased independent learning underscore a broader trend toward reducing formal education and prior experience in favor of LLM-powered skill acquisition. As discussed in the previous chapter, the increased demand for independent learning is already present in the current IT climate, with professionals turning to LLMs for new technologies, features and programming languages. This ability to quickly acquire proficiency in new programming languages and tools via LLMs could shift the hiring focus from traditional academic qualifications to adaptability, logical capabilities and technological competence.

The insights from participants 2, 8, and 10 share the view that the future role of software developers may move from manual coding toward reviewing and managing AI-generated outputs. Participants 12 and 4 take this prediction a step forward, describing a future where software developers assume supervisory roles, managing networks of agents responsible for automated tasks. These predictions align closely with recent discussions on the impact of AI in software development, where, in some frameworks, automation is expected to take over lower-level tasks, allowing developers to focus on more complex oversight functions (Sauvola, 2024). Additionally, the concept of task reallocation echoes Cascio and Montealegre's (2016) notion of reshaped work processes. The reduction of bulk, outsourced tasks mentioned by P8 further hints at a potential shift in the global labor market, where AI may disrupt established workflows in outsourcing-heavy industries.

In addition to the change in the tasks and responsibilities at work, participants also discussed how the amount of work might change in the future. Several participants predicted a decrease in the overall demand for software workforce due to task automation. As LLMs continue to optimize processes, routine and lower-level tasks are expected to require fewer human workers. This highlights a growing expectation that certain roles, especially those involving repetitive or straightforward tasks, may become redundant. The concept of AI-driven teams replacing human employees stresses concerns about job displacement, with many seeing this as a likely trajectory if LLM capabilities continue to expand.

On the other hand, this reductionist view was countered by the belief that technological progress could also generate new opportunities. While some participants foresaw job losses, others anticipated the creation of new roles driven by emerging, LLM-based companies and new technologies. It was also noted by Participant 7 that in the short term, the demand for workforce will increase as companies try to catch up with the developing LLM tools and technologies, creating a transitional phase. The split in the participants' answers could also be interpreted as decreased demand for lower-level employees, such as standard programmers, and increased demand for employees with higher-level specializations in emerging technologies and AI solutions. Thus, while certain jobs may disappear, others requiring advanced expertise will likely emerge, creating a more dynamic but potentially polarized job market. Naturally, some participants also remained cautious when discussing the implications of LLM technology in their field. A few stressed the uncertainty surrounding the pace and scale of these changes, noting that the development trajectory of LLMs would ultimately be the one to dictate their impact on workforce needs.

The collective insights gathered in this theme suggest a need for organizations to rethink their workforce strategies. The results highlight a dual narrative surrounding the impact of LLMs on the IT workforce. While automation-driven job reductions seem inevitable for lower-level tasks, the emergence of new roles requiring specialized skills offers a counterbalancing opportunity. The evolving technological landscape points to a future where adaptability, continuous learning, and higher-level expertise will be critical for IT professionals to remain relevant and competitive.

6.2.3 Effect of LLMs on workforce skills

The interview findings suggest that future IT professionals will need a diverse set of skills that extend beyond traditional technical expertise. Participant 2 emphasized the necessity of broad understanding while specializing in a specific area, illustrating the expected balance between versatility and expertise in the future IT workforce. This dual expectation reflects the evolving nature of IT work, where adaptability and continuous learning are essential to navigate a wide range of tools and methodologies in a rapidly changing environment.

In addition to technical prowess, interdisciplinary skills emerged as critical competencies. Participant 11 emphasized the potential relevance of psychology and philosophy, pointing to the underlying logical principles that shape LLMs and the ability to manipulate language patterns effectively. As language models increasingly shape how information is processed and communicated, understanding human behavior, logic, and reasoning may become essential for optimizing LLM outputs and ensuring ethical implementations. Participant 3, an information security specialist, also discussed the growing importance of information security. These discussions suggest a future where IT professionals might need training that blends technical education with social sciences and humanities.

Another recurring theme across the interviews was the ability to communicate effectively with LLMs. Several participants underscored the need for clear and precise communication when working with these models. Participant 4 stressed the importance of expressing tasks clearly to LLMs, while Participant 2 highlighted the necessity of developing confidence when using such tools. Experience and knowledge about LLMs can prove to be a competitive edge, as observed by Participant 6, even when such experience is not required. Throughout the interviews, the ability to craft detailed prompts, interpret results, and refine outputs is increasingly recognized as a valuable skill in IT roles. This trend of keeping up with developing LLM technology recognized by the participants mirrors the findings of Pappas et al. (2018), whose Digital Transformation and Sustainability model emphasizes the importance of keeping up with technological megatrends to harness the potential of digital transformation.

