An Analytical Study on the Effect of Large Language Model on the Unemployment of Software Engineers

by
Md Sanowar Hossain
CSE2301028147

Md. Mahmudul Hasan
CSE2301028185

MD. Salman Minna
CSE2301028083

Ringku Sutradhar
CSE2301028085

Md Minhaj Mustakim Ayat
CSE2301028164

Supervised by
Tanjil Mahmud

Submitted in partial fulfillment of the requirements for the degree of
Bachelor of Science in Computer Science and Engineering



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SONARGAON UNIVERSITY (SU)
September 2026

APPROVAL

The thesis titled "**An Analytical Study on the Effect of Large Language Model on the Unemployment of Software Engineers**" submitted by Md Sanowar Hossain (CSE2301028147), Md. Mahmudul Hasan (CSE2301028185), MD. Salman Minna (CSE2301028083), Ringku Sutradhar (CSE2301028085), Md Minhaj Mustakim Ayat (CSE2301028164) to the Department of Computer Science and Engineering, Sonargaon University (SU), has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering and approved as to its style and contents.

Board of Examiners

Tanjil Mahmud
Lecturer & Assistant Coordinator,
Department of Computer Science and Engineering
Sonargaon University (SU)

Supervisor

(Examiner Name and Signature)
Department of Computer Science and Engineering
Sonargaon University (SU)

Examiner 1

(Examiner Name and Signature)
Department of Computer Science and Engineering
Sonargaon University (SU)

Examiner 2

(Examiner Name and Signature)
Department of Computer Science and Engineering
Sonargaon University (SU)

Examiner 3

DECLARATION

We, hereby, declare that the work presented in this report is the outcome of the investigation performed by us under the supervision of **Tanjil Mahmud, Lecturer** and **Assistant Coordinator,** Department of Computer Science and Engineering, Sonargaon University, Dhaka, Bangladesh. We reaffirm that no part of this **An Analytical Study on the Effect of Large Language Model on the Unemployment of Software Engineers** has been or is being submitted elsewhere for the award of any degree or diploma.

Countersigned

(Tanjil Mahmud)
Supervisor

Signature

Md Sanowar Hossain
CSE2301028147

Md. Mahmudul Hasan
CSE2301028185

MD. Salman Minna
CSE2301028083

Ringku Sutradhar
CSE2301028085

Md Minhaj Mustakim Ayat
CSE2301028164

ABSTRACT

This dissertation examines the impact of Large Language Models (LLMs) on software engineers' joblessness, focusing on the debate over whether LLMs lead to technological unemployment or complement human work. The positivist, quantitative developmental study, with a cross-sectional design, was conducted using purposive sampling to obtain software and IT professionals and students in the amount of 63 people, who were then asked to complete a structured questionnaire followed by data analysis using reliability test, one-way ANOVA, Spearman correlation and multiple regression in SPSS. Research reveals that LLM use is widespread and common, and that there is a greater ‘risk of losing job' if using LLM than in using it if they are a student or senior employee, and that it is more common among engineers with less than one year of experience. The number of times they had used LLM was not the best independent predictor of risk perception, as was familiarity with the tools rather than experience. The results confirm a task substitution rather than whole occupation displacement trend and suggest that early career engineers, employers, and educators be targeted for LLM literacy to cope with the risk of employment due to continued industry-wide uptake of LLM use.

ACKNOWLEDHMENT

At the very beginning, we would like to express my deepest gratitude to the Almighty Allah for giving us the ability and the strength to finish the task successfully within the scheduled time.

We are grateful that we had the kind association as well as supervision of **Tanjil Mahmud**, Lecturer and Assistant Coordinator, Department of Computer Science and Engineering, Sonargaon University, whose heartfelt and valuable support with best concern and direction acted as a necessary resource to carry out our project.

We are also thankful to all our teachers during our whole education, for exposing us to the beauty of learning.

Finally, our deepest gratitude and love to my parents for their support, encouragement, and endless love.

TABLE OF CONTENTS

Title	Page No.
APPROVAL	ii
DECLARATION	iii
ABSTRACT	iv
ACKNOWLEDGEMENT	v
CHAPTER 1	1 - 7
INTRODUCTION	
1.1 Introduction: Artificial Intelligence and Large Language Models	1
1.2 Background of the Study: Evolution of AI and Natural Language Processing in Software Engineering	1
1.3 Problem Statement: Positive, Negative, and Neutral Consequences on the Workforce	2
1.4 Research Objectives	3
1.5 Research Questions	3
1.6 Significance of the Study	3
1.7 Research Scope and Limitations	4
1.8 Definition of Key Terms	5
1.9 Structure of the Thesis: A Roadmap of the Investigation	5
1.10 Chapter Summary	7
CHAPTER 2	8 - 19
LITERATURE REVIEW	
2.1 Introduction	8
2.2 Concept of Large Language Models	9
2.3 AI and Automation in Software Engineering	10
2.4 Technological Unemployment Theory	11
2.5 Labour Market Impact of Artificial Intelligence	14
2.6 Changing Role of Software Engineers	16
2.7 Emerging Employment Opportunities	17
2.8 Research Gap	18
2.9 Chapter Summary	19
CHAPTER 3	20 - 27
RESEARCH METHODOLOGY	
3.1 Introduction	20
3.2 Research Philosophy	20
3.3 Research Approach	21
3.4 Research Design	21
3.5 Population and Sampling Technique	22
3.6 Sample Size	23
3.7 Data Collection Method	24

3.8 Data Analysis Technique 25
3.9 Reliability and Validity 26
3.10 Ethical Considerations 26

CHAPTER 4 28 - 43

RESULTS AND ANALYSIS

4.1 Introduction 28
4.2 Response Rate and Sample Profile 28
4.3 LLM Awareness and Usage Patterns 31
4.4 Reliability of the Measurement Scales 33
4.5 Descriptive Analysis of Perceived Benefit and Risk 34
4.6 Inferential Analysis 35
4.7 Summary of Key Findings 42

CHAPTER 5 44 - 48

DISCUSSION

5.1 Introduction 44
5.2 Discussion by Research Objective 44
5.3 Theoretical Implications: Technological Unemployment and Skill-Biased Technological Change 48

CHAPTER 6 49 - 51

CONCLUSION AND RECOMMENDATIONS

6.1 Introduction 49
6.2 Summary of the Study 49
6.3 Conclusion 49
6.4 Recommendations 50
6.5 Limitations of the Study 50
6.6 Suggestions for Future Research 51

LIST OF TABLES

Table No.	Title	Page No.
Table 4.1	Sample Profile of Respondents (N = 63)	31
Table 4.2	LLM Familiarity, Usage Frequency, and Work Share	32
Table 4.3	Reliability Scales	33
Table 4.4	Descriptive Statistics for the Eleven Attitude Statements	35
Table 4.5	Descriptive Statistics: Perceived Risk Index by Experience Group	36
Table 4.6	One-Way ANOVA: Perceived Risk Index by Experience	36
Table 4.7	Tukey HSD Post Hoc Comparisons: Perceived Risk Index	37
Table 4.8	Descriptive Statistics: Productivity Benefit Index by Experience Group	38
Table 4.9	One-Way ANOVA: Productivity Benefit Index by Experience	39
Table 4.10	Spearman Correlation Coefficients Between Key Variables	40
Table 4.11	Regression Model Summary	41
Table 4.12	Regression Coefficients: Predictors of Perceived Risk Index	42

LIST OF FIGURES

Figure No.	Title	Page No.
Fig 2.1	Growth in Developer Use and Planned Use of AI Tools	8
Fig 2.2	Current AI Tool Use Frequency in Development Process, 2025	9
Fig 2.3	GitHub Copilot Experiment: Average Coding Task Completion Time	10
Fig 2.4	Potential Labour-Market Exposure to LLMs	12
Fig 2.5	AI-Assisted Software Development Outcomes, 2025	15
Fig 2.6	Projected Employment Growth, 2024–2034	18
Fig 3.1	Population and Sampling Technique	22
Fig 3.2	Data Collection Method	24
Fig 3.3	Data Analysis Technique	25
Fig 4.1	Relationship of the software	29
Fig 4.2	Age Distribution	29
Fig 4.3	Employment Status of Respondents	30
Fig 4.4	Professional Experience of Respondents	30
Fig 4.5	Reliability Scales	33
Fig 4.6	Analysis of Perceived Benefit and Risk	34
Fig 4.7	Experience on Perceived Risk	36
Fig 4.8	Productivity Benefit	38
Fig 4.9	Correlation Analysis	40
Fig 4.10	Predicting Perceived Risk	41
Fig 5.1	Impact of LLM Adoption on Software Engineers' Employment Opportunities	45
Fig 5.2	Software Engineers Are Most Affected by LLM Incorporation	46

CHAPTER 1

INTRODUCTION

1.1 Introduction: Artificial Intelligence and Large Language Models

The fast growth of Artificial Intelligence (AI) has dramatically changed many industries, and the most prominent one is the advent of Large Language Models (LLMs). These intelligent AI systems, which have the capability of comprehending and producing human-like text, are being incorporated more and more within the software development process [1]. Code generation, debugging, and documentation have all started to change the traditional role of software engineers due to the introduction of LLMs.

The introduction of the LLMs into software engineering development has also posed a serious concern for the future of the software engineering profession. Although other experts claim that the technologies are helping to increase productivity and offer new opportunities, there are also concerns about the possible job displacement and routine programming tasks being automated [2]. This ambivalent view enables us to possess a tripartite account of the effect of the LLMs on the workforce market.

This dissertation will aim to explore the effects of Large Language Models on the unemployment of software engineers. It explores the idea of the people being threatened by the emergence of the LLMs (as they substitute) or it is a complementary resource, which enhances human capacities [5].

1.2 Background of the Study: Evolution of AI and Natural Language Processing in Software Engineering

The last decade has witnessed the rapid development of AI technologies, and one of the areas in which it has been applied is the field of natural language processing (NLP). The GPT-based systems, in particular, have shown impressive skills in code generation, documentation generation, as well as, in addressing complicated program-related issues [8]. They are trained on large volumes of data and can even execute the tasks that were once performed by highly skilled human workers.

Previously, the software engineering profession has been deemed to be immune to automation as it is based on problem solving, creativity, and complicated decision making. But the

introduction of LLMs has come to refute this assumption. The tools that run on LLMs can create snippets of code, routines, and even entire applications, and can be used even when the user is asking them to generate such snippets.

The large technology companies and start-up adoption have also contributed to the process of increased adoption of the technologies, which introduces efficiency and reduces time to develop. Nonetheless, this effectiveness can be at a price [10]. Due to the attempts of organisations to maximise resources, the change in the employment pattern can happen since the entry-level and middle-level software engineers can be less in demand with the implementation of the LLMs.

In the meantime, new jobs are emerging, such as AI experts, prompt engineers, and AI-system trainers. This implies that as some of the traditional roles will be on the decline, some will change or be invented [13].

1.3 Problem Statement: Positive, Negative, and Neutral Consequences on the Workforce

The increasing importance of Large Language Models in the software development lifecycle has raised the question of the future of software engineer jobs. However, unlike the latter, where the development process is less time-consuming, and the effectiveness of the latter is higher, they can also replace the work of a human developer [14].

It poses a very important question: will the introduction of LLMs lead to the high unemployment of software engineers, or will it cause the nature of their work to be different, without making the unemployment rate high?

In addition, the effects of LLMs might not apply equally within the board as far as all the degrees of expertise are concerned [20]. The repetitive nature of entry-level developers predisposes them to automation, and therefore, they might not be as resistant to automation as senior developers, who will most likely be needed to do system design and make decisions.

Thus, the issue of comprehending how much Large Language Models influence unemployment among software engineers and whether the consequences are negative, positive, or neutral will be solved in this study.

1.4 Research Objectives

The main aim of the study is to examine how Large Language Models can influence the unemployment of software engineers.

The particular aims are:

- To understand how LLMs can be used in contemporary software development.
- To understand the degree of automation of software engineering by LLMs.
- To experiment with the impact of the adoption of LLM on software engineers' employment opportunities.
- To determine the type of software engineers who will be influenced the most by the incorporation of LLM.
- To explore whether the technology industry has created any new employment for LLMs.
- To offer suggestions on how to adapt to AI-driven changes in the job market.

1.5 Research Questions

The research questions that are used in the current research are:

1. What is a Large Language Model in software development?
2. How far are LLMs automating the activities that software engineers perform?
3. How does the adoption of the LLM affect the software engineers' unemployment rate?
4. What are the most at-risk software engineers that the LLMs will take over?
5. Do Large Language Models bring about new jobs in the software sector?
6. How can software engineers be of use in the world of AI?

1.6 Significance of the Study

The research is of great importance, both in theory and in practice.

Academically, the research makes a contribution to the research area in the connection of Artificial Intelligence and labour economics. Although automation in manufacturing and service sectors has been a topic of many studies, there has been scant research on the implications of LLM on the employment of software engineers [24]. The specified work plugs that void since it offers a closer examination of the impact that AI-based tools have on a profession that is highly skilled.

It gives the organisations a clue on how to integrate the use of LLMs without having a negative impact on the stability of the workforce [25]. The research catches the eye of policymakers on the possible disruption of the labour market and the necessity to introduce reskilling and upskilling programs.

Moreover, the research comes at the right time, at a time when the world is going digital, and the use of AI is becoming more widespread within industries. Knowledge of its effects on employment will aid in coming up with strategies that will allow both technological progress and economic stability.

1.7 Research Scope and Limitations

This research paper tries to target the impact of Large Language Models on software engineering jobs [26]. The definition of the scope is based on various dimensions and in the following manner:

1.7.1 Target Population and Participant Profile

The focal point of the study is the software engineers of the various levels of experience (entry-level, mid-level, and top-level employees). It also takes into account those who work in other industries like technology companies, start-ups, and freelance setups [27]. The research may be on a global scale; however, the particular areas where AI is rapidly being adopted can be given special attention.

1.7.2 Key Research Themes

The studies are founded based on the following topics: Artificial Intelligence, automation, employment, and software development. It particularly looks at the impact of the introduction of the LLM on job descriptions, skills, and job trends in software engineering [28].

1.7.3 Geographic Focus of the Study

It is analysed on a global basis, taking into consideration the tendencies in the developed countries and the developing countries. Nonetheless, it can prioritise more those areas that have embraced AI on a grand scale, such as North America, Europe, and some sections of Asia, to present relevant and current data.

1.7.4 Theoretical Framework

It is premised on the technological unemployment theories, the skills-based technological change theories, and the innovation economics [31]. The impact of technological changes on

the labour markets and patterns of employment can be appreciated using these theoretical models.

1.7.5 Practical Implications and Application Scope

The paper gives effective information on how entities and individuals might adjust to the growing utilisation of LLMs. It provides suggestions on how skills development should be accomplished, workforce planning, and the effective implementation of AI technologies into the software development process [27].

1.8 Definition of Key Terms

To be clear and consistent, the following terms are the key ones that are defined:

Large Language Models (LLMs): An innovative type of AI that is trained on massive amounts of data to understand how to process and generate human-like text and code.

Software engineers: Software system engineers, software designers, software developers, software testers, and software maintainers [26].

Unemployment: It is a condition in which working people cannot find employment.

Automation: Technological application to do work with minimum human involvement.

Artificial Intelligence (AI): The simulation of the process of human intelligence by the machine, especially computer systems [25].

Job Displacement: A job displacement due to technology or other structural forces.

Reskilling: It entails the acquisition of new skills so as to adjust to the change in job requirements.

1.9 Structure of the Thesis: A Roadmap of the Investigation

Such a dissertation has several chapters in which the research topic is subdivided to present a thorough analysis of the research topic.

Chapter 1: Introduction:

This chapter sets the research groundwork by exploring the emergence and impact of Artificial Intelligence and Large Language Models (LLMs) in the software engineering field. It outlines the background of the study, problem to be solved, research objectives, research questions, significance of the study, scope and limitations, and key words used in the study and how the

thesis will be organized. The chapter lays the groundwork to the analysis of whether LLMs pose a threat of unemployment, evolve software engineering jobs, or provide new opportunities.

