



**Middle East Technical University  
Informatics Institute**

# **USAGE AREAS AND POTENTIAL RISKS OF ADOPTING LLM TOOLS IN SOFTWARE PROJECTS**

**Advisor Name: Asst. Prof. Dr. Özden Özcan Top  
(METU)**

**Student Name: İpek Çobanoğlu  
(IS)**

**January 2025**

**TECHNICAL REPORT  
METU/II-TR-2025-**



**Orta Doğu Teknik Üniversitesi  
Enformatik Enstitüsü**

# **USAGE AREAS AND POTENTIAL RISKS OF ADOPTING LLM TOOLS IN SOFTWARE PROJECTS**

**Danışman Adı: Dr. Öğr. Üyesi. Özden Özcan Top  
(ODTÜ)**

**Öğrenci Adı: İpek Çobanoğlu  
(BS)**

**Ocak 2025**

**TEKNİK RAPOR  
ODTÜ/II-TR-2025-**

# REPORT DOCUMENTATION PAGE

1. AGENCY USE ONLY (Internal Use)

2. REPORT DATE

10.01.2025

3. TITLE AND SUBTITLE

USAGE AREAS AND POTENTIAL RISKS OF ADOPTING LLM TOOLS IN SOFTWARE PROJECTS

4. AUTHOR (S)

İpek Çobanoğlu

5. REPORT NUMBER (Internal Use)

METU/II-TR-2025-

6. SPONSORING/ MONITORING AGENCY NAME(S) AND SIGNATURE(S)

Informatics Master's Programme, Department of Information Systems, Informatics Institute, METU

Advisor: Assistant Prof. Dr. Özden Özcan      Signature:

Top

7. SUPPLEMENTARY NOTES

8. ABSTRACT (MAXIMUM 200 WORDS)

There is widespread use of Large Language Models (LLM) in various industries. However, the legal and ethical issues and risks of using LLM in projects are unclear. This study investigates the usage areas of LLM in software development projects, and the legal/ethical concerns and risks regarding the use of LLM. The aim of the study is to determine the usage areas of LLM and increase awareness on the legal/ethical issues and risks regarding the use of LLM in software projects. A survey was conducted with 60 participants and the results were analyzed. The results indicated that legal/ethical concerns related to informational privacy and group privacy, as well as bias, are the most widely recognized categories among participants. In contrast, concerns under the ethical auditing category—primarily related to regulations and standards governing the use of LLM tools—exhibited the lowest level of awareness among professionals. Regarding risks, the findings demonstrated a high level of awareness about the insufficiency of LLM-generated outputs for direct use, emphasizing the need for expert review and validation. Additionally, participants recognized the risk that LLM-generated outputs may run without errors but still produce incorrect results, further highlighting the necessity for thorough evaluation before deployment.

9. SUBJECT TERMS

10. NUMBER OF PAGES

45

# Table of Contents

Table Of Contents .....	i
List of Tables .....	ii
List of Figures .....	iii
Abbreviations .....	iv
Introduction .....	1
Related Work .....	2
Methodology .....	4
Aim of the Study .....	4
Survey Instrument Development.....	4
Survey Organization .....	4
Data Collection Process .....	6
Ethical Clearance .....	6
Results.....	7
Preparation of Data for Analysis .....	7
Demographics of the Participants .....	8
RQ1: What are the usage areas of LLM tools in software projects?.....	11
RQ2: What are the legal/ethical concerns regarding the use of LLM tools in the software industry?.....	15
RQ3: What are the risks of adopting LLM tools for software development? .....	17
Discussions.....	21
LLM Usage Areas.....	21
Legal/Ethical Concerns of Using LLM.....	22
Risks of Using LLM.....	22
Limitations.....	24
Conclusions .....	25
References .....	26
APPENDICES.....	29

# List of Tables

Table 1: Number of Papers in Each Iteration.....	4
Table 2: Number of Survey Statements of Each Project Phase.....	5
Table 3: Number of Survey Statements of Each Category of Legal/Ethical Problems .....	5
<i>Table 3 cont.</i> .....	6
Table 4: Open Ended and Likert-Scale Questions.....	6
Table 5: Classified Role Responses.....	7
Table 6: Planning Phase - LLM Usage Areas.....	11
Table 7: Management Phase - LLM Usage Areas.....	12
Table 8: Requirements Phase - LLM Usage Areas .....	12
Table 9: Design Phase - LLM Usage Areas .....	13
Table 10: Coding Phase - LLM Usage Areas .....	13
<i>Table 10 cont.</i> .....	14
Table 11: Testing Phase - LLM Usage Areas.....	14
Table 12: Deployment Phase - LLM Usage Areas.....	14
<i>Table 12 cont.</i> .....	15
Table 13: Maintenance Phase - LLM Usage Areas .....	15
Table 14: Legal/Ethical Concerns.....	16
<i>Table 14 cont.</i> .....	17
Table 15: Risks of Using LLM Tools in Software Projects .....	18
Table 16: Responses to Open-Ended Questions .....	19