Overall, the results suggest a redefinition of the IT skill set driven by the integration of LLMs. Future professionals must blend technical expertise, interdisciplinary knowledge, strong communication skills, and adaptability to change to stay competitive in an ever-evolving field. This emphasis on interdisciplinary skills reinforces Capretz and Ahmed's (2010) emphasis on the importance of soft skills in the software industry. Similarly, the predicted need for effective communication with LLMs mirrors the idea that digital transformation extends beyond technical upgrades to include broader management and cultural changes (Akter et al., 2022; Brock & von Wangenheim, 2019). The theory of cultural lag by Ogburn (1957) provides further context, explaining participants' concerns about slow institutional adaptation leading to increased emphasis on personal learning. While companies will likely seek LLM-trained workers quickly, educational and legal systems may struggle to keep pace, reflecting the cultural lag between technological progress and non-material societal structures.

6.3 Theoretical and Practical Implications

The results of this study align with and expand upon existing theoretical frameworks in the fields of technological evolution and labor market transformation. Theories of automation and job displacement (Berger & Frey, 2016) highlight how repetitive and routine tasks are increasingly automated, leading to shifts in job roles and the emergence of new occupational categories. This study reinforces these concepts by showing how LLMs could further reduce the need for manual coding and routine IT tasks, pushing software developers toward oversight, review, and strategic roles.
The cultural lag theory (Ogburn, 1957) is also relevant to this study's findings. As LLM technology rapidly advances, societal and institutional frameworks—such as ethical frameworks, legal regulations, educational systems, and corporate structures—struggle to adapt. The participants' concerns about slow academic and organizational adoption and regulatory ambiguity echo this theoretical perspective, emphasizing the gap between technological capabilities and societal readiness. Additionally, the findings support theories of continuous digital transformation in organizations (Vial, 2019;

Pappas et al., 2018). Participants' predictions about LLM-driven changes in IT work processes underscore the dynamic nature of digital transformation, where evolving AI technologies continuously reshape job roles, organizational structures, and required skill sets.

Finally, this study supports and showcases the frameworks for developing and integrating generative AI into software development processes, as described by Sauvola et al. (2024). The current use cases of LLMs in software development, as explored in the first part of this study, support the beginnings of scenario two, where AI is used to automate selected parts of manual and repetitive tasks such as code generation, documentation, and testing. The analysis of the interviews revealed predictions that software developers are primarily responsible for overseeing the workings of LLMS and performing more complex tasks, indicating transitions to scenarios three or four. These transitions correspond to case 3, as described by Sauvola et al. (2024), where scenario two is reached in the short-to-mid term, followed by slow transitions to scenario 3, and ultimately, scenario four as AI tools and their integration into the field matures.

From a practical standpoint, the study provides actionable insights for several key stakeholders, including professionals, educational institutions, and policymakers. The findings suggest that IT professionals should prioritize developing adaptable skill sets, including familiarity with LLM technologies, critical thinking, and advanced problem-solving. Skills like reviewing AI-generated material, managing AI-driven projects, and effective communication with AI systems could become essential in future IT roles. On the other hand, companies should invest in continuous learning programs that help employees acquire and update LLM-related skills. The findings of this study show that employees are motivated and ready to adapt LLM technologies to assist them in their work, giving companies the go-ahead to start the integration process of the available tools. Organizations might also want to rethink their hiring criteria, placing greater emphasis on adaptability, lifelong learning, and digital competency over traditional qualifications. While AI- and LLM-related curriculum already exists in some Finnish universities, the educational sector must adapt its curricula to reflect the growing importance of LLM technologies. Universities and training programs should integrate courses on AI-powered development tools, legal and ethical considerations of AI, and interdisciplinary fields like psychology and linguistics to prepare graduates for future job markets. Finally, given the rapid development of LLMs, policymakers should create clear regulations governing AI use, data privacy, and ethical standards. Legislative frameworks should balance innovation with societal protection, ensuring that advancements benefit both industry and society.

6.4 Limitations and Future Work

While this study provides valuable insights into the evolving role of LLMs in IT work, several limitations should be acknowledged. First, the study relied on qualitative interviews with a relatively small sample size, limiting the generalizability of the findings. Additionally, while this study addressed using LLMs in the software industry, not all participants worked there. Participants were also predominantly from IT-related professions in a specific geographic and cultural context, which may not reflect national or global perspectives on LLM adoption.