Chapter 2: Literature Review:

This chapter examines the previous academic works pertaining to Large Language Models, AI automation, technological unemployment, labour market transformation, and software engineering job market. It is an in-depth discussion of the use of LLMs in code generation, code debugging, code testing, code documentation, etc. in software development. Another point discussed in the chapter is that, there could be different impacts of AI on junior and senior software engineers, and this study will fill the conceptual gap.

Chapter 3: Research Methodology:

This chapter details the research design and methods employed to carry out the research. It includes the target population, sampling, data collection, research instrument, data analysis, and ethics. The chapter shows how survey data will be collected and analysed to understand the relationship between LLM adoption and software engineers' unemployment risk.

Chapter 4: Results and Analysis:

This chapter contains the results that were garnered from the gathered data. These responses are analysed to find patterns concerning LLM usage, task automation, productivity changes, job concerns, skills requirements, and perceived unemployment risk of software engineers. Using suitable statistical/descriptive analysis, the chapter offers a structured interpretation of the data.

Chapter 5: Discussion:

The result of the research is discussed in this chapter based on the literature that reviewed in Chapter 2. It helps to understand if the findings do support or challenge the current theories on Technological Unemployment, Skill-Biased Technological Change and Labour Market transformation. The chapter also elaborates on the ramifications of LLM use for junior and senior software developers, employers, schools and universities, and policymakers.

Chapter 6: Conclusion and Recommendations:

This is the final chapter and summarises the key findings of the study and draws conclusions in line with the research objectives and research questions. It contains actionable tips for software engineers, organisations, universities and policy makers to adjust to the AI reshaped labour market. The chapter further identifies the study's limitations, and makes suggestions for further areas of study.

1.10 Chapter Summary

The chapter has unveiled the theme of the research by showing how Large Language Models are increasingly becoming powerful in software engineering. It has provided a context in detailing the disruptive aspect of AI technologies and the appropriate job fears.

The problem statement has identified the uncertainty in terms of the effects of the use of LLMs on the employment of software engineers, and the research objectives and research questions have provided the research scope. The contribution of the study in terms of scholarly and practical contributions has been discussed.

Moreover, the study area has been well defined, both in demographics, themes, geographically, theoretically, and practically. It has employed operationally defined key terms to simplify it, and the general shape of the dissertation has been provided.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

The fast development of Artificial Intelligence (AI) has also brought a great revolution in the world of numerous industries such as healthcare, education, finance, and software engineering. The appearance of Large Language Models (LLMs) is one of the most powerful trends in AI, as they are complex systems, able to comprehend and generate human speech, as well as process it. ChatGPT and GitHub Copilot are all examples of LLMs that have found their way into the software development environment. Some of the tasks that these AI-based tools can help developers with include code generation, debugging, documentation, and problem-solving. Their increasing use has enhanced productivity and lessened the time spent on repetitive programming tasks. Nevertheless, the growing popularity of LLMs has also raised concerns about their possible effects on jobs in the software engineering field. The chapter provides a review of the existing literature on Large Language Models, automation in software engineering, technological unemployment, the transformation of the labour market, and the changing role of software engineers. The literature review examines how AI-driven technologies can affect the employment trends by not only automating some software development processes but also establishing new skills and professional opportunities.

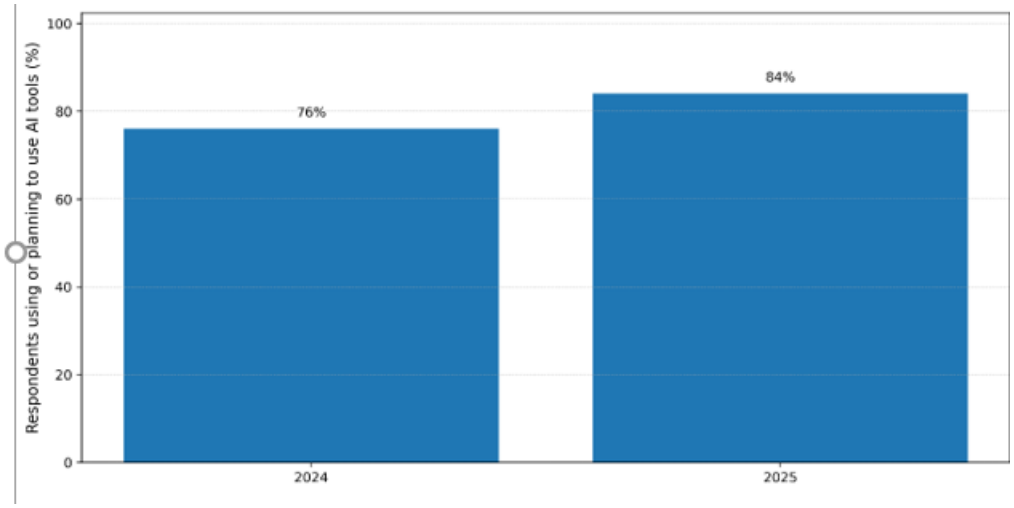


Fig 2.1: Growth in Developer Use and Planned Use of AI Tools

2.2 Concept of Large Language Models

Large Language models (LLMs) are sophisticated artificial intelligence (AI) models that process and comprehend human language and also generate it in a human-like manner. Deep learning methods are used to create these models, and they are trained on large datasets comprising books, websites, programming code, research articles and other text-based sources. The transformer architecture used in LLAMs allows them to identify patterns, context, and relationships in language [11]. Because of this, they are capable of doing a vast range of tasks like text generation, summarisation, translation, question answering and code creation. They are very useful both in school and the workplace due to their rapidity in processing large volumes of information [32]. Using AI-driven systems, it is possible to generate code snippets automatically, explain programming concepts, identify software bugs, propose corrections, and write documentation [18]. These functionalities have helped in increasing the use of AI tools within the software development environment. The assistants, which are coded with the help of LLM technology, assist a developer to spend less time on repetitive and routine tasks. Consequently, software engineers have the opportunity to concentrate more on solving more complex problems, system design and innovations instead of wasting a lot of time on manual code work.

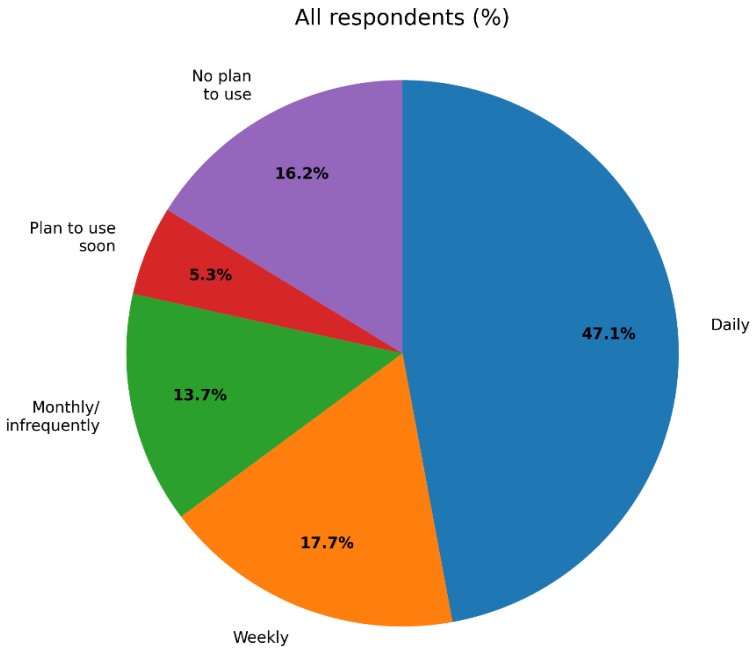


Fig 2.2: Current AI Tool Use Frequency in Development Process, 2025

Along with these benefits, researchers report that LLMs still have significant drawbacks. They can give faulty results, misinterpret, or provide invalid coding recommendations. Moreover,

LLMs frequently fail tasks which involve deep reasoning, creativity, ethical decision-making and domain-specific knowledge [15]. To achieve quality, accuracy and security, human supervision is still necessary for the development of software.

2.3 AI and Automation in Software Engineering

Automation has been a significant factor in enhancing production in the industry by enhancing efficiency and minimising manual labour. Automation in the previous decades was primarily applied to manufacturing and physical production sectors. Nevertheless, with the development of Artificial Intelligence (AI), automation has spread into knowledge-oriented careers, such as software engineering. Software engineering has traditionally been regarded as a profession that could not be automated due to its need to be creative, analytical, possess technical skills and be able to solve complex problems [16]. With the advent of Large Language Models (LLMs) and AI-based code-writing tools, this assumption has been questioned, as some software development work can now be automated partially.

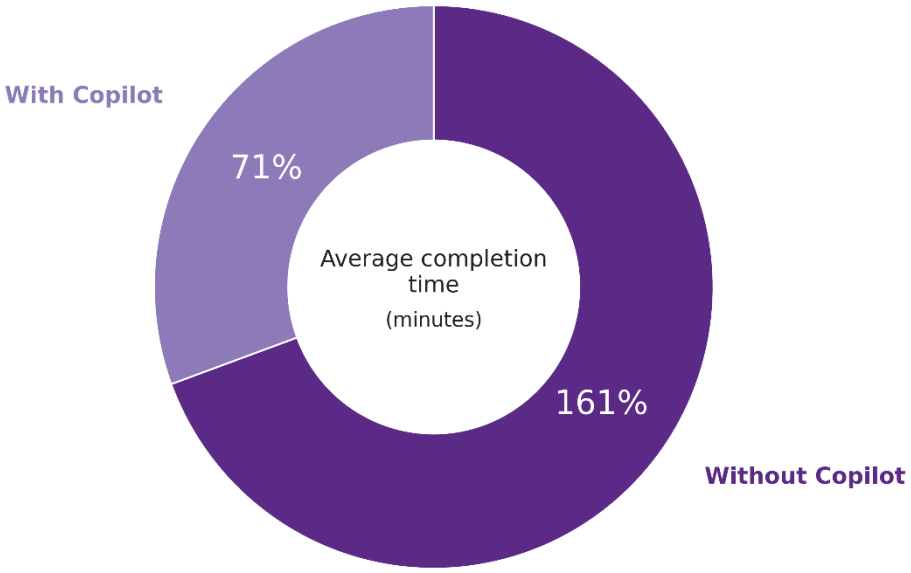


Fig 2.3: GitHub Copilot Experiment: Average Coding Task Completion Time

Studies indicate AI technologies can automate a host of activities related to programming. These involve code generation, automatic software testing, debugging, code review, documentation writing and maintenance support [6]. The use of LLM-based tools is rapidly becoming popular among software companies to shorten the software development cycle,

decrease operational expenses, and improve the efficiency of the workflow. This makes AI an essential aspect of software engineering today [29].

Most scholars believe that automation in software engineering is not about to end the job of software engineers but redefine their job descriptions. Automation can be more prone to entry-level developers since they often do repetitive coding, testing, and debugging. On the contrary, senior software engineers continue to be critical in undertaking roles like software architecture, project management, strategic planning, and decision-making [2]. Such advanced positions demand the ability of human judgment, creativity and communication that cannot be completely modelled by AI systems.

In addition, some research indicates that automation can bring about a gulf between low-skill and high-skill positions in software engineering. Monotonous and unsurprising jobs are being automated, and higher-level jobs like professional duties are still left to human skills [3].

2.4 Technological Unemployment Theory

Technological unemployment is a phenomenon whereby technological development lowers the need for human labour as it substitutes roles that were played by employees. The idea has long been debated in economics and labour studies, as well as in industrial sociology. Technological innovation has historically reshaped the labour market by adding machines, automation systems, and digital tools to the labour market, which make the labour market more efficient and less dependent on manual labour [19]. One of the earlier instances of technological unemployment is presented by the industrial revolution, as machines supplanted numerous customary manufacturing jobs. The same issues were raised when computers and digital technologies began to develop, changing the administrative, communication, and service-based jobs. Artificial Intelligence (AI) and Large Language Models (LLMs) are also a new form of automation in the modern age that not only impacts manual labour but also knowledge-based jobs [7]. Contrary to conventional automation, which was mainly aimed at automating repetitive physical processes, the AI systems now can process cognitively, including language processing, problem-solving and decision-making assistance. This has cast doubts on whether the highly skilled jobs, such as software engineering, can face employment disturbance.

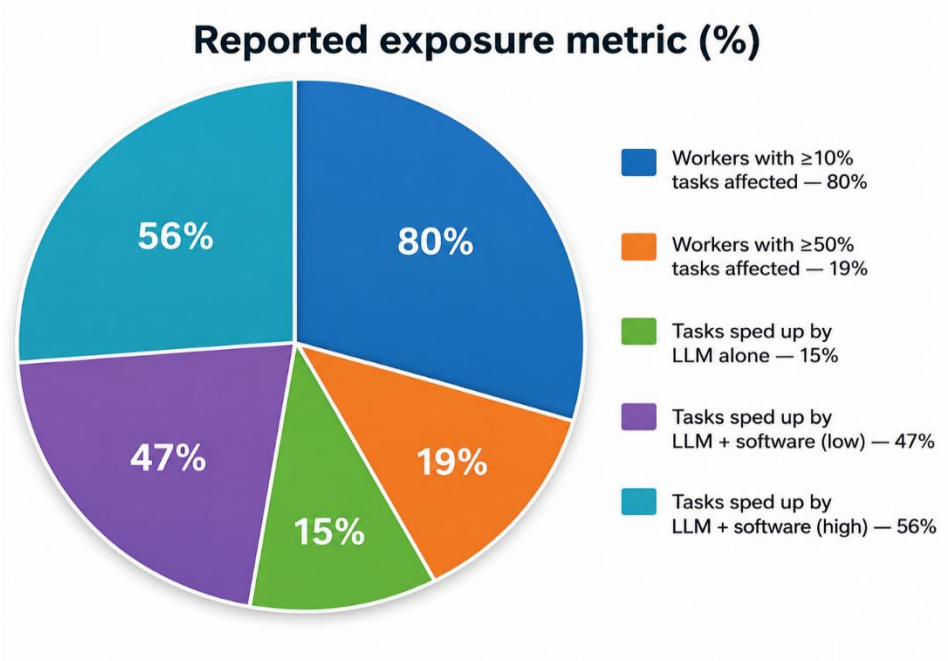


Fig 2.4: Potential Labour-Market Exposure to LLMs

Technological unemployment theory posits that with the development of machines that can perform tasks more effectively, organisations can decrease the number of required labour in order to keep operational costs as low as possible. Technology is commonly used in businesses to enhance efficiency, minimise errors, and maximise productivity [22]. Software engineering, Software projects, Software LLM-based systems can be used to create software, detect software bugs, automate testing, and aid in technical documentation. The abilities suggest that some software development tasks might not need as much human intervention as previously.

Researchers insist that the software engineering career is particularly applicable in the analysis of technological unemployment since it is on the frontline of creating technology and being affected by technological change. Conventionally, software engineers were regarded as rather resistant to automation because of the difficulty of the programming work. Coding involves logical thinking, innovation, interpersonal skills, and systems thinking. Nevertheless, this belief has been disproven by the rapid development of AI-based coding assistants, which have shown that a great number of programming tasks can be partially automated [14]. One of the main issues with technological unemployment theory is task substitution. Task substitution is where technology replaces specific activities and not a whole occupation. In software engineering, lower-level tasks such as repetitive coding, bug fixing and documentation can be automated, but more strategic tasks continue to be human-involved. This implies that

unemployment does not necessarily come in the form of complete removal but is achieved by a gradual decline in labour demand of given types of workers [23].

The entry-level software engineers can be considered especially susceptible in this theoretical construct. Junior developers are usually involved in repetitive and structured programming tasks that can be done with the help of AI. Writing boilerplate code, debugging standard errors, and performing regular testing are some of the tasks that are becoming more and more assisted by AI-powered coding assistants [17]. With the introduction of such technologies by organisations, there might be a reduced number of junior-level employees to perform the same workload. Nonetheless, technological unemployment theory also appreciates the fact that labour displacement does not necessarily lead to long-term unemployment. Adaptation to technological change in labour markets tends to take the form of economic restructuring, skills transformation, and occupational transformation. Employees might move to newer jobs that are brought about by advances in technology [30].