# List of Figures

Figure 1: Age Distribution.....	8
Figure 2: Gender Distribution .....	8
Figure 3: Country Distribution.....	9
Figure 4: Work Model Distribution .....	9
Figure 5: Years of Experience Distribution .....	9
Figure 6: Number of Employees Distribution .....	10
Figure 7: Role Distribution.....	10
Figure 8: Industry Distribution .....	11
Figure 9: Distribution of Increase in Awareness of Legal and Ethical Issues .....	20
Figure 10: Distribution of Increase in Awareness of Risks .....	20

## Abbreviations

Abbreviation	Meaning
SLR	Systematic Literature Review
LLM	Large Language Model

# Introduction

Large Language Models (LLMs) are capable of approximating human-level performance across a range of tasks. They have emerged as advanced artificial intelligence systems capable of processing and generating text with coherent communication while demonstrating generalization across multiple tasks. (Naveed et al., 2024)

They are mostly used in programming tasks, such as code generation, code analysis and debugging. The use of LLMs increases in several domains, such as computing and literature. There are advantages and disadvantages in using LLM. These tools offer several advantages, including enhanced productivity, reduced time required for specific tasks, improved understanding of programming concepts, and more effective evaluation of code quality. However, they also present certain limitations. Due to their non-deterministic nature, they may produce inconsistent outputs. Additionally, understanding the generated output can be challenging, and there are potential security risks associated with their use. Furthermore, achieving the desired outcomes often depends on the user's ability to craft precise and accurate prompts. (Etsenake & Nagappan, 2024)

Although several LLM tools, such as ChatGPT<sup>1</sup>, Gemini<sup>2</sup> and GitHub Copilot<sup>3</sup> are widely used in software development, the risks associated with their use remain unclear. This study's aim is to determine the usage areas of LLM in software projects, and the legal/ethical concerns and risks arising from using LLM, while increasing awareness on legal/ethical issues and risks of using LLM.

The research questions for this study are provided:

- RQ1: What are the usage areas of LLM tools in software projects?
- RQ2: What are the legal/ethical concerns regarding the use of LLM tools in the software industry?
- RQ3: What are the risks of adopting LLM tools for software development?

To address the research questions, a quantitative research methodology was applied. A survey was conducted to 60 participants. The findings of the study indicate an increase in participants' awareness of the legal and ethical challenges and risks associated with the use of LLMs.

The rest of the paper is structured as follows:

In Related Work section, literature review is provided. In Methodology section, research methodology is provided and the making of the survey is explained. In Results section, results of the survey are provided. In Discussions section, discussions on results are provided. In Limitations section, limitations of the study are mentioned. In Conclusion section, conclusions are made, and future work are suggested.

---

<sup>1</sup> OpenAI. 2023. ChatGPT. <https://chat.openai.com/chat>

<sup>2</sup> Google. 2023. Gemini. <https://gemini.google.com/>

<sup>3</sup> GitHub. 2023. GitHub Copilot. <https://copilot.github.com/>



## Related Work

There are studies on the usage areas of LLM in software projects. Neyem et al. (2024) mentioned that LLM is used for code maintenance, generating documentation, receiving task recommendation and collaborative problem solving. Zhang et al. (2023) stated that generating personas for understanding potential user needs and defining the features of the software, and analyzing and processing user feedback data are usage areas of LLM. Adapa et al. (2024) mentioned code review, code formatting and automating code review as also usage areas of LLM. Fan et al. (2023) stated that LLM is used to automatically repair and fix bugs, summarize codes and translate code into natural language. Oswal (2024) stated that LLM is used to transform requirements into user stories. Davila (2024) mentioned that auto-completing code snippets to reduce coding effort is another usage area of LLM. Fu & Tantithamthavorn (2022) found that LLM is used for agile story point estimation. Vito et al. (2024) mentioned that LLM is used to generate unit and acceptance tests. Dhar (2024) found that generating architectural design decisions for a given context is a usage area of LLM. Hamer (2024) stated that LLM is used for software security testing. The findings of these studies will be utilized to develop a survey for participants. The survey aims to determine which of the usage areas identified in these studies are recognized by participants. Additionally, new usage areas will be identified through self-developed survey statements on LLM tool applications in software projects.

For legal/ethical concerns, the studies in literature are as follows. Olson (2024) mentioned that compliance with GDPR and EU AI Act is a concern for using LLM. Zhenfeng (2024) stated that data privacy and security is a concern. Xia et al. (2024) mentioned the accountability issue because of lack of transparency, and concern for biased decision making of LLM. Ebert & Louridas (2023) mentioned the concerns for backdoors of LLM, which also leads to data privacy. Also, they mention that LLM is unable to explain how it reached the output, which raises reliability concern. Davila et al. (2024) stated that concerns arise because of the usage of LLM outputs without human intervention. Gupta et al. (2023) mentioned several legal and ethical concerns, which are the possibility of malicious users to use personal information, unauthorized access to user conversations and the biased outputs that reflect harmful prototypes. The findings of these studies will be utilized to develop a survey for participants. This survey aims to identify which of the legal and ethical concerns highlighted in these studies have a high level and low level of awareness among participants. Additionally, it seeks to enhance awareness of legal and ethical concerns related to LLM tools in software projects.

There are also some studies about the risks of using LLM in literature. Davila et al. (2024) stated that LLM generates inadequate coding decisions, its suggestions are usually lack of context. Also, LLM has limited support for non-popular programming languages, which might be a problem for