Second, the fast-paced nature of AI development means that the findings represent only a snapshot in time. Given the rapid advancements in LLM capabilities, some technological predictions made by participants may soon become outdated. Similarly, the regulatory and societal responses discussed could change as legal frameworks and institutional practices adapt. Finally, while the study explored a wide range of implications for software work, it did not examine other industries where LLMs may also have transformative impacts. Future research could explore a more diverse range of sectors to provide a comprehensive understanding of LLM-driven changes across the labor market.

The current climate in the IT- and software industry could provide an ideal time for longitudinal studies tracking how LLMs reshape work roles over time, showcasing the effect of digital transformation on the industry. Future research could also include a culturally and economically more diverse pool of participants to explore whether LLMs affect global labor markets differently from the Finnish sample. The growing integration of LLM-based tools also opens possibilities for research on human-AI collaboration and the effects of AI management on employee productivity and cognitive load. Finally, the findings of this study showcase the importance of examining the broader social and economic consequences of LLM adoption, including job displacement, wage polarization, and the digital divide. Such research would offer a holistic understanding of LLM-driven societal change.

7 Conclusion

This study aimed to explore how software professionals in Finland currently use AI tools to assist them in their work and to find out the possible implications of developments in LLM technology on future workforce needs and skills. To answer these questions, 12 IT professionals were interviewed. The analysis of the results revealed that the LLM tools are currently employed to assist and automate programming-related efforts, assist in non-related work activities such as data analysis and documentation, help in self-directed learning and overall be a good team player. These findings suggest a trend in the industry of employing tools for automation, increased productivity and self-improvement. This study also found that IT professionals predict the rapid advancement of LLM technology, leading to the automation of work tasks and displacement of jobs, fundamentally shifting the industry from manual labour to more complex cognitive tasks. These changes were predicted to change the skillset of future software workers, where adaptability and a diverse skillset are sought over the traditional technical skillset. Together, these results showcase an ongoing shift from manual tasks to automation, increasing the value of self-directed learning and more complex skills. Based on these results, more research on human-AI teamwork, as well as the implication of LLMs on other sectors, are recommended together with actions from businesses to support lifelong learning and adaptability and from institutes to design more training programs and courses focused on interacting with LLMS

8 Appendix A

Table 3. The interview questions

1. How old are you
2. What is your role at your workplace?
3. How long have you been working in IT?
4. What is your workplace's policy regarding LLMs?
5. Does your workplace or your team have its own practices regarding LLMs?
6. How often do you use LLMs at work?
7. Can you describe a specific situation or project where you used a LLM to help you?
8. In what kind of situations are LLMs most useful to you?
9. Do you know any other ways to utilize LLMs in your work environment?
10. Can you see any clear trends on How LLMs are used in IT industry at the moment?
11. Can you see any clear trends on How LLMs are used in ICT industry in the future?
12. In which direction do you think the development of LLMs is headed for?
13. Do you think the use of LLMs increases in the future?
14. How can language models influence the methods used in the IT industry?
15. How will LLMs change the workforce demands in the IT industry?
16. What kind of skills will be needed from the IT workers in the future?
17. How have LLM changed the IT industry in 5 to 10 years?

9 Appendix B

Table 4. First-order codes with related quotes from participants.

To write code for me	"[While describing most useful use cases for LLMs] …and also to make and run Python-code." (P9) "[When describing situations in which LLMs are useful] ... and writing code for me" (P8)
Generate basic code	"[While describing a project in which LLMs were used] I last used it for basic coding, just to make the work faster […] Small snippets of code can be obtained using the language model faster than otherwise." (P8)
Getting help while coding	"[When asked how often they use LLMs] It's with me all the time to help while coding" (P10)
Getting suggestions while coding	"[When describing most useful use cases for LLMs] During the coding phase, Copilot helps with coding suggestions." (P6)
Generating tests	"[When listing use cases for LLMs] even up to generating test cases for your code." (P4)
For different tests	"[While describing most useful use cases for LLMs] It's also really good for doing different tests." (P11)
Helping with configuration	"[When describing most useful use cases for LLMs] When you need to implement a feature in some configuration, so help is sought from language models to make the software compatible." (P5)
To find out why a framework does not work	"[When describing ways to use LLMs for schoolwork] If the framework I made doesn't work, I'll ask ChatGPT why not. Or I ask if the method I am considering is the right one for the situation of my code task." (P1)
Debugging and Error Solving	"I feel that now is the most optimal time to use language models, as a middle level developer, for example for problem solving. […] When I'm looking for frames for certain types of problems that are not already familiar. Admittedly, this is partly a bad way because verifying a problem fix is difficult, because blindly trusting a fix has