Machine learning engineers, AI ethics specialists, data scientists, prompt engineers, and AI governance professionals are just some of the new positions emerging in the field of AI. These jobs were not as large before the emergence of sophisticated AI systems. Hence, according to some economists, technological unemployment can be short-term, as opposed to being permanent [9]. Unemployed people displaced in one region have the potential to be employed in new industries. The next concept that is significant in the technological unemployment theory is skill-biased technological change. According to this theory, the development of technology directly lowers the chances of lower-skilled employees and provides more opportunities to highly skilled workers [4]. Software engineers with high levels of AI skills, expertise in system architecture, and cross-disciplinary skills can be in higher demand.

Opponents of the technological unemployment theory believe that technology tends to complement and not to substitute human labour. There is still a need for supervision, validation, ethical supervision and understanding of contexts in AI systems. No, Large Language Models can be used to produce code in a short time, but they can also be inaccurate, insecure, or half-baked. There is still a need for human engineers who would analyse quality, provide security and integrate systems [21]. Moreover, AI can be used by organisations to increase their productivity instead of cutting down on their staff. Rather than sidelining developers, firms can empower engineers to be more productive, and teams will develop more software in shorter timeframes. This is one of how AI can be seen as an increase in productivity, not a substitute for labour. The correlation between the adoption of LLM and unemployment is complicated

by the fact that technological change has a different influence on industries. Employment outcomes are affected by the economic conditions, organisational strategy, workforce skills and regulatory environments [12].

2.5 Labour Market Impact of Artificial Intelligence

AI has emerged as one of the technological changes that has had the most significant impact on the world's labour markets. The growing use of AI systems in industries has changed the way production is carried out, decision-making, and the organisation of the workforce. The issue of whether AI is a job creator, job killer, or labour pattern redefiner remains a topic of controversy among labour economists and researchers. With the emergence of Large Language Models, this debate has been exacerbated as the systems have the capability to do what used to be done by highly educated professionals. Labour market impact is the effect of the technological innovation on the employment demand, skill requirement, wages, productivity and the occupational structures. The impact of AI technologies on industries can vary in relation to the complexity of the tasks, the possibility of automation, and the organisational adoption strategies [11].

AI tools have brought massive transformations to the dynamics of the labour market in the context of software engineering. LLM-powered coding assistants can build code at high speed, help debug it, automatic documentation, and recommend development solutions. These tools save time required in routine programming activities, and companies are able to save on efficiency [18]. Consequently, companies will re-evaluate workforce demands and change their recruitment practices. Studies have shown that the labour market can be affected the most by junior-level software engineers. Coding tasks can usually be repetitive and highly structured, and they are usually done by entry-level developers.

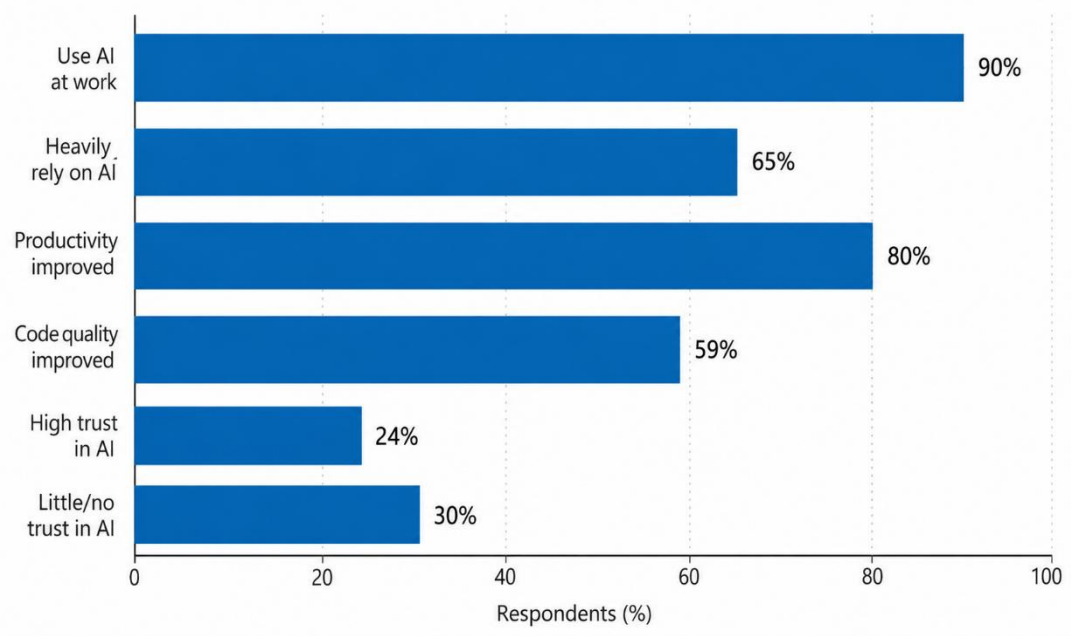


Fig 2.5: AI-Assisted Software Development Outcomes, 2025

Not all software engineering jobs are equally impacted by AI. Complex tasks that still require senior engineers would be software architectureS, project planning, strategic decision making and communication with stakeholders. These tasks include contextual knowledge and interdisciplinary thought, which cannot be completely simulated by AI systems [15]. The labour market effects, therefore, can differ considerably in terms of the level of skills and professional experience. The nature of the skills that are required by employers also varies because of AI. The knowledge of traditional coding might not be adequate in a highly automated development environment. Software engineers who know how to incorporate AI, machine learning, cloud computing, cybersecurity, and how to work with humans are becoming the desired employees of employers. This change in the skill demand helps in the transformation of the labour market instead of just job eradication [16].

Productivity growth is another significant impact of the labour market. The efficiency of AI-powered systems can be enhanced by saving time on repetitive tasks. Increased productivity can lead to economic growth, allowing businesses to increase their activities and open up new positions. Software-producing businesses will be able to release new products, enter new markets, and recruit workers with specialised roles [6]. Other economists claim that AI could lead to a polarisation of jobs in the field of software engineering. Job polarisation is the division of the labour market between high- and low-skilled jobs, and the middle-range routine jobs are decreasing. Professionals with a high level of specialisation have improved chances of being employed, but employees who engage in repetitive technical tasks may find it challenging [29].

The organisational approach of AI implementation equally has a significant influence on employment outcomes. Those organisations that adhere to using AI as a team working tool can similarly boost the performance of the employees without laying off employees. Such applications of AI are a supporting technology which enhances the workflow. On the contrary, organisations that support cost-reduction can employ AI to reduce the working personnel [2]. The effects of global labour markets might also vary across geographical locations. Developed economies with powerful digital infrastructure can accept AI at a quicker rate, which translates to labour at a quicker rate. The rate of adoption in newer economies will be slower, but they will face problems of reskilling their employees and technological access.

Researchers also indicate that the use of AI can transform the wage hierarchy within the software engineering field. The salaries of specialists with knowledge related to AI can be raised, given the already advanced demand. Meanwhile, the jobs tied to mainstream programming might be held back either in salary or even eliminated [3]. The long-term results in terms of adaptation are determined by the labour market. One of the most significant roles to be played in the development of employees trained towards change regarding AI is for schools, universities, and government officials. The incumbent software engineers can be trained via training programs, reskilling programs and lifelong learning to keep abreast with the dynamic labour environments. On the whole, AI possesses certain threats and opportunities as far as employment in the field of software engineering is concerned [19].

2.6 Changing Role of Software Engineers

Technological innovation has transformed the role of software engineers significantly in the last few decades. Historically, software engineers worked on the design, code writing, testing, maintenance and debugging of software systems. The programming was manual, and the software development works were centred on manual writing line by line of code. Nevertheless, the advent of AI-driven development tools has greatly transformed these roles. Large Language Models have brought new workflows to software engineering settings [7]. The AI systems are becoming more common among developers to create code, propose solutions, and automate repetitive development processes. Instead of creating all elements of software manually, engineers have been working with smart tools to enhance efficiency.

Among the greatest changes that have been made, there is the replacement of code construction with managing the code. Software engineers are progressively becoming the screening systems for AI products. The developers are not left to build the entire code, and in fact, they read,

review and verify the suggestions given by the AI systems [22]. This involves proper judgment and evaluation. The generated code by AI can also include defects, bugs or errors in the code. The check into software quality, performance and security is still based on human checks. Programmers should ensure that technological and organisational requirements are met by the products of AI. The other drastic change is the focus on problem-solving and system-level thinking. The architecture design, integration, user requirements and innovation all spend more time by the engineers as the routine programming tasks are automated with the help of the AI [14]. AI is incapable of such tasks and needs imagination and experience.

Researchers indicate that today, software engineers have to be in a position to come up with more competencies. Any experience with the old model of coding would be sensible, and, once more, new capabilities are gradually gaining importance. They include such concepts as AI collaboration, in-time engineering, ethical reasoning, communication, information interpretation and interdisciplinary problem-solving.

The developer productivity might also be enhanced by AI tools. Tasks which took hours to code now can be done faster with the help of AI. This will enable engineers to spend more time on innovation, experimentation and planning [17]. Even though automation is a concern, several researchers suggest that software engineers will not be eliminated. Rather, transformation is being experienced in the profession. There is still a need for human skills in comprehending user needs, ethical development, and overcoming difficult technical challenges.

The evolving nature of software engineers shows that technological advancements tend to alter jobs instead of destroying them altogether [30].

2.7 Emerging Employment Opportunities

Despite the fact that the issue of job displacement is still salient, the use of AI has also resulted in the emergence of new jobs. Historically, technological innovation has generated new jobs and de-skilled simultaneously, and AI is not an exception. With the implementation of systems based on the use of LLM, the number of specialised positions demanded is growing. Another area with a swift increase is AI development and maintenance. AI systems are designed, trained, monitored and enhanced by professionals [9].

Another growth field of employment is AI governance and ethics. Companies are becoming more aware of the necessity to consider fairness, bias, transparency, and accountability in AI systems. Ethical AI professionals make certain that technology is implemented in a responsible

manner [4]. AI-related software engineers can become interdisciplinary as well. With data science, cybersecurity, cloud computing, or business strategy, combined with software development knowledge, there are a variety of career opportunities. The expansion of AI has led to the creation of a need to hire individuals who can assess the outputs of AI. Human reviewers will be necessary to promote reliability, accuracy, and compliance. The importance of quality assurance specialists and AI auditors can increase [21].

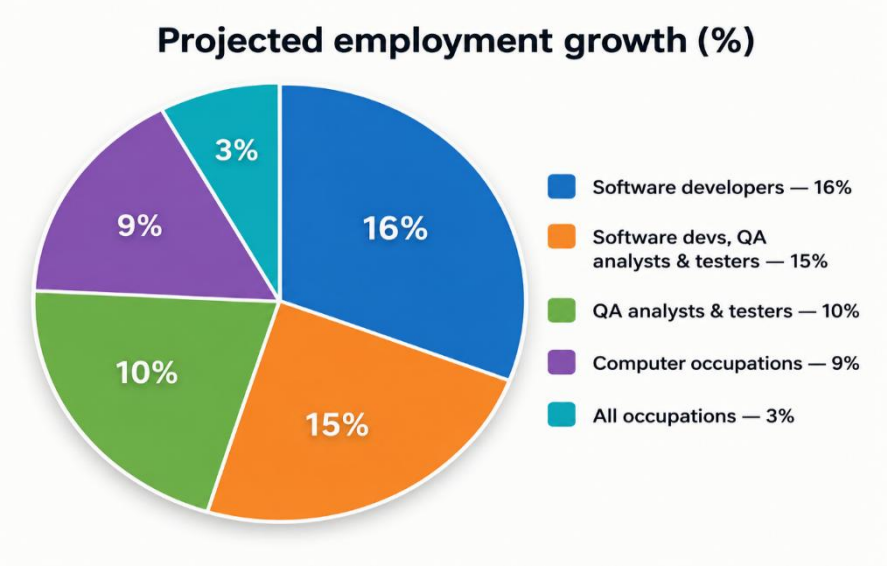


Figure 2.6: Projected Employment Growth, 2024–2034

Researchers say adaptation via constant education is essential in order to take advantage of newly arising opportunities. Career growth instead of unemployment may be experienced by engineers who develop new skills. The advent of AI jobs proves that technological transformation tends to provide parallelism in the employment trajectory [12].

2.8 Research Gap

Despite the growing popularity of Artificial Intelligence (AI) and workforce numbers in recent years, there is a lack of studies that specifically focus on the correlation between Large Language Models (LLMs) and unemployment among software engineers. The vast majority of current research is devoted to the overall tendencies of automation, the increase in productivity, and the overall effect of AI on labour markets [11]. Although these studies offer valuable inputs on technological change, they fail to give sufficient details on how the adoption of LLM can be directly related to employment in the software engineering career [18]. Existing literature has less empirical data on whether the use of LLMs is associated with quantifiable unemployment in software engineers. The opportunity of AI to automate the process of coding,

debugging, and documentation is discussed by numerous scholars, but there is a lack of data on the factual implications of these technologies. Current evidence usually brings about opposing views. There is a view that AI can lead to fewer jobs as it is able to perform repetitive programming tasks, but others believe that AI is more likely to alter the nature of the job and open new positions.

The other important gap is related to the different levels of software engineering experience in the context of unequal effects of automation. The exposure to automation risks can vary among entry-level developers, middle-level professionals, and senior engineers. Nevertheless, not many studies present in-depth comparisons of these groups [15]. In addition, the geographic variations are still under-explored, where AI implementation is uneven based on the technological infrastructure, labour policies, education systems, and economic development. It is also limited to long-term analysis since the LLM technologies are relatively new. The majority of studies concentrate on the short-term productivity impacts as opposed to long-term employment outcomes [16].

2.9 Chapter Summary

This chapter examined the current literature on the topic of Large Language Models, automation, technological unemployment, labour market transformation and the changing role of software engineers. The review found that LLMs have gained more and more power in software development through automating the code, debugging, and documentation. The available literature indicates that AI can decrease the need for regular programming operations, but it also opens new technical jobs. According to the literature, there is no guarantee that employment in software engineering will be eliminated, but this could be transformed. The technological unemployment and labour market adaptation theories offer significant insights into interpreting these changes. Nevertheless, there are still serious gaps in terms of the direct correlation between the LLMs and software engineer unemployment. Thus, more research is needed to comprehend the role of AI adoption on various groups of software professionals.

CHAPTER 3

RESEARCH METHODOLOGY

3.1 Introduction

In this chapter, the procedure followed in the research on the impact of Large Language Models (LLMs) on the unemployment of software engineers is detailed. It is based on the principles and results of theoretical understanding laid out in Chapter 2 and especially technological unemployment theory and skill-biased technological change, and it converts the research goals and research questions of Chapter 1 into a practical research design. The chapter defend the research philosophy, research approach, research design, research population, sampling method, sample size, data collection tool and description of data analysis protocol used to study the effect of adoption of LLM on employment outcomes of software professionals with varying levels of experience before culminating in a discussion of the reliability of the study, its validity and safeguarding against ethical considerations.

3.2 Research Philosophy

This research embraces the positivist research philosophy. Positivism is the view that the social world can be perceived in an objective manner, which can be quantified based on structured and measurable tools, and that knowledge must be the result of observable phenomena and logical arguments and not based on subjective understanding [33]. Positivist scientists carry on established theory to hypothesis and observation to obtain patterns and relationships that may be statistically validated [42].

A positivist position is appropriate in this study since the perceived employment risk in relation to the adoption of LLM is a construct to be measured as opposed to being a subjective experience. This operationalises the constructs found in the literature review, e.g., task automation, improved productivity and the perceived risk of job displacement, into closed-ended, Likert-scaled questions, and tests statistically whether they vary with experience levels and levels of use, as motivated by the positivist assumption that generalisable knowledge can be generated through structured measurement, rather than interpretive inquiry [42]. An interpretivist alternative based on open-ended interviews and meaning-driven open theories was deemed inappropriate, as the research questions demand the establishment of relationships

between specified variables, instead of an attempt to uncover individual and self-constructed meanings.