	its own risks, also in terms of one's own development." (P11)
Debugging and Finding syntax errors	"[When describing most useful use cases for LLMs] It's really good for debugging, and for finding syntax errors." (P10)
Dealing with customer support error messages	"[When listing uses for LLMs] It is also used in customer support matters to deal with error messages." (P5)
Asking how to fix error codes	"When a program error occurs, asking LLMs how to fix specific errors that are not user friendly, but also diagnose a program." (P2)
Retrieving important points from long text	"[When listing uses for LLMs] For retrieving important points from a long text." (P1)
Going through large materials	"[When listing uses for LLMs] For reviewing of large materials." (P3)
Understanding and Summarizing Documents	"[When listing cases in which LLMs are useful] To understand and summarize documents. [...] Putting your own documents in the language model." (P6)
Getting summaries and references	"[When describing other known ways to use LLMs] You can generate and insert text references. Summarize text or search for sources, etc. " (P7)
Assisting in Transcription	"Artificial intelligence helped with transcription during the project." (P9)
Commenting own work	"[When describing most useful use cases for LLMs] It's really good for debugging, and for finding syntax errors. And for commenting own work" (P10)
Data Analysis	"[While showing a slide of how employees use LLMs] For data analytics." (P8)
Making Graphs	"The Analyst function has been a "wow factor," for example you can ask GPT to make a graph and analyze the results." (P9)
Transforming data into another format	"[When describing most useful use cases for LLMs] If the data is in a certain format and you want it in another format, the language model does it for you." (P10)
Mapping Data	"[When describing most useful use cases for LLMs] In trivial matters that are not very recent, but older, that you can define well. Like data mapping." (P12)
Learning about unfamiliar technologies	"I most often go on ChatGPT or Bard and ask specific questions about certain technologies that I am not familiar with." (P2)

As a replacement for a teacher	"I also use LLMs for learning. I can ask it questions and clarifications instead of a teacher. To understand better." (P1) "I have also used it to learn a new programming language." (P6)
Learning a new programming language	"I've noticed that if a new instruction or a language comes along, you don't need to read anything, you can start from a blank slate and talk to that language model, and you can get things done surprisingly quickly. What should have been learned earlier in the course or through articles. Now all you have to do is read what the tool is and where you download it and start talking with it." (P6)
Elaborating on complex things	"[When describing most useful use cases for LLMs] If you have very complex things to elaborate, it can guide you the way to find things." (P4)
As a replacement for Google	"The most typical usage situation is to replace Google searches." (P9) "[When describing most useful use cases for LLMs] Much better than Google, as it puts it to the point." (P4)
To find out something	"[When describing most useful use cases for LLMs] If something needs to be found out, ChatGPT is surprisingly quick to produce answers." (P5)
Ask how to change data to another format	"[When describing how LLMs are utilized at work] For asking, for example, how to change the data type from one to another." (P7)
Asking for information	"[When describing most useful use cases for LLMs] You can easily get information by asking." (P10)
Asking about things you cannot remember	"[When describing other known ways to use LLMs] Instead of searching for information on different websites, you can just ask the language model." (P5)
Correcting language in emails	"[When describing most useful use cases for LLMs]I might correct the language in emails in different languages." (P8)
Automated emails	"Automatically generated emails." (P12)
Improve the style of written output	"I might correct the language in emails in different languages. I speed up and improve the style of the written output (e.g. I want to say these things in this style)." (P8)
Help communicating in different languages	"[When describing most useful use cases for LLMs] If there are a lot

	of customers and you have to communicate in different languages, check the dialects. Just in daily communication." (P3)
Generating PowerPoints	"[While listing most useful use cases for LLMs] PowerPoint generators for Office." (P1)
Generating Content based on words	"[When describing most useful use cases for LLMs] Ability to create content based on words." (P5)
To find a starting point	"[When asked in which situations LLMs are the most useful] For when you're looking for a starting point." (P3)
Getting ideas on how to start a project	"[When describing last time using LLMs] Just throwing in some ideas like how to start a project is a common way to use it." (P4)
As a sparring partner for brainstorming	"[When describing most useful use cases for LLMs] I personally use it for generic sparring for ideas." (P11)
Getting different points of view in order to make a decision	"It's a good sparring partner for brainstorming or to get different points of view to help make a decision." (P8)
Asking how and why a certain implementation	"[When describing a case in which LLMs were used] In the conversational model, I ask, for example, how I would do a repo pattern and discuss why or why you wouldn't use another method." (P6)

10 Appendix C

Table 5. Participant quotes for each theme related to the second research question.

Better LLMs	"[When asked which direction the development of LLMs is headed for] It depends on how the law intervenes in the matter like training language models by stealing people's data." (P1) "[When asked which direction the development of LLMs is headed for] Artificial general intelligence could be putting together an intelligence from LLMs and vector databases, comparable to a human brain." (P2) "You will have a team of LLMs (virtual software team, that each "member" has a specific role), which will all be automatic. With someone to run it. […] I think in software development, I believe the user interface will change. Maybe a more verbal way of communicating with IT and machines." (P4) "[When asked which direction the development of LLMs in headed to] It's difficult to see very far. Every month there is something new." (P5) "I believe it's in the tools. I don't think that a certain part can be replaced by a language model, but language models bring more and make the work easier. It's an added touch, whether it's a chat or another tool." (P5) "Some think that 99% of the development of language models has been achieved, in the minds of others it is 1%. I don't have enough understanding myself, but I think we are around one percent. We will catch up the rest of it and leave it at that." (P6) "As the language models get bigger and better, and their intelligence or reasoning ability improves. In addition, the so-called context window, i.e. can enter more text and words into the language model. Then the language models could become more multimodal/"generalist" models that could interpret images and videos and audio." (P8) "The capacity that is utilized will expand. Reasoning ability, quality and speed will improve. […] I see a big change in software development when combining in multimodality." (P9) "Maybe at some point we'll see that technology will reach a point where it's harder to develop anymore. multimodality will be added there." (P10) "[When asked which direction the development of LLMs is headed for] Probably for

	language models to know how to learn something new. I can't really say in which direction, since there are a lot of ethical and regulatory problems associated with it (the development of language models). Maybe even in a scary direction, if we go ahead with the incentives in mind and don't think about any potential risks." (P11) "[When asked which direction the development of LLMs is headed for] I can't really take a stance, but yes, they will definitely get better all the time. Beneath the surface, there's a lot of potential available." (P12)
Change in the nature of work	"The role of software engineer will change from creating stuff, to maybe just reviewing stuff." (P2) "I believe my job (or any software developer) will change in direction of just giving an instruction and there will be a mesh of agents, that will do the task (better than any human) and at some point you will have to intervene and stop. You will have a lot of help from experts (LLMs)." (P4) "It will be that "tedious" i.e. boring, repetitive and simple tasks will be left out. It means that you make a documentation of my code, press enter and then everything happens exactly as you said. Those tasks are automated with a language model." (P6) "Language models will be able to produce code at some point. One language model could code and the other test. Could be used as a code scanner that catches bugs." (P7) "The nature of jobs will change. For example, bulk tasks that have been subcontracted from India, so the number of them can be reduced considerably." (P8) "We will probably go down in terms of workforce needs. I believe that there is enough work, but the nature of the work will change." (P9) "Coders will be more like code readers, because artificial intelligence can make the code. Artificial intelligence agents could be software entities that you could talk to with natural language." (P10) "I believe that the nature of work will be different, manual things (things that have to be repeated) will radically decrease. […] It has resulted in more productive activities, which enables us to make software faster, and people have moved from time-consuming tasks to tasks that require more thinking and cognitive ability." (P12)
Change in workforce needs	"[When asked how LLMs could change workforce needs] It depends a lot on the employers. Streamlining wears people out quickly and the skilled workers run out. Quality suffers and the industry's culture is lost." (P1) "Lots of software engineers like me, will be losing or changing their jobs to something different. LLMs are going to be optimized and automated to replace people trained to

	do certain things." (P2) "There will still be coders in the future, but they will just be different. I believe there won't be shortage of work, as there will be more time for developers and coders to do other things with help of LLMs. Just in general how we do things will be much easier." (P4) "[When asked how LLMs could change workforce needs] Nothing changes at the moment. The situation may be different in the near future." (P5) "Yes, if the same work can be done with fewer people. Thus, in the initial phase, people will still be needed to teach how to use language models. /// Probably a larger part of the employees in the IT field will work for companies whose business is to offer optimized language models. The industry expands or changes." (P7) "In management-type positions, the tasks have decreased radically, but in some places they have also increased. […] The technical ability of language models will change the fastest, but society and professions will change with a delay." (P8) "I don't think that employees will be replaced, however. […] I think that there will be more companies in the IT field, for example, individual software solutions for people's problems will be developed. Hard to say if this means more jobs." (P10) "[When asked how LLMs could change workforce needs] Not much. If the language models are able to automate your work, then you should think about whether you are doing meaningful work […] At the current point, [LLM] is only a tool that can be used to support each task, such as indexing documentation, giving feedback, etc. in the above. However, as such, it will not replace any work task. However, automation will increase." (P11) "There will certainly also be new jobs. Simple work is decreasing in the IT sector, but there are more jobs." (P12)
Wider skillset	"Companies will want employees who know how to work with LLMs. […] Like, knowing general things about LLMs and feeling confident about using them." (P2) "In the future, job seekers must have knowledge of how to use language models and how to apply them to jobs." (P3) "[When asked about what skills will be needed from IT workers] It will be required that there is a lot of experience with language models. If we think about a developer who has built, for example, this RAG or other applications with language models, that is already a plus. It's already visible now that it's a trending word, that companies will

hire you if you have some experience, even if it's not used there.” (P6)