3.3 Research Approach

Quantitative research methodology was followed, which is in line with the above positivist philosophy. The quantitative is a deductive one: it starts with the theoretical propositions discussed in Chapter 2, including the varying exposures of junior and senior engineers to automation, and tries to test them with numerical data [34]. This takes the research out of the research gap (i.e., the parsimonious and merely case-study-based evidence) found in the literature review and generates statistically based evidence of the perceived risk, scale, and distribution of LLM-driven unemployment by software professionals and allows responses to be aggregated, compared across sub-groups, and analysed using inferential statistics.

3.4 Research Design

The research design used in the study is descriptive, cross-sectional research design, which has a correlational aspect of it. The descriptive component provides an overview of the existing state of distribution of attitudes, use patterns, perception risk among the respondents, using frequencies, percentages, and mean scores of agreement. The element of cross-section is that the data were measured at one moment not over a period of time, which is appropriate as the perceived labour-market effect of LLMs is a new, yet not yet well-established phenomenon to which a longitudinal design would be too early in the context of the current dissertation [37]. Correlational dimension is the ability to test relationships between important variables, including years of experience, frequency of use of LLM and perceived risk of unemployment, which is statistically tested, not merely described as an increase the chapter can both describe what the overall perceptions of the sample are, but test the theoretically based expectation, as was derived in Chapter 2 that risk perception is not seen identically at all experience levels.

3.5 Population and Sampling Technique

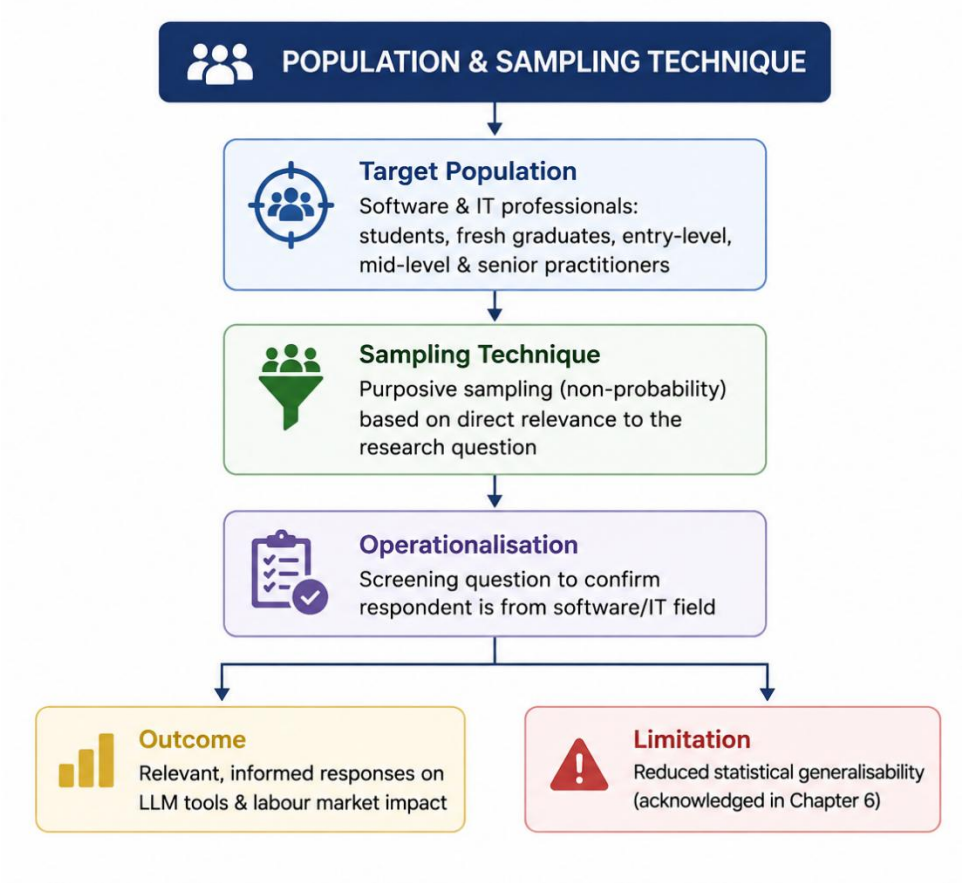


Fig 3.1: Population and Sampling Technique

The intended audience is software and IT professionals spanning the entire range of experience, including students and fresh graduates, entry-level, mid-level and senior practitioners, as is outlined in Section 1.7.1. This population is selected since it is the one that is directly affected by or will soon affect the labour market circumstances due to adoption of LLM.

The respondents were recruited through the use of a purposive sampling technique, which is a type of non-probability whereby participants are sampled based on their selection due to their direct relevance to the research question [35]. It was selected instead of the possible alternatives based on probability sampling because there is no full sampling frame of software engineers to draw a random sample, and because the study also demands that the respondents are in direct or informed contact with software development work and LLM tools and not a cross-section of the general population [36]. This was operationalised by having a screening question to verify that the respondent is related to the software or IT sphere before continuing. A trade-off diminished statistical generalisability relative to probability sampling is recognised as a weakness that is revisited in Chapter 6.

3.6 Sample Size

A total of 63 valid responses were received in the data collection period which were consented. Although small relative to overall standards, this is in the mode of operation traditionally considered as acceptable in exploratory, questionnaire based student research working within viable time and resource limits, where recommended sample sizes of about 30-100 respondents are often cited as adequate to descriptive analysis as well as simple inferential statistics like correlation and analysis of variance [33]. The sample obtained also represents an acceptable distribution balance in terms of the experience level, including having no professional experience and three-to-five years of experience, facilitating the purpose of the study, which is to comparatively analyze the perceptions of the career stages. The size of the sample is treated however as an objection instead of a foundation of generalising on the population otherwise discussed in the final chapter.

3.7 Data Collection Method

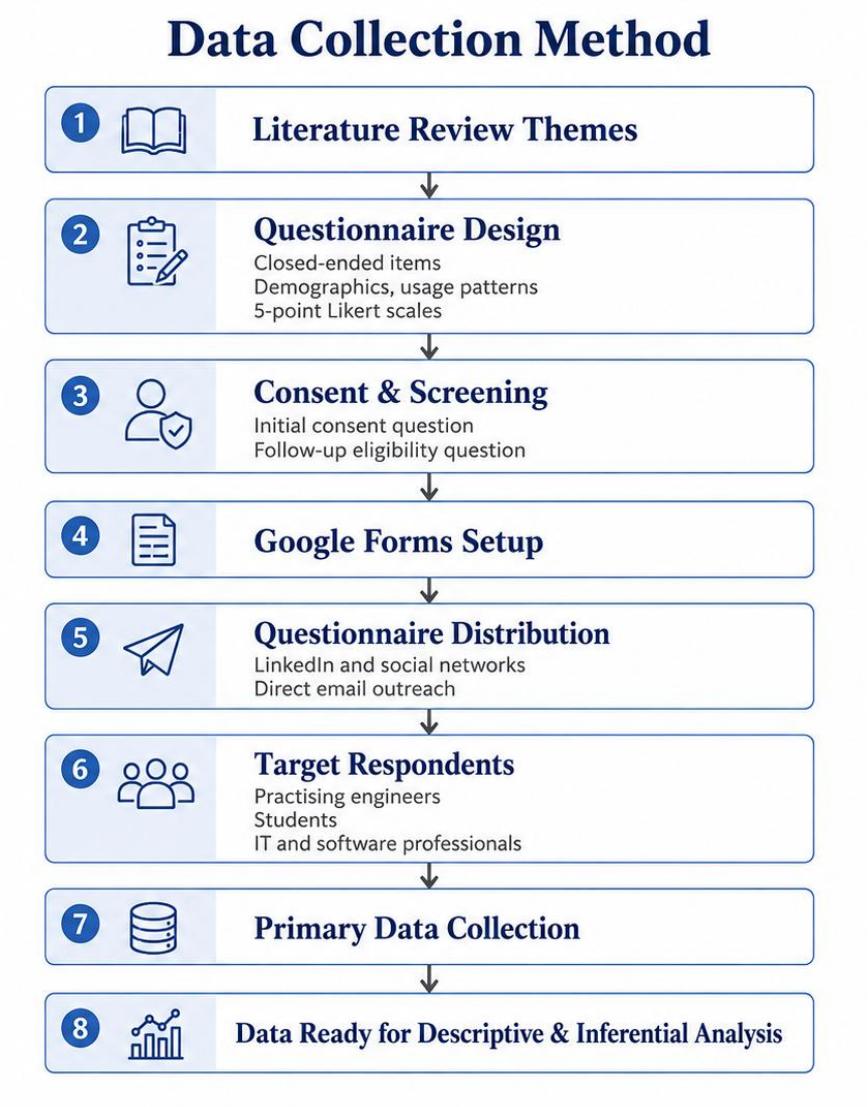


Fig 3.2: Data Collection Method

A self-administered online questionnaire was created and distributed using Google Forms to collect primary data in the form of a structured self-administered questionnaire. The instrument had strictly closed-ended items; single-select and multiple-select items of categorical data (demographics, usage-patterns), and five-point Likert-scale items (Strongly Disagree to Strongly Agree) of attitudinal and perception-based constructs. Likert format was considered as an effective way of developing the extent of attitudes held by the respondents about an abstract notion like perceived job risk or productivity gain, which generates ordinal data that are amenable to descriptive and inferential analysis [38].

The content of the questionnaire reflected the thematic analysis of the literature review, with blocks of items explaining the familiarity of the program and usage pattern of the program,

positive or negative impact of the program on productivity, negative or positive effect of the program on entry-level employment, new job creation and general demographic and professional background, which reinforced the content validity of the instrument, as discussed in the following section 3.9. The purposive sampling criterion above was operationalised by providing a follow-up screening question, and the first question had to explicitly consent to proceed with further questions. Outreach to the questionnaire link was done via social networks, including LinkedIn as a source of contact in the professional world, as well as direct emailing of contacts in the software and IT profession to ensure that practising engineers, students and industry professionals were reached within the resource limitation of the study.

3.8 Data Analysis Technique

The analysis of the data was done in two steps, the first step was carried out with the help of IBM SPSS Statistics which was chosen because of its high reliability, great acceptance in the context of social science researches on Likert-scale survey data and the full range of procedures of data analysis in terms of descriptive and inferential tests [39].

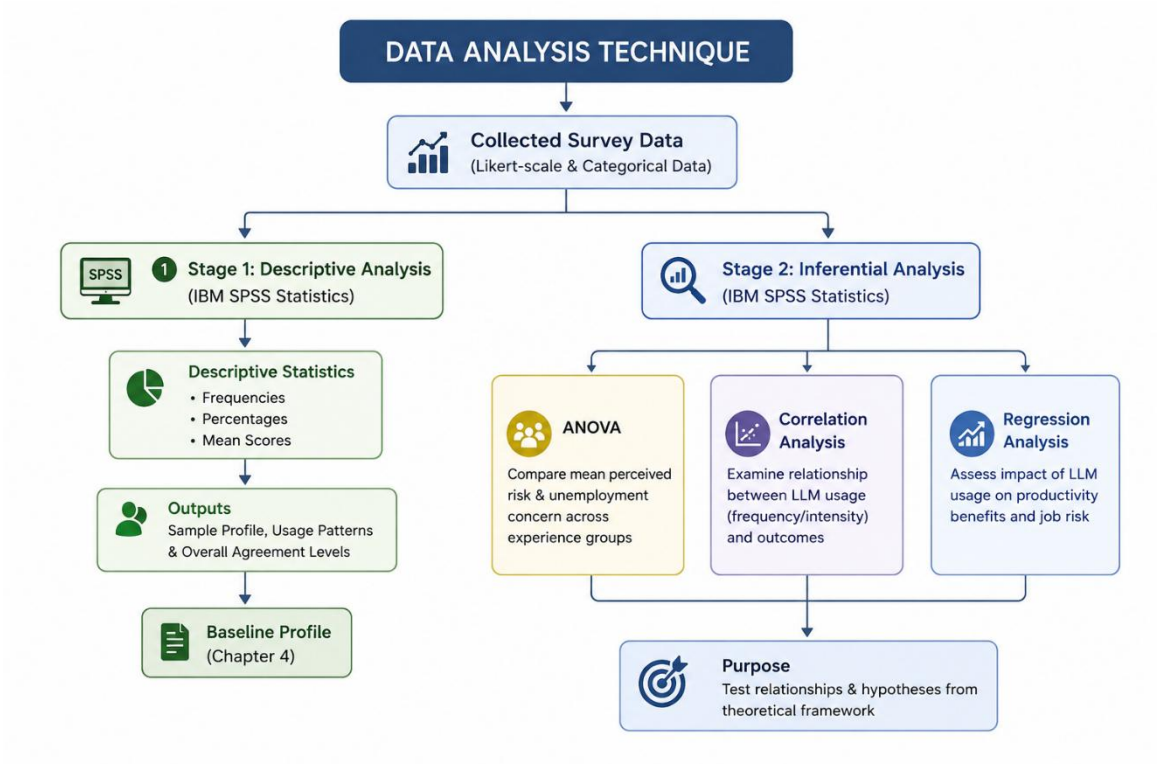


Fig 3.3: Data Analysis Technique

In the former stage, the demographic make-up of the sample, patterns of use of LLD and the general level of consent with every attitudinal declaration was described using descriptive

statistics such as frequencies, percentages and means scores, thus in Chapter 4 as the baseline profile where propositions of the theoretical hypotheses in Chapter 2 were tested.

Inferential methods were used in the second stage to test relationships that were the focus of research questions. ANOVA compared the average levels of perceived risk of and concern and the perceived risk of unemployment between groups that were based on experience, which tested the put forward theoretically based expectation that there would be differences in risk perception between junior and senior professionals. Correlation and regression analysis were used to assess the relationship between the frequency and intensity of use of the LLM tools and the perceived productivity benefits and job risk with the hypothesis being whether an increase in exposure to the LLM tools leads to an increased or decreased employment concern. The combined set of operations helps the study go beyond a broad portrait of what people believe to be the case to a statistically based test of the correlation expected based on the technological unemployment and skill-based technological change literature.

3.9 Reliability and Validity

A design solution to content validity was to create each item as a direct result of established constructs within the literature review such as task automation, productivity impacts, perceived threat to entry-level positions, and new employment openings, therefore the contents of the instrument align rather well with the theoretical area that it is designed to represent [40].

Cronbach alpha, the most popular reliability coefficient to use on attitudinal scales of this nature, was used to determine the internal consistency of the constructs of the multi-item Likert-scale, and a threshold of 0.70 (the traditional coefficient representing reasonable levels of internal consistency) was adopted to indicate the presence of acceptable internal consistency [41]. The reliability statistics are then reported in Chapter 4 with the corresponding analyses.

3.10 Ethical Considerations

The research was conducted following the best ethical standards in research studies involving humans. Respondents were invited to participate and the type of informed consent was a compulsory consent question which was presented at the beginning of the questionnaire; those who did not consent are not shown the other questions. No identifiable data like names, email addresses, and employer identities were gathered personally, and the responses were kept confidential and employed in the framework of the academic needs of this dissertation. If the

respondent had to leave the questionnaire, he/she could do so any time before handing it in and the information would be preserved when it is secure and only the researcher can access it.

CHAPTER 4

RESULTS AND ANALYSIS

4.1 Introduction

This chapter outlines results from the 63 valid survey responses described in Chapter 3, both in a descriptive and inferential manner. It moves from the sample profile and LLM use patterns to the reliability of the two composite measurement scales, then onto the inferential tests, analysis of variance (ANOVA), Spearman correlation and multiple linear regression that are used to analyse the relationships between experience, LLM use and expected employment outcomes as outlined by the technological unemployment and the skill-biased technological change literature in Chapter 2. The results are not only reported numerically, but also accompanied by a critical interpretation of the significance of those results for the research questions posed in Chapter 1.