“[When asked about what skills will be needed from IT workers] Basic knowledge of using AI and language models and communicating with them.” (P7)

“[When asked about what skills will be needed from IT workers] Having knowledge on 'many different things', but also be specialised in something.” (P2)

“[When asked about what skills will be needed from IT workers] Information security solutions and know-how will be highlighted. The importance of social skills has increased.” (P3)

“Communication skills will be very important, how do you express things (to LLMs). Ability to review what outsourced LLMs create.” (P4)

“[When asked about what skills will be needed from IT workers] Hard to say... Judgment (whether this thing is necessary or not), ability to communicate with AI.” (P5)

“Requirements can be lowered. […] The need for training decreases and the initial level of requirements should decrease in a certain way […] You can get further by being independent and using tools. You can learn everything from language model.” (P6)

“[When asked about what skills will be needed from IT workers] I can't really say. If you want to make models that code well, then "prompt engineering" would be a good skill. And communication skills with AI.” (P7)

“[When asked about what skills will be needed from IT workers] All the people in the world have access to the same smart engines, but the value of tasks next to these, such as knowledge management, is increased.” (P8)

“[When asked about what skills will be needed from IT workers] Communication, people-oriented things.” (P9)

“The importance of accurate personas will increase. Understanding how to read code will be necessary.” (P10)

“[When asked about what skills will be needed from IT workers] Probably psychology to some extent, when language patterns can be so called "manipulated" to give better and more accurate answers. Probably also philosophical skills, when logic is based on philosophy.” (P11)

“[When asked about what skills will be needed from IT workers] Understanding of technology, legality, concepts related to language models such as how they work, etc.,

basic technological and software development understanding." (P12)

11 References

Akter, S., Michael, K., Uddin, M., McCarthy, G., & Rahman, M. (2022). Transforming business using digital innovations: the application of AI, blockchain, cloud and data analytics. *Annals of Operations Research*, *308*. https://doi.org/10.1007/s10479-020-03620-w

Begel, A., & Simon, B. (2008). Novice software developers, all over again. *Proceedings of the Fourth International Workshop on Computing Education Research*, 3–14. https://doi.org/10.1145/1404520.1404522

Berger, T., & Frey, C. B. (2016). Structural Transformation in the OECD. *OECD Social, Employment and Migration Working Papers*, *193*.

Braun, V., & Clarke, V. (2012). *Thematic analysis.* (pp. 57–71).

Braun, V., & Clarke, V. (2019). Reflecting on reflexive thematic analysis. *Qualitative Research in Sport, Exercise and Health*, *11*(4), 589–597. https://doi.org/10.1080/2159676X.2019.1628806

Brock, J. K.-U., & von Wangenheim, F. (2019). Demystifying AI: What Digital Transformation Leaders Can Teach You about Realistic Artificial Intelligence. *California Management Review*, *61*(4), 110–134. https://doi.org/10.1177/1536504219865226

Cambon, A., Hecht, B., Edelman, B., Ngwe, D., Jaffe, S., Heger, A., Vorvoreanu, M., Peng, S., Hofman, J., Farach, A., Bermejo-Cano, M., Knudsen, E., Bono, J., Sanghavi, H., Spatharioti, S., Rothschild, D., Goldstein, D. G., Kalliamvakou, E., Cihon, P., … Teevan, J. (2023). *Early LLM-based Tools for Enterprise Information Workers Likely Provide Meaningful Boosts to Productivity* (Issue MSR-TR-2023-43). https://www.microsoft.com/en-us/research/publication/early-llm-based-tools-for-enterprise-information-workers-likely-provide-meaningful-boosts-to-productivity/

Capretz, L. F., & Ahmed, F. (2010). Making Sense of Software Development and Personality Types. *IT Professional*, *12*(1), 6–13. https://doi.org/10.1109/MITP.2010.33

Cascio, W. F., & Montealegre, R. (2016). How Technology Is Changing Work and Organizations. *Annual Review of Organizational Psychology and Organizational Behavior*, *3*(Volume 3, 2016), 349–375. https://doi.org/https://doi.org/10.1146/annurev-orgpsych-041015-062352

Coutinho, M., Marques, L., Santos, A., Dahia, M., França, C., & de Souza Santos, R. (2024). The Role of Generative AI in Software Development Productivity: A Pilot Case Study. *Proceedings of the 1st ACM International Conference on AI-Powered Software*, 131–138. https://doi.org/10.1145/3664646.3664773

Cui, K. Z., Demirer, M., Jaffe, S., Musolff, L., Peng, S., & Salz, T. (2024). The Productivity Effects of Generative AI: Evidence from a Field Experiment with GitHub Copilot. *An MIT Exploration of Generative AI*.