4.2 Response Rate and Sample Profile

Sixty-three (63) respondents agreed to participate and fill out the questionnaire. The sample's composition is summarised in Table 4.1. Most (82.6%) said they were either in the software/IT industry or were students/fresh graduates in a related discipline, reflecting the success of the purposive sampling strategy described in Section 3.5 of reaching the desired population; only the remaining 11.1% reported no connection to the field. As would be expected in a study with a particular emphasis on the attitudes of LLM adopters as they enter the profession, there is a high proportion of the respondents who are in the early stages of their careers (84.1% between the ages of 20 and 29) and only 30.2% have no professional experience, although long-tenured senior engineers are necessarily less well represented.

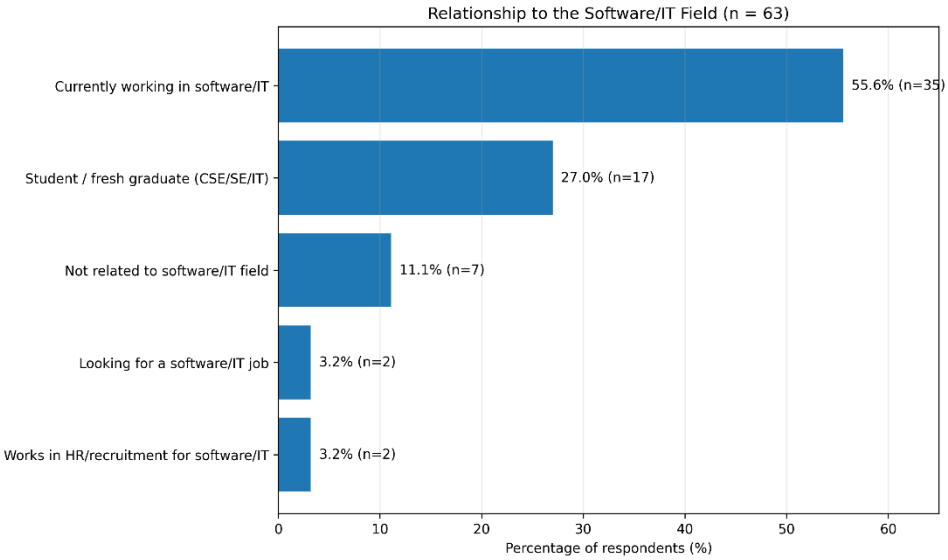


Fig 4.1: Relationship of the software

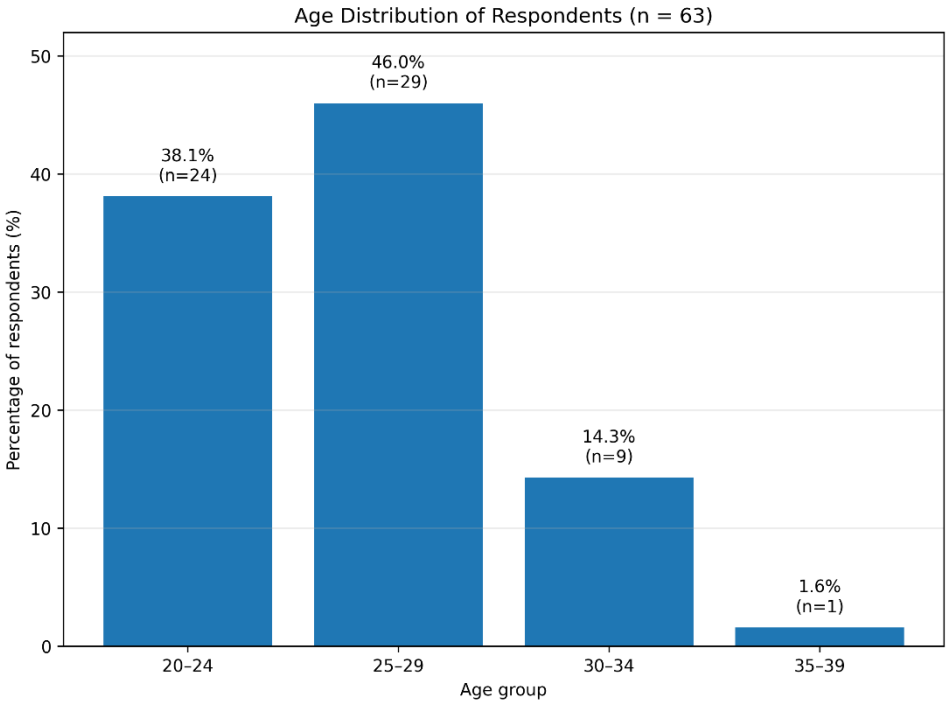


Fig 4.2: Age Distribution

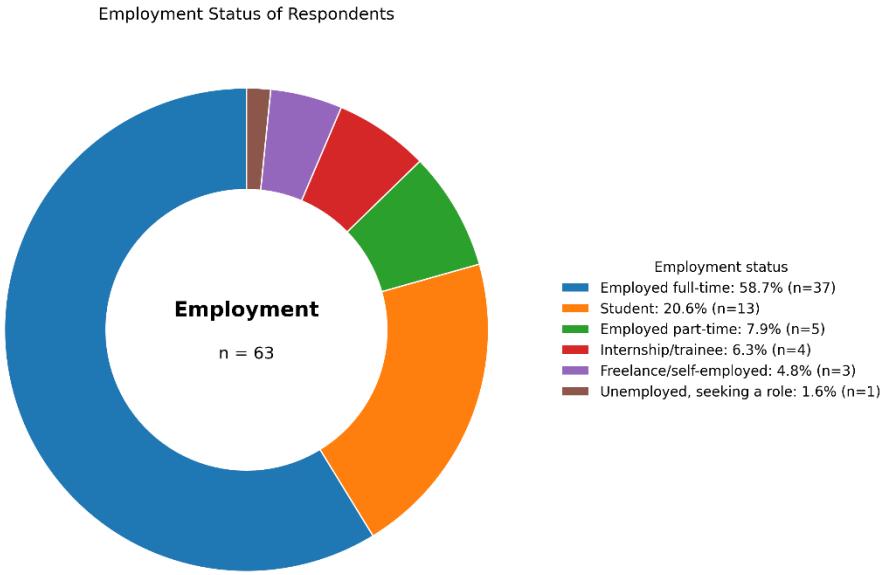


Fig 4.3: Employment Status of Respondents

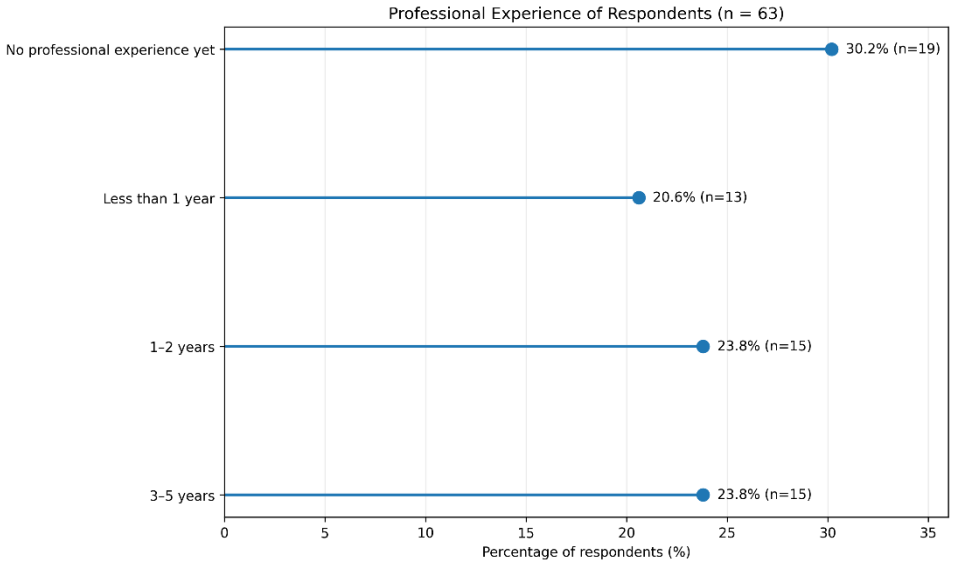


Fig 4.4: Professional Experience of Respondents

Variable	Category	n	%
Relationship to software/IT field	Currently working in software/IT	35	55.6
	Student / fresh graduate (CSE/SE/IT)	17	27.0
	Not related to software/IT field	7	11.1

	Looking for a software/IT job	2	3.2
	Works in HR/recruitment for software/IT	2	3.2
Age group	20-24	24	38.1
	25-29	29	46.0
	30-34	9	14.3
	35-39	1	1.6
Employment status	Employed full-time	37	58.7
	Student	13	20.6
	Employed part-time	5	7.9
	Internship/trainee	4	6.3
	Freelance/self-employed	3	4.8
	Unemployed, seeking a role	1	1.6
Years of experience	No professional experience yet	19	30.2
	1-2 years	15	23.8
	3-5 years	15	23.8
	Less than 1 year	13	20.6

Table 4.1: Sample Profile of Respondents (N = 63)

4.3 LLM Awareness and Usage Patterns

As per Table 4.2, the familiarity with using LLM tools is high with 71.4% are either familiar or strongly familiar with tools like ChatGPT, GitHub Copilot, and Claude, while 3.2% are not. The number of users is equally heavy: 46.0% indicate that they use LLM tools "Always" when working on software, while an additional 12.7% use them "Often" and just 7.9% say they never

use them. Most respondents (85.7%) have reported that they use LLM assistance between 21-60%. Altogether, these trends suggests that this sample isn't just familiar with LLMs in principle, but that they have already built them into their work with software—making their judgments about the use impact more credible, as they're based on hands-on experience.

Variable	Category	n	%
Familiarity with LLM tools	Strongly Agree / Agree	45	71.4
	Neutral	14	22.2
	Disagree	2	3.2
Frequency of LLM use	Always	29	46.0
	Sometimes	12	19.0
	Often	8	12.7
	Rarely	8	12.7
	Never	5	7.9
% of work assisted by LLMs	41-60%	30	47.6
	21-40%	24	38.1
	1-20%	4	6.3
	61-80%	3	4.8
	0% / More than 80%	2	3.2

Table 4.2: LLM Familiarity, Usage Frequency, and Work Share

4.4 Reliability of the Measurement Scales

Cronbach's alpha was used to check the internal consistency of the two composite scales employed in this chapter: Productivity Benefit Index and Perceived Risk Index. As seen from Table 4.3, the Productivity Benefit Index (coding speed, debugging time, junior learning, and experienced-engineer productivity) had an alpha coefficient of .775, which is a good internal consistency as per conventional standards. The Perceived Risk Index (LLMs performing junior tasks, reduction in entry level jobs, fewer junior hires and personal job concern) had alpha = .701, which is greater than the .70 level for acceptable alpha (scale reliability), but the item-total statistics indicate that the perceived concern about their own jobs was the least correlated item on the scale (correlation = .335), suggesting that the items are related but not fully interchangeable. Both scales are considered to be sufficient for use in the composite-index analyses that are presented later; the modest level of reliability for the risk scale is discussed as a limitation in Chapter 6.

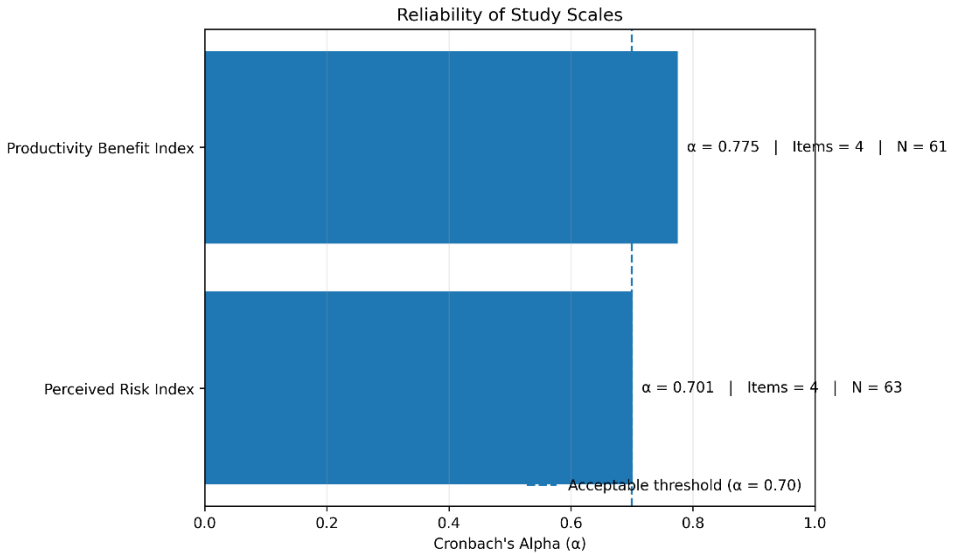


Fig 4.5: Reliability Scales

Scale	Items	N	Cronbach's Alpha
Productivity Benefit Index	4	61	0.775
Perceived Risk Index	4	63	0.701

Table 4.3: Reliability Scales

4.5 Descriptive Analysis of Perceived Benefit and Risk

The mean and standard deviation for each of the 11 individual Likert items is reported in Table 4.4. Mean scores across the eleven statements are above the scale midpoint of 3.0, with a narrow spread from 3.71–4.03, suggesting that on average participants agree with both the productivity-enhancing and risk-generating characterisations of LLMs. The picture presented in the Chapter 2 literature review—also called the ambivalent picture—is not a contradiction; rather, it is two sides of the same technological change, as seen by respondents, because respondents do not perceive productivity gain and employment risk as mutually exclusive. Looking at the items with the highest scores, there is a single item rated high with a mean of 4.03: LLMs enable completing coding tasks faster and a single item rated high with a mean of 4.03: LLMs are used to learn programming by junior developers, indicating that LLMs are perceived as a true aid for skill acquisition rather than as a pure threat. Meanwhile, the item that LLMs were doing things for juniors also achieved M=3.98 at the same time, and this perceived 'help' was not perceived as unrelated to a simultaneous 'push out' of the routine work done by juniors. The least-favored item (M=3.71) is that senior engineers are less likely to be replaced than juniors, and is the most contested in the set of statements, indicating that perhaps the literature tried to present the idea that seniority is protecting you is taken with less conviction than the other items.