Directorate-General for Research and Innovation (European Commission), Hollanders, H., & Es-Sadki, N. (2023). *Regional Innovation Scoreboard 2023*.

Ebert, C., & Louridas, P. (2023). Generative AI for Software Practitioners. *IEEE Software*, *40*, 30–38. https://doi.org/10.1109/MS.2023.3265877

European Data Protection Supervisor. (2023). *Resolution on Generative Artificial Intelligence Systems*.

Exter, M. (2014). Comparing educational experiences and on-the-job needs of educational software designers. *Proceedings of the 45th ACM Technical Symposium on Computer Science Education*, 355–360. https://doi.org/10.1145/2538862.2538970

Gioia, D. A., Corley, K. G., & Hamilton, A. L. (2013). Seeking Qualitative Rigor in Inductive Research: Notes on the Gioia Methodology. *Organizational Research Methods*, *16*(1), 15–31. https://doi.org/10.1177/1094428112452151

Gorenek, T., & Kohont, A. (2019). *OF DIGITALIZATION: OPPORTUNITIES AND CHALLENGES FOR ORGANIZATIONS IN THE EURO-MEDITERRANEAN AREA*. https://api.semanticscholar.org/CorpusID:248660995

Gurcan, F., & Cagiltay, N. (2019). Big Data Software Engineering: Analysis of Knowledge Domains and Skill Sets Using LDA-Based Topic Modeling. *IEEE Access*, *PP*, 1. https://doi.org/10.1109/ACCESS.2019.2924075

Hadi, M. U., Al-Tashi, Q., Qureshi, R., Shah, A., Muneer, A., Irfan, M., Zafar, A., Shaikh, M., Akhtar, N., Wu, J., & Mirjalili, S. (2023). *Large Language Models: A Comprehensive Survey of its Applications, Challenges, Limitations, and Future Prospects*. https://doi.org/10.36227/techrxiv.23589741.v1

Hamdi, M., & Kim, L. D. (2023). A Prompt-Based Approach for Software Development. *2023 International Conference on Computational Science and Computational Intelligence (CSCI)*, 1612–1614. https://doi.org/10.1109/CSCI62032.2023.00267

Jaffe, S., Shah, N. P., Butler, J., Farach, A., Cambon, A., Hecht, B., Schwarz, M., & Teevan, J. (2024). *Generative AI in Real-World Workplaces* (Issue MSR-TR-2024-29). https://www.microsoft.com/en-us/research/publication/generative-ai-in-real-world-workplaces/

Khemka, M., & Houck, B. (2024). Towards Effective AI Support for Developers: A Survey of Desires and Concerns. *ACM Queue*, *22*(3). https://www.microsoft.com/en-us/research/publication/towards-effective-ai-support-for-developers-a-survey-of-desires-and-concerns/

Li, H., Su, J., Chen, Y., Li, Q., & Zhang, Z. (2023). *SheetCopilot: Bringing Software Productivity to the Next Level through Large Language Models*. https://arxiv.org/abs/2305.19308

Liu, Z., Tang, Y., Luo, X., Zhou, Y., & Zhang, L. F. (2024). No Need to Lift a Finger Anymore? Assessing the Quality of Code Generation by ChatGPT. *IEEE Transactions on Software Engineering*, *50*(6), 1548–1584. https://doi.org/10.1109/TSE.2024.3392499

Lukkarinen, H., & Tuomaala, E. (2024). *Digital economy rivals construction in Finland*.

Mäkinen, V. (2021). *OHJELMOINNIN HARJOITTELIJAN PÄIVÄKIRJA*. Turun AMK.

Maruping, L. M., & Matook, S. (2020). The evolution of software development orchestration: current state and an agenda for future research. *European Journal of Information Systems*, *29*(5), 443–457. https://doi.org/10.1080/0960085X.2020.1831834

Matinpuro, T. (2021). *OHJELMISTOKEHITTÄJÄN PÄIVÄKIRJA*. Oulun Ammattikorkeakoulu.