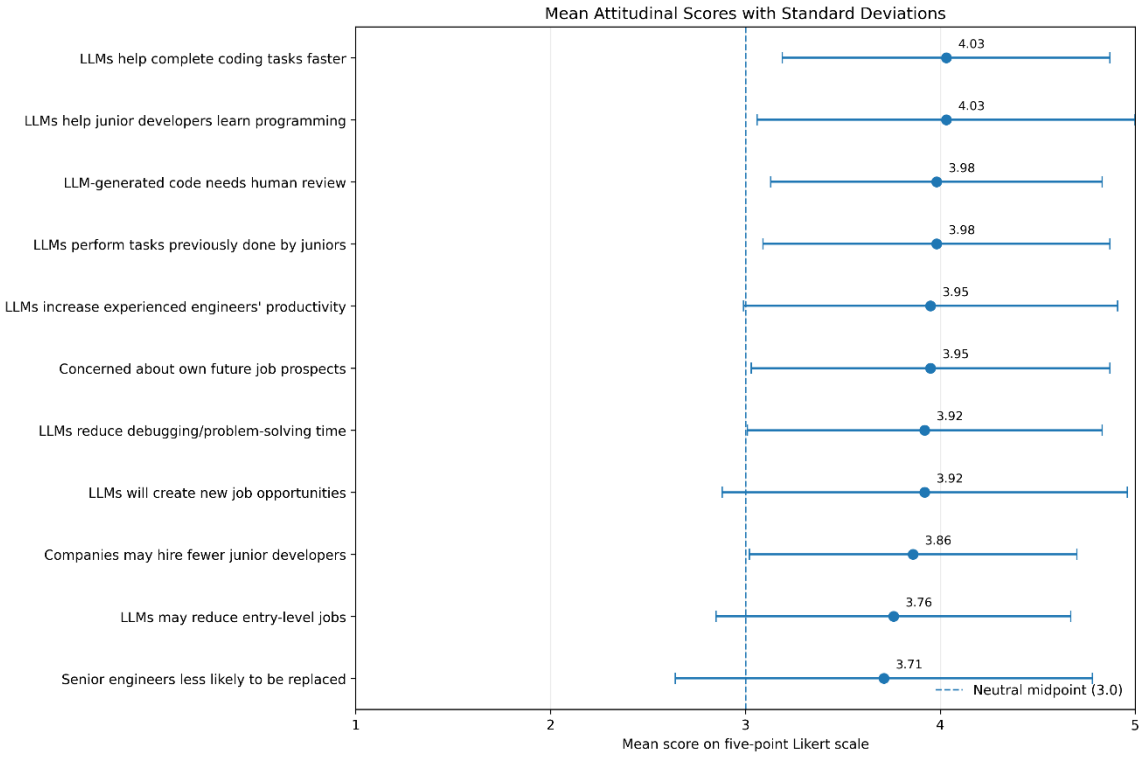


Fig 4.6: Analysis of Perceived Benefit and Risk

Statement	Mean	SD
LLMs help complete coding tasks faster	4.03	0.84
LLMs help junior developers learn programming	4.03	0.97
LLM-generated code needs human review	3.98	0.85
LLMs perform tasks previously done by juniors	3.98	0.89
LLMs increase experienced engineers' productivity	3.95	0.96
Concerned about own future job prospects	3.95	0.92
LLMs reduce debugging/problem-solving time	3.92	0.91
LLMs will create new job opportunities	3.92	1.04
Companies may hire fewer junior developers	3.86	0.84
LLMs may reduce entry-level jobs	3.76	0.91
Senior engineers less likely to be replaced	3.71	1.07

Table 4.4: Descriptive Statistics for the Eleven Attitude Statements

4.6 Inferential Analysis

4.6.1 Effect of Experience on Perceived Risk

Directly addressing the research question as to whether perceived risk is differentiated by career stage, a one-way ANOVA was run to test the differences in job risk among the four experience categories. Levene's test showed that variance was homogenous (p=.824, not shown), thus allowing for the standard F-test to be used. The results indicated that differences in the career stage (experience) accounted for a significant portion of the variance in the Perceived Risk Index (PR) as the difference was statistically significant ($F(3,58) = 4.719$, $p = .005$).

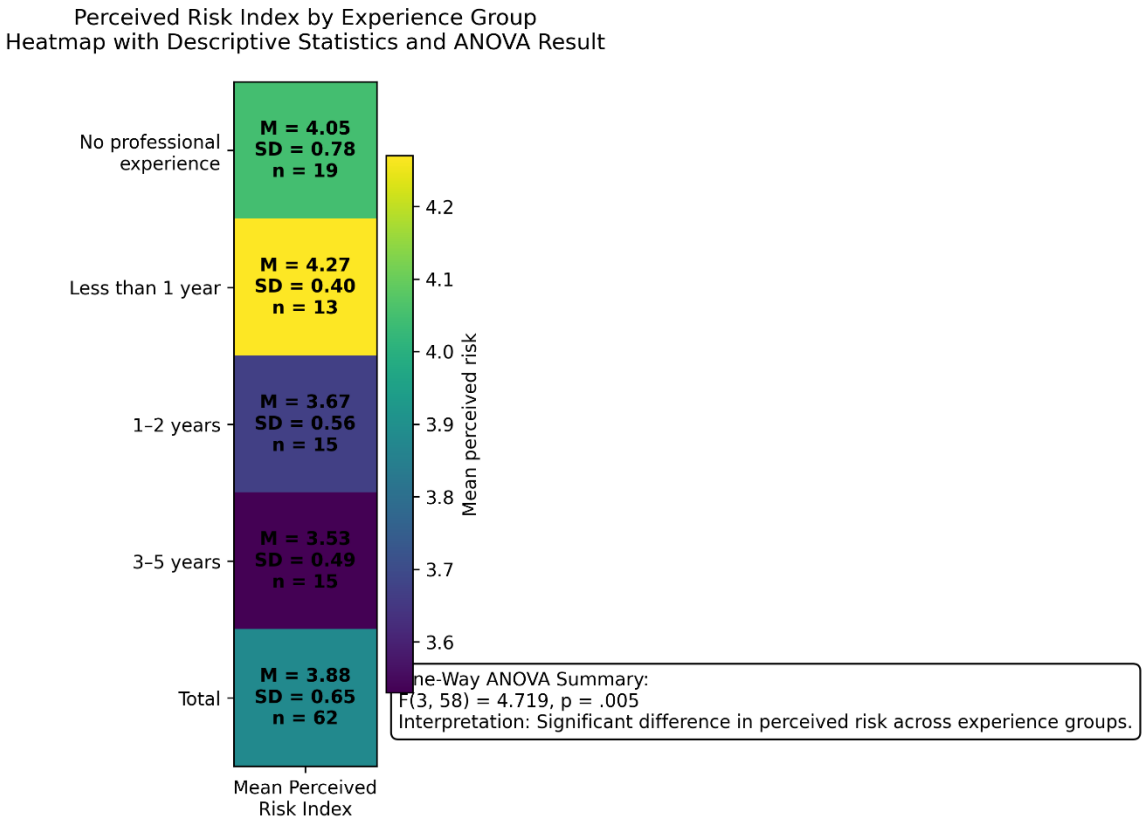


Fig 4.7: Experience on Perceived Risk

Experience group	n	Mean	SD
No professional experience	19	4.05	0.78
Less than 1 year	13	4.27	0.40
1-2 years	15	3.67	0.56
3-5 years	15	3.53	0.49
Total	62	3.88	0.65

Table 4.5: Descriptive Statistics: Perceived Risk Index by Experience Group

Source	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	5.021	3	1.674	4.719	.005

Within Groups	20.572	58	0.355		
Total	25.593	61			

Table 4.6: One-Way ANOVA: Perceived Risk Index by Experience

The Tukey post hoc comparison in Table 4.7 elaborates upon this more subtly than a junior versus a senior dichotomy. The two pairs of groups that have significant differences are both based on the "Less than 1 year" group; the Less than 1 year group is significantly different from the 1-2 year group (p=.047) and the 3-5 year group (p=.010). What is important is that the group with the highest mean risk perception (M=4.27) are not the group with no paid employment whatsoever (students, M=4.05), but the group just entered paid work. This is a very real detail in the technological unemployment argument presented in Chapter 2 as the anxiety about LLM-driven displacement is not at its highest when beginning to venture into the labour market, but rather as soon as a respondent is directly introduced to how LLMs are used in a real world job. The perception of risk then diminishes over time as experience is gained, in line with the hypothesis of so called skill-biased technological change in the literature, which suggests that the more experience engineers gain in their jobs, the less they perceive the risk of their jobs.

Comparison	Mean Difference	Sig.
Less than 1 year vs 1-2 years	0.603	.047*
Less than 1 year vs 3-5 years	0.736	.010*
No experience vs 3-5 years	0.519	.067
No experience vs Less than 1 year	-0.217	.744
No experience vs 1-2 years	0.386	.249
1-2 years vs 3-5 years	0.133	.928

Table 4.7: Tukey HSD Post Hoc Comparisons: Perceived Risk Index

4.6.2 Effect of Experience on Perceived Productivity Benefit

A parallel ANOVA was run on the Productivity Benefit Index. Experience proved to have a statistically significant impact, again, but less so than for perceived risk (Table 4.9) $F(3,58)=3.534$ ($p=0.020$). Those with no professional experience ($M=4.32$) have the highest perceived benefit, and it decreases towards the 1-2 year group where the only statistically significant pairwise difference is between no pro experience and 1-2 years ($p=.039$; pairwise differences Tukey table shown for brevity). This indicates that the most positive perceptions regarding the use of LLMs as a productivity and learning aid lie among students and newcomers who haven't yet faced the drawbacks of the tools in a production environment, which is perhaps why engineers with 1–2 years of hands-on experience exhibit lower scores, and perhaps explain the scores for those with 2–3 and 3–5 years of experience ($M=3.01$ and $M=3.36$, respectively).

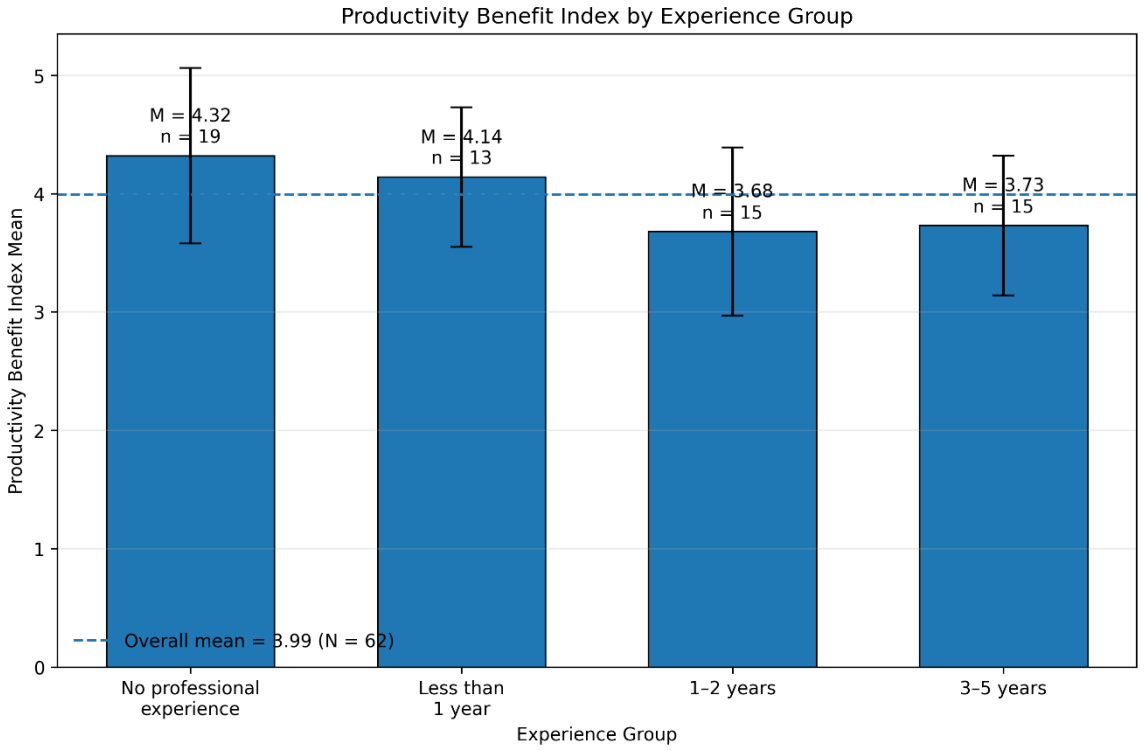


Fig 4.8: Productivity Benefit

Experience group	n	Mean	SD
No professional experience	19	4.32	0.74

Less than 1 year	13	4.14	0.59
1-2 years	15	3.68	0.71
3-5 years	15	3.73	0.59
Total	62	3.99	0.71

Table 4.8: Descriptive Statistics: Productivity Benefit Index by Experience Group

Source	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	4.766	3	1.589	3.534	.020
Within Groups	26.070	58	0.449		
Total	30.836	61			

Table 4.9: One-Way ANOVA: Productivity Benefit Index by Experience

4.6.3 Correlation Analysis

Spearman correlations were calculated between the two composite indices and three exposure variables – experience, frequency of LLM use, and percentage of work assisted by LLM – along with familiarity for the benefit index. As is seen in Table 4.10, the variables Perceived Risk and Productivity Benefit were moderately but significantly positively correlated with usage frequency (rho=.510, p<.001 and rho=.520, p<.001 respectively), whereas, there was moderate, significant negative correlation between the variables experience and Perceived Risk (rho=-.432, p<.01) and experience and Productivity Benefit (rho=-.418, p<.01). Productivity Benefit (rho=.578, p<.001) is the strongest single correlation displayed in the table, and comes from familiarity with LLM tools. Interestingly, the proportion of work that participants actually use LLMs for is not significantly correlated with either of the indexes (rho=.119 and rho=.061, neither significant), which contradicts the logical expectation that more use of LLMs in their job would correspond directly to higher risk or benefit perceptions. This disconnect between the frequency of use and volume of work implying that it is perhaps not the extent of the tasks delegated to LLM tools that influences engineers' judgement of their impacts on their jobs, but rather the habit and familiarity of dealing with them.

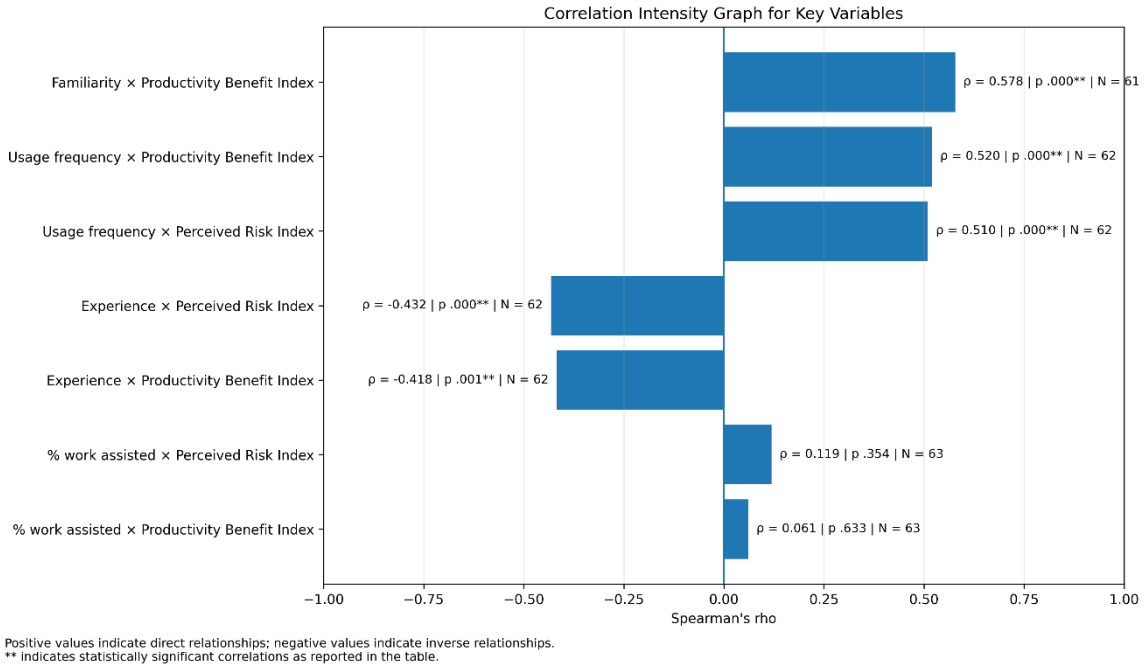


Fig 4.9: Correlation Analysis

Variable pair	Spearman's rho	Sig. (2-tailed)	N
Usage frequency × Perceived Risk Index	0.510	.000**	62
Experience × Perceived Risk Index	-0.432	.000**	62
% work assisted × Perceived Risk Index	0.119	.354	63
Usage frequency × Productivity Benefit Index	0.520	.000**	62
Familiarity × Productivity Benefit Index	0.578	.000**	61
Experience × Productivity Benefit Index	-0.418	.001**	62
% work assisted × Productivity Benefit Index	0.061	.633	63

Table 4.10: Spearman Correlation Coefficients Between Key Variables

4.6.4 Multiple Regression Predicting Perceived Risk

To identify which of the exposure factors explain perceived risk when the other are taken into account, a multiple linear regression was run: Perceived Risk Index as the dependent variable, with experience, familiarity, usage frequency and percentage of work assisted as the independent variables entered simultaneously. The overall model accounted for a significant 35.2% of the variance in perceived risk (Adjusted R^2=.305; Table 4.11), which is a reasonably high figure for an individual, one-shot cross-sectional study such as this.