Mees-Buss, J., Welch, C., & Piekkari, R. (2022). From Templates to Heuristics: How and Why to Move Beyond the Gioia Methodology. *Organizational Research Methods*, *25*(2), 405–429. https://doi.org/10.1177/1094428120967716

Meyer, A. N., Barton, L. E., Murphy, G. C., Zimmermann, T., & Fritz, T. (2017). The Work Life of Developers: Activities, Switches and Perceived Productivity. *IEEE Transactions on Software Engineering*, *43*, 1178–1193. https://api.semanticscholar.org/CorpusID:35085514

Ogburn, W. F. (1957). Cultural lag as theory. *Sociology & Social Research*, *41*, 167–174.

Pappas, I. O., Mikalef, P., Giannakos, M. N., Krogstie, J., & Lekakos, G. (2018). Big data and business analytics ecosystems: paving the way towards digital transformation and sustainable societies. *Information Systems and E-Business Management*, *16*(3), 479–491. https://doi.org/10.1007/s10257-018-0377-z

Peng, S., Kalliamvakou, E., Cihon, P., & Demirer, M. (2023). *The Impact of AI on Developer Productivity: Evidence from GitHub Copilot*. https://arxiv.org/abs/2302.06590

Peng, X., Xing, Z., & Sun, J. (2019). AI-boosted software automation: learning from human pair programmers. *Science China. Information Sciences*, *62*(10), 200104. https://doi.org/https://doi.org/10.1007/s11432-018-9854-3

Porkka, J. (2018). *Koodarin päiväkirja*. Haaga-Helia ammattikorkeakoulu.

Saklamaeva, V., & Pavlič, L. (2024). The Potential of AI-Driven Assistants in Scaled Agile Software Development. *Applied Sciences*, *14*(1). https://doi.org/10.3390/app14010319

Sauvola, J., Tarkoma, S., Klemettinen, M., Riekki, J., & Doermann, D. (2024). Future of software development with generative AI. *Automated Software Engineering*, *31*(1). https://doi.org/10.1007/s10515-024-00426-z

Scaffidi, C. (2018). Employers' Needs for Computer Science, Information Technology and Software Engineering Skills Among New Graduates. *International Journal of Computer Science, Engineering and Information Technology*, *8*, 1–12. https://doi.org/10.5121/ijcseit.2018.8101

Seppälä, S. (2024). *Full stack-kehittäjän päiväkirja*. Tampereen ammattikorkeakoulu.

Simonen, J., Institute, M. A., Seppänen, V., Sohlo, S., & Svento, R. (2016). What happened to the growth?-The case of the ICT industry in Oulu, Finland Timo Koivumäki. In *Int. J. Entrepreneurship and Small Business* (Vol. 29, Issue 2).

Strauss, A., & Corbin, J. (1998). Basics of qualitative research: Techniques and procedures for developing grounded theory, 2nd ed. In *Basics of qualitative research: Techniques and procedures for developing grounded theory, 2nd ed.* Sage Publications, Inc.

Suomen virallinen tilasto (SVT). (2024). *Työssäkäynti*. Helsinki.

Tampere University. (2024a). *Conversational Interaction with Artificial Intelligence*. https://www.tuni.fi/en/students-guide/curriculum/course-units/otm-e9ac04d0-a88c-423c-bd44-3e9a9126dffa?year=2024

Tampere University. (2024b). *Human-Centred and Societal Perspectives to Artificial Intelligence*. Computing Sciences Studies. https://www.tuni.fi/en/students-guide/curriculum/course-units/otm-822390ae-eb09-4714-916a-2300881ff4ec?year=2024

Tampere University. (2024c). *Information Search and Generation with Large Language Models*. Communication Sciences Studies. https://www.tuni.fi/en/students-guide/curriculum/course-units/uta-ykoodi-48106?year=2024

Turunen, J. (2022). *OHJELMISTOKEHITTÄJÄN PÄIVÄKIRJA URAN ALKUVAIHEESTA*. Oulun ammattikorkeakoulu.

Vial, G. (2019). Understanding digital transformation: A review and a research agenda. *The Journal of Strategic Information Systems*, *28*(2), 118–144. https://doi.org/https://doi.org/10.1016/j.jsis.2019.01.003

Volti, R. (2004). William F. Ogburn "Social Change with Respect to Culture and Original Nature." *Technology and Culture*, *45*(2), 396–405. http://www.jstor.org/stable/40060750

Wiedemann, A., Wiesche, M., Gewald, H., & Krcmar, H. (2020). Understanding how DevOps aligns development and operations: a tripartite model of intra-IT alignment. *European Journal of Information Systems*, *29*(5), 458–473. https://doi.org/10.1080/0960085X.2020.1782277