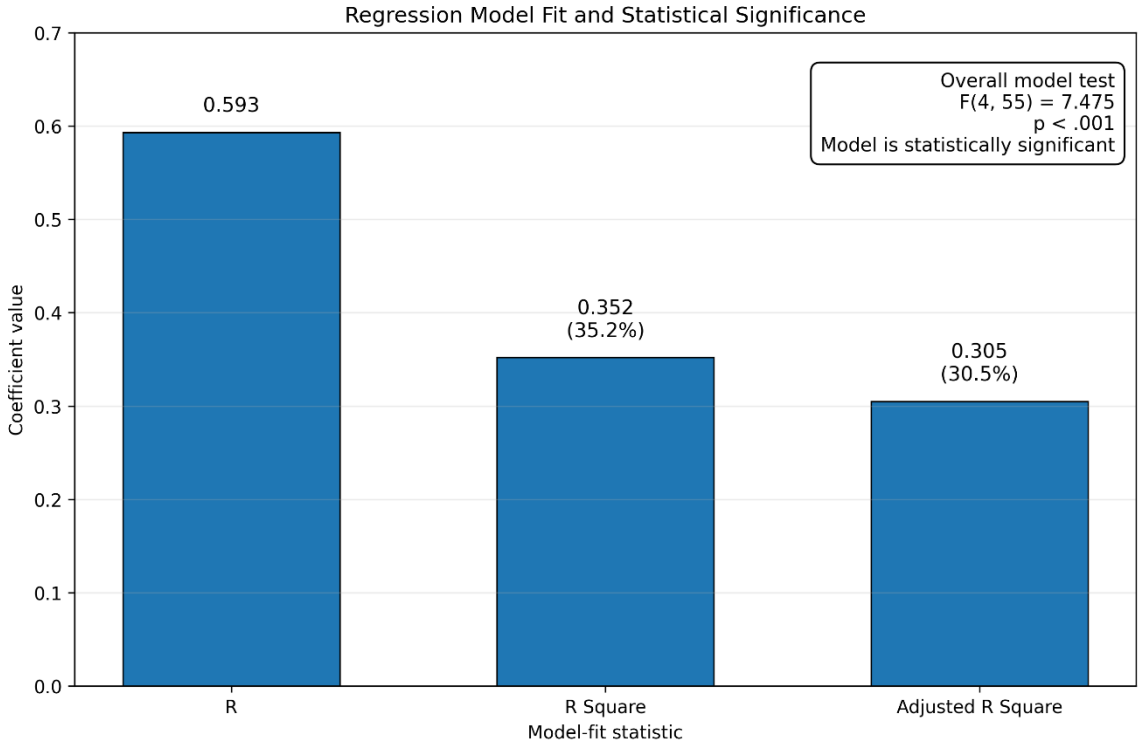


Fig 4.10: Predicting Perceived Risk

Model Summary	Value
R	0.593
R Square	0.352
Adjusted R Square	0.305
F (4, 55)	7.475
Sig.	.000

Table 4.11: Regression Model Summary

The coefficients of the Table 4.12 provides a result that is a very significant qualification on the correlation findings in Section 4.6.3. When all four predictors are computed simultaneously, familiarity with LLM tools is the only statistically significant independent predictor of perceived risk (B=.323, beta=.432, p=.001); experience, usage frequency and percentage of work assisted are all no longer statistically significant in the presence of familiarity (p= .669, .173, and .617, respectively). This suggests that the bivariate relationships observed for experience and usage frequency in Section 4.6.3 are largely accounted for by the relationship of these variables with familiarity, and not by any direct influence of these. From a practical perspective, the best, strongest predictor of the hunter's sense of threat of LLMs is not how many years they have spent doing engineering or how often they actually use LLMs, but how well they understand the tools – what they can and cannot do. This is an important detail to be noted when discussing Chapter 5 interventions that focus on perceived risk from the job are likely to be more effective at cultivating structured understanding of LLM capabilities than at trying to raise or lower exposure to the tools themselves.

Predictor	B	Std. Error	Beta	t	Sig.
(Constant)	2.153	0.463		4.650	.000
Experience	-0.032	0.074	-0.058	-0.430	.669
Familiarity with LLM tools	0.323	0.095	0.432	3.412	.001**
Frequency of LLM use	0.093	0.067	0.200	1.382	.173
% work assisted by LLMs	0.044	0.087	0.057	0.503	.617

Table 4.12: Regression Coefficients: Predictors of Perceived Risk Index

4.7 Summary of Key Findings

The findings presented in this chapter give a direct empirical answer to a number of the research questions presented in Chapter 1. First, the surveyed software professionals use LLMs extensively and the use of these tools is already well entrenched in the daily workflow, which means that the labour-market impacts from these that are explored in this work are not

hypothetical, but they happen alongside the current widespread use of LLMs. Second, perceived risk is not monotonic in job tenure: both the very young (just entering the labour market) and the very old (very experienced) face considerably higher levels of risk than others, directly supporting the differentiated-risk hypothesis (as implied by the literature on skill-biased technological change in Chapter 2) while providing a more specific finding that the increased risk is found only after rather than before entering the labour market. Finally, the perceived risk and perceived benefits increase with the frequency of use, which is consistent with the theoretically ambivalent nature and experience of the LLM adoption described in the literature review. Fourth, and most importantly, the regression analysis reveals that familiarity, not a direct measure of experience or usage intensity, is the most powerful independent predictor of perceived risk; this is an important contribution of this dissertation which has not received a large amount of prominence within the literature to date.

CHAPTER 5

DISCUSSION

5.1 Introduction

This chapter draws conclusions from the results detailed in Chapter 4 in the context of the six research objectives outlined in Section 1.4 and the theoretical background, technological unemployment theory and skill-biased technological change outlined in Chapter 2. Instead of summarising the statistical results, the conversation tackles the meaning of these results for the original question "Are Large Language Models changing employment in software engineering, and if so, how, and for whom?". The various objectives are taken up in sequence and then a synthesis is done on the basis of what the evidence implies for the theoretical debate between displacement and complementation. Where the empirical results support the expectations raised in the literature review, it is stated explicitly; where the results support or do not support the expectations raised by the literature, this is deemed analytically significant, as that is where the original contribution of this thesis lies.

5.2 Discussion by Research Objective

5.2.1 Understanding How LLMs Are Used in Contemporary Software Development

The first goal was to gain insights on the use of LLMs in practice. As seen in Section 4.3, this is no longer a niche or fringe activity: 71.4% of the respondents said they know how to use LLM tools, and 46.0% report using them Always in their software work, with the majority saying between 21% and 60% of their work is LLM-assisted. This validates empirical observations made in Chapter 2 that LLMs have become a common staple in the software development toolbox, serving as a primary tool for code generation, code explanation and code debugging. This finding provides the basis with which each subsequent objective can be founded since it is no longer hypothetical but actual in a high-intensity adoption situation; it also supports the choice of studying perceptions of practising or soon-to-be-practising engineers, instead of perceptions of the general public.

5.2.2 The Degree of Automation of Software Engineering by LLMs

The second objective investigated the extent of the automation of software engineering tasks by LLMs. Descriptive results in Section 4.5 reveal that the results were consistently high across all eleven items (M=3.71-4.03) indicating that LLMs can meaningfully improve coding and debugging speed and do the work of junior engineers. This pattern provides for the task substitution feature as described in chapter 2 which allows automation to take over certain tasks, boilerplate coding, standard debugging, writing documentation, etc. without actually abolishing the job. Of particular relevance, the statement which drew the highest agreement in conjunction with coding speed was the LLM-generated code requires human review (M=3.98), suggesting that while AI can offer assistance, it does not yet fully replace human technical decision-making, which aligns with the notion that AI acts as a complement to rather than a complete replacement for human insight [21].

5.2.3 The Impact of LLM Adoption on Software Engineers' Employment Opportunities

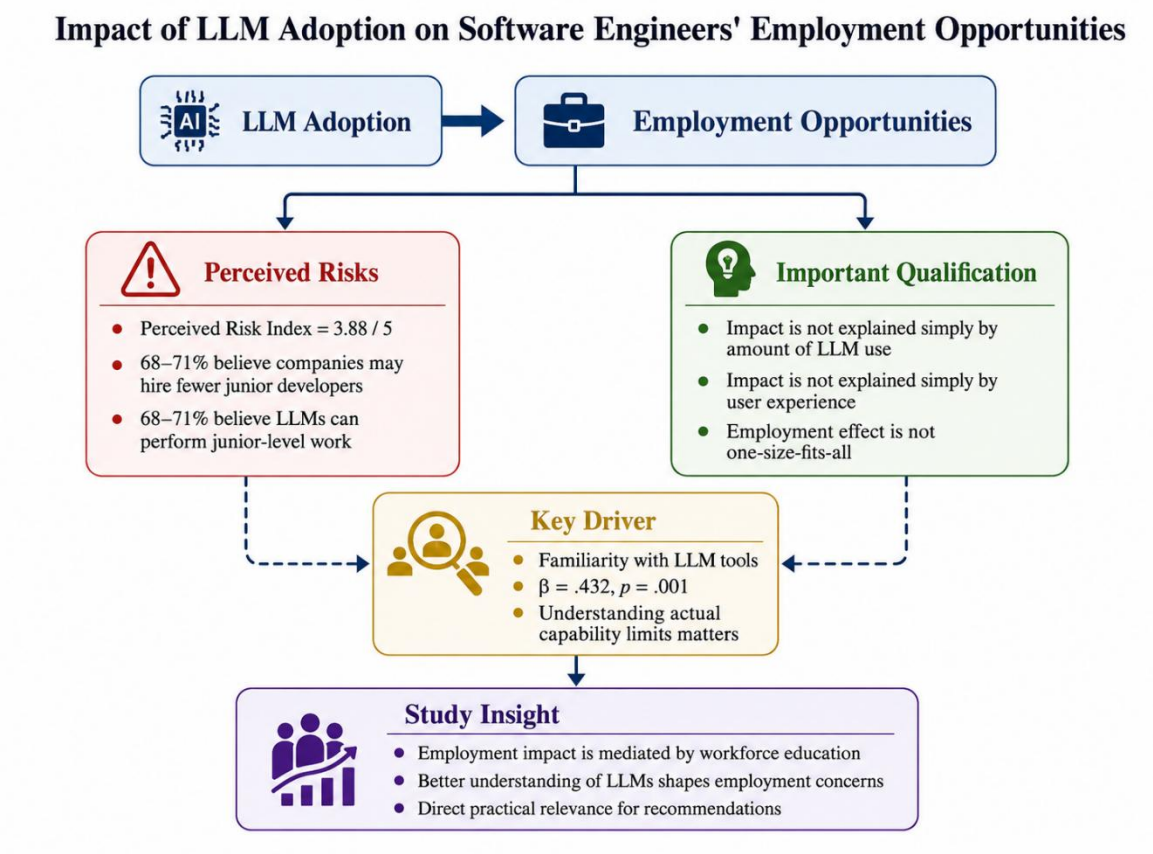


Fig 5.1: Impact of LLM Adoption on Software Engineers' Employment Opportunities

Explanations on the impact of LLM adoption on employment opportunities, the third and central objective, are included in the results. The evidence of this is indeed two-sided not one-sided. Conversely, the Perceived Risk Index was rated as 3.88 out of 5, while 68-71% indicated that companies could hire less junior developers and LLMs could do junior-level work. The concern was, however, not just a function of amount of LLM use or respondent experience with LLMs; the beta of .432 (p=.001) for familiarity with the tools was the only significant exposure variable within the regression analysis in Section 4.6.4. This is a significant qualification of the displacement narrative: the effect on employment that this thesis intended to measure is not just the effect of exposure to the technology, but impact through engineers understanding of the actual capability limits of the technology. This nuance was not noticeable in the literature in Chapter 2 and it is one specific empirical contribution of this study. It also indicates that employment impact is not necessarily a one-size-fits-all outcome of adopting the technology, but rather an impact which is mediated by the education of the workforce about the technology with which it is asked to cooperate, with direct practical relevance for the recommendations to be developed in Chapter 6.

5.2.4 Which Software Engineers Are Most Affected by LLM Incorporation

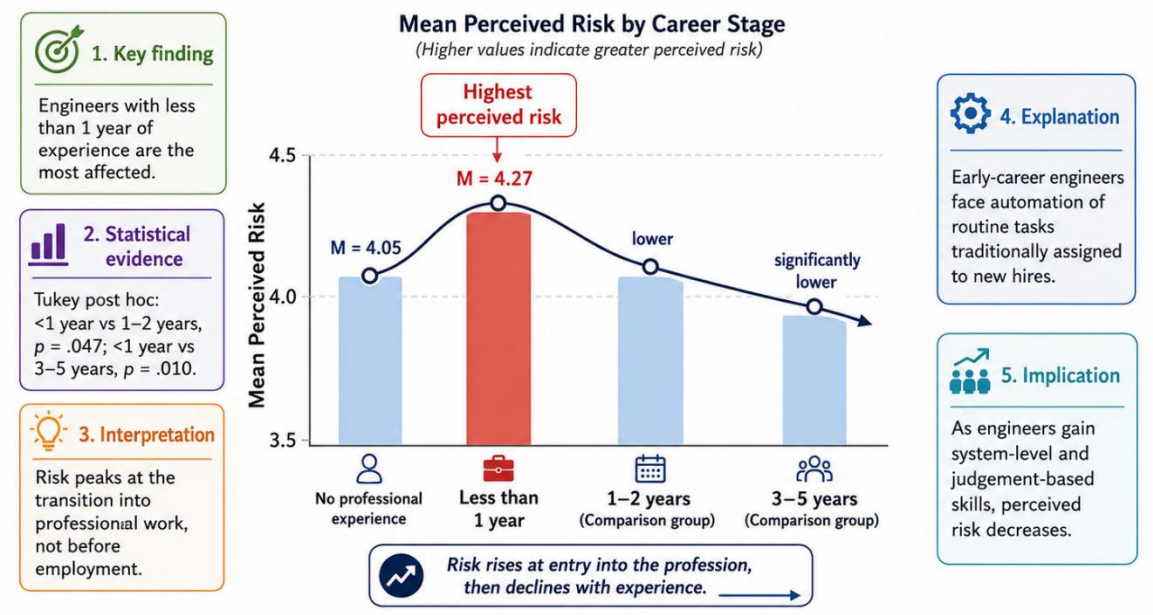


Fig 5.2: Software Engineers Are Most Affected by LLM Incorporation

The fourth goal was to find out which kind of software engineer is most impacted. The results from the ANOVA and Tukey in section 4.6.1 provides a more precise answer to this than the in general junior versus senior framing of the literature review. The mean perceived risk is significantly higher for the less than one year group (M=4.27) compared to the 1-2 year group

(p=.047) and the 3-5 year group (p=.010); although marginally higher than that of the group who had no professional experience (M=4.05). Rather than the pre-employment cohort that the automation literature expects to be most affected, this indicates that engineers who have recently begun their careers and are facing the automation of the kind of rote work traditionally assigned to new hires are the most impacted, at least in the sense of perceiving the risk. This is a refinement of the skill-biased technological change (SBTC) hypothesis: that risk will not be simply linearly proportional to seniority, but rather will reach a maximum at the transition point into the profession and diminish as engineers gain the system-level, judgement-based skills mentioned in Chapter 2, which are less automatable [17].

5.2.5 Whether the Technology Industry Has Created New Employment

The fifth goal sought to investigate if LLM has created new jobs. The answers here were also rather positive (M=3.92), with the same intensity as for the risk-related items suggesting that LLMs will bring new job opportunities rather than being an afterthought. This reflects the discussion in Chapter 2 on the new roles of people like AI/ML engineers, prompt engineers and AI governance specialists [9] and supports the notion that technological unemployment (where it exists) may be temporary as engineers and institutions seek to reskill through pathways outlined in the literature [30]. The survey results do not provide a measure of the scale or accessibility of these new roles as this was a perception and rather than something that was seen, this would need to be addressed through a further longitudinal research study.

5.2.6 How Software Engineers Can Adapt to AI-Driven Changes

The ultimate goal was to provide concrete adaptation advice. One empirical finding that seems to most clearly touch the objective is the regression result that familiarity, rather than tenure or frequency of use, is the predictor of lower or higher perceptions of risk. If the tech understanding is the factor most influencing what engineers believe most needs to be done to adapt, then the most direct and evidence-based suggestion for adapting to LLM that this thesis makes is that it is not more or less LLM use, but having structured, accurate understanding of what it can and cannot do; as the strongest agreement item in Section 4.5 shows it does. This aligns with the discussion of the software engineer's evolving role with AI collaboration, prompt engineering and judgement at system level, highlighted in Chapter 2, which are the skills that will differentiate the less risky, more experienced respondents in this sample.

5.3 Theoretical Implications: Technological Unemployment and Skill-Biased Technological Change

Taking as a whole, these findings point towards a qualified and not an unqualified version of technological unemployment theory. This data aligns with the notion of task substitution, as LLMs are not seen as replacing jobs but rather specific tasks, and with skill-biased technological change, as risk perception drops markedly as experience and, more specifically, familiarity with the technology grows [4]. But they do not support a simple linear view of seniority as protection: It is the newly employed, rather than the entirely inexperienced, who report the greatest concern, indicating that the theory should have a transitional-exposure term, not specifically included in existing formulations of skill-biased technological change. The data are also consistent with the complementation perspective to the displacement-only views [21] as benefit and risk perceptions are not trading off against each other and respondents perceive LLMs at the same time as augmenting and partially replacing their labour; a phenomenon which, although predicted in principle by Chapter 2 literature, has now been demonstrated statistically in a single coherent sample of practising and prospective software engineers. This dual-nature also means that policies and/or organisational interventions focused solely on either encouraging or discouraging adoption would only address part of the picture that this data tell.

CHAPTER 6

CONCLUSION AND RECOMMENDATIONS

6.1 Introduction

The final chapter concludes the dissertation by summarizing the important findings of the dissertation and suggests a general conclusion for the dissertation according to the research objectives in Chapter 1, suggestions for the groups that have been exposed to the analysis of the dissertation, limitations of the research design, and directions for further research.

6.2 Summary of the Study

This dissertation investigated the impact of Large Language Models (LLMs) on employment status of software engineers in a positivist quantitative design with a cross-sectional approach. A structured questionnaire was used in the purposively sampling of 63 software and IT practitioners and students who were analysed descriptively and inferentially with SPSS using reliability test, one way ANOVA, Spearman correlation and multiple regression. The study was driven by a controversy in the literature surrounding the concept of technological unemployment, which predicts the potential for job displacement, and the idea that LLMs complement and enhance, rather than replace, human engineers.

6.3 Conclusion

The evidence collected does not lend itself to a pure displacement-based or a pure complementary account, but to a more nuanced account in between. The adoption of LLM by the sample is high and frequent, and is seen to be useful for coding and debugging and also does tasks that were traditionally performed by junior engineers, suggesting a task-substitution rather than whole-occupation displacement scenario. Perceived employment risk is real – and on average, more than halfway into the scale – but not homogeneous with seniority; it is highest for engineers with less than one year of experience (not students), and drops significantly with experience. This builds on the skill-biased technological change hypothesis discussed in Chapter 2, and suggests that it is the move to paid work rather than the lack of paid work experience per se that is the most vulnerable time.

Perhaps most importantly, the regression analysis showed that it was not years of experience or regular use of LLM tools that was a significant independent predictor of perceived risk but

rather familiarity with their true capabilities and limitations. This suggests that understanding LLM tools is more important than the number of years or how often an engineer has used them. New employment opportunities are seen as feasible, although this study did not assess employment outcomes, but rather perception. All these results provide a response to the central research question: The use of LLMs is not generally driving job loss for software developers, and the fear posed by LLM use is primarily at the level of entry into the workforce and is primarily shaped by technological literacy, not by usage amount or length of service.

6.4 Recommendations

Increasingly, for software engineers, especially those who are new to the field, the most straightforward and useful advice is to take action today to carefully understand the capabilities and the limitations of LLMs; in this study, knowledge was the most important factor in reducing perceived risk. While the experience gained in a year is important, the skills acquired in the design, coding of systems, prompt engineering, and optimizing workflows through AI may be more valuable as a safeguard, and this is especially crucial for engineers in their first year of employment as a time of deliberate upskilling rather than passive learning.

For both employers and organisations, the results indicate that the approach to recruitment and induction of engineers should be targeted at the higher level of anxiety seen among newly hired engineers and not at all in the case of all junior staff members. It may be that structured induction to create LLM literacy, rather than just access to LLM tools, could alleviate not only concerns but actual skills deficits at the same time and organisations could be clear about their use of the tools in hiring and workload decisions, as uncertainty seems to be a major contributor to concerns.

The research highlights the need for ongoing efforts to build AI and LLM literacy in computer science and software engineering programs, as familiarity was shown to influence perceptions of employment outcomes, and not simply exposure. Guidance and reskilling programs for software experts should be rolled out early — before jobs are lost, not after — and should be complemented with increased transparency and public awareness of what jobs and tasks are truly being created by LLM adoption.

6.5 Limitations of the Study

The limitations of this study need to be noted in the interpretation of the results obtained. Although the number of respondents (63) is suitable for exploratory descriptive and inferential

analysis, it is only small and not sufficient to allow anything more than tentative generalization to the world's software engineers. The purposive, non-probability sample and the skewed age distribution (20-29) of those responding, compared to the usual age distribution of senior engineers (35-44), tends to give a strong focus to the perceptions of junior and early-career engineers rather than senior engineers. The cross-sectional design provides perceptions at one moment in the development of a technology, and cannot account for cause and effect or for long-term outcomes. Additionally, the Perceived Risk Index, although acceptable, was less reliable than the Productivity Benefit Index and indicates that the risk construct might be improved in future instrumentations. Lastly, no metrics are verified employment or hiring data; the study reflects perceived employment or hiring outcomes resulting from LLM-driven unemployment.

6.6 Suggestions for Future Research

Future studies could use a larger, more representative sample of engineers, with more senior engineers and a broader geographic and organizational diversity. A longitudinal study of engineers over the course of several years of LLM use would lend itself to a study of the transitional-exposure pattern described in Chapter 5, instead of just a cross-segmental study. Future research might also combine self-reported survey data with objective hiring and labour-market data to confirm the relationship between perceived risk and perceived new job creation and actual job outcomes.

REFERENCES

[1] Liu, J., Wang, K., Chen, Y., Peng, X., Chen, Z., Zhang, L. and Lou, Y., 2024. Large language model-based agents for software engineering: A survey. ACM Transactions on Software Engineering and Methodology. https://dl.acm.org/doi/pdf/10.1145/3796507

[2] Sarma, W., Tiwari, S. and Dey, S., 2024. Revolutionizing software engineering with generative AI and large language models: Strategies for innovation and efficiency. unpublished. https://www.researchgate.net/profile/Sundar-Tiwari-3/publication/387831903_Revolutionizing_Software_Engineering_with_Generative_AI_and_Large_Language_Models_Strategies_for_Innovation_and_Efficiency/links/677f1edcfb021f2a47e31b34/Revolutionizing-Software-Engineering-with-Generative-AI-and-Large-Language-Models-Strategies-for-Innovation-and-Efficiency.pdf

[3] Chukwuere, J.E., 2024. Conceptualizing predictive conceptual model for unemployment rates in the implementation of Industry 4.0: Exploring machine learning techniques. arXiv preprint arXiv:2403.13536. https://arxiv.org/pdf/2403.13536

[4] A. Zarifhonarvar, "Economics of ChatGPT: A labour market view on the occupational impact of artificial intelligence," *Journal of Electronic Business & Digital Economics*, vol. 3, no. 2, pp. 100–116, 2024.

[5] D. Chen, C. Kane, A. Kozlowski, N. Kunievsky, and J. A. Evans, "The (short-term) effects of large language models on unemployment and earnings," arXiv:2509.15510, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2509.15510

[6] Khan, H.U., Ali, F. and Nazir, S., 2024. Systematic analysis of software development in cloud computing perceptions. *Journal of Software: Evolution and Process*, 36(2), p.e2485. https://onlinelibrary.wiley.com/doi/pdf/10.1002/smr.2485

[7] D. I. MacKay, D. Boddy, J. Brack, J. A. Diack, and N. Jones, *Labour Markets Under Different Employment Conditions*. London, U.K.: Routledge, 2024.

[8] Qiao, D., Rui, H. and Xiong, Q., 2024. AI and freelancers: Has the inflection point arrived?. https://scholarspace.manoa.hawaii.edu/bitstreams/4f39375d-59c2-4c4a-b394-f3eed7858c80/download

[9] Gopinathan, V.R., 2025. Software engineering practices for AI-driven systems: From development to deployment (MLOps perspective). International *Journal of Science, Research*

and Technology, 8(1), pp.13493-13500. https://www.ijsrat.com/index.php/ijsrat/article/download/120/114

[10] Adebayo, E. and Simakina, K., 2025. Impact of Large Language Models on Software Development Team's Work Processes: A Case Study of Product-Based Software Development Company. https://www.diva-portal.org/smash/get/diva2:2009721/FULLTEXT01.pdf

[11] Golboyz, M., 2025. Governing the Digital Transition: The Moderating Effect of Unemployment Benefits on Technology-Induced Employment Outcomes. Social Inclusion, 13. https://www.cogitatiopress.com/socialinclusion/article/viewFile/10114/4455

[12] H. Zhou, L. Wang, Y. Cao, and J. Li, "The impact of artificial intelligence on the labour market: A study based on bi[12] bliometric analysis," *Journal of Asian Economics*, vol. 98, p. 101926, 2025.

[13] Hou, X., Zhao, Y., Liu, Y., Yang, Z., Wang, K., Li, L., Luo, X., Lo, D., Grundy, J. and Wang, H., 2024. Large language models for software engineering: A systematic literature review. ACM Transactions on Software Engineering and Methodology, 33(8), pp.1-79. https://scholar.google.com/scholar?output=instlink&q=info:gp2dTSmtt5cJ:scholar.google.com/&hl=en&as_sdt=0,5&as_ylo=2024&scillfp=13229277304947231299&oi=lle

[14] K. Park, Y. J. Choe, Y. Jiang, and V. Veitch, "The geometry of categorical and hierarchical concepts in large language models," *arXiv preprint arXiv:2406.01506*, 2024.

[15] L. Barrault, P. A. Duquenne, M. Elbayad, A. Kozhevnikov, B. Alastruey, P. Andrews, *et al.*, "Large concept models: Language modelling in a sentence representation space," *arXiv preprint arXiv:2412.08821,* 2024.

[16] L. Chen, D. Xia, Z. Jiang, X. Tan, L. Sun, and L. Zhang, "A conceptual design method based on concept–knowledge theory and large language models," *Journal of Computing and Information Science in Engineering*, vol. 25, no. 2, p. 021001, 2025.

[17] M. A. K. Raiaan, M. S. H. Mukta, K. Fatema, N. M. Fahad, S. Sakib, M. M. J. Mim, *et al.*, "A review on large language models: Architectures, applications, taxonomies, open issues and challenges," *IEEE Access*, vol. 12, pp. 26839–26874, 2024.

[18] M. Alenezi and M. Akour, "AI-driven innovations in software engineering: A review of current practices and future directions," *Applied Sciences*, vol. 15, no. 3, p. 1344, 2025.

[19] Li, N., Gao, C., Li, M., Li, Y. and Liao, Q., 2024, August. Econagent: large language model-empowered agents for simulating macroeconomic activities. In Proceedings of the 62nd

Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 15523-15536). https://aclanthology.org/2024.acl-long.829.pdf

[12] Arora, C., Grundy, J. and Abdelrazek, M., 2024. Advancing requirements engineering through generative ai: Assessing the role of llms. In Generative AI for Effective Software Development (pp. 129-148). *Cham: Springer Nature Switzerland*. https://arxiv.org/pdf/2310.13976

[21] Q. Chen, J. Ge, H. Xie, X. Xu, and Y. Yang, "Large language models at work in China's labour market," *China Econ. Rev.*, vol. 92, p. 102413, 2025, doi: 10.1016/j.chieco.2025.102413.

[22] Q. Zhang, "The role of artificial intelligence in modern software engineering," *Applied and Computational Engineering*, vol. 97, pp. 18–23, 2024.

[23] S. Westby and A. Modestino, "The impact of generative AI on job opportunities for junior software developers," SSRN working paper, 2025.

[24] T. Chumwatana and A. K. K. Hpone, "Bridging the IT skill gap with industry demands: An AI-driven text mining approach to job market trends using a large language model," *J. Theor. Appl. Inf. Technol.*, vol. 103, no. 6, pp. 2270–2282, 2025. [Online]. Available: https://jatit.org/volumes/Vol103No6/12Vol103No6.pdf

[25] T. Eloundou, S. Manning, P. Mishkin, and D. Rock, "GPTs are GPTs: Labour market impact potential of LLMs," *Science*, vol. 384, no. 6702, pp. 1306–1308, 2024, doi: 10.1126/science.adj0998.

[26] T. Meireles, "Artificial intelligence and the labour market," *Internet Sectoral Overview*, no. 4, pp. 1–6, 2024.

[27] V. Indrašienė, V. Jegelevičienė, O. Merfeldaitė, D. Penkauskienė, J. Pivorienė, A. Railienė, *et al.*, *Critical Thinking in Higher Education and Labour Market*. Berlin, *Germany: Peter Lang International Academic Publishers*, 2021.

[28] V. Terragni, A. Vella, P. Roop, and K. Blincoe, "The future of AI-driven software engineering," *arXiv preprint arXiv:2406.07737*, 2024.

[29] X. Hou *et al.*, "*Large language models for software engineering: A systematic literature review*," *ACM Trans. Softw. Eng. Methodol.*, vol. 33, no. 8, pp. 1–79, Dec. 2024, doi: 10.1145/3695988. [Online]. Available: https://dl.acm.org/doi/full/10.1145/3695988

[33] M. Saunders, P. Lewis, and A. Thornhill, Research Methods for Business Students, 8th ed. Harlow, *U.K.: Pearson Education*, 2019.

[34] J. W. Creswell and J. D. Creswell, Research Design: Qualitative, Quantitative, and Mixed Methods Approaches, 5th ed. Thousand Oaks, CA: Sage, 2018.

[35] I. Etikan, S. A. Musa, and R. S. Alkassim, "*Comparison of convenience sampling and purposive sampling*," American *Journal of Theoretical and Applied Statistics, vol.* 5, no. 1, pp. 1-4, 2016.

[36] H. Taherdoost, "*Sampling methods in research methodology; how to choose a sampling technique for research*," *International Journal of Academic Research in Management*, vol. 5, no. 2, pp. 18-27, 2016.

[37] X. Wang and Z. Cheng, "*Cross-sectional studies: strengths, weaknesses, and recommendations*," Chest, vol. 158, no. 1, pp. S65-S71, 2020.

[38] H. Taherdoost, "*What is the best response scale for survey and questionnaire design; review of different lengths of rating scale/attitude scale/Likert scale," International Journal of Academic Research in Management*, vol. 8, no. 1, pp. 1-10, 2019.

[39] A. Field, Discovering Statistics Using IBM SPSS Statistics, 5th ed. London, U.K.: Sage, 2018.

[40] L. Sürücü and A. Maslakçı, "*Validity and reliability in quantitative research," Business & Management Studies: An International Journal*, vol. 8, no. 3, pp. 2694-2726, 2020.

[41] J. J. Vaske, J. Beaman, and C. C. Sponarski, "*Rethinking internal consistency in Cronbach's alpha*," Leisure Sciences, vol. 39, no. 2, pp. 163-173, 2017.

[42] H. H. Alharahsheh and A. Pius, "*A review of key paradigms: Positivism vs interpretivism*," Global Academic *Journal of Humanities and Social Sciences*, vol. 2, no. 3, pp. 39-43, 2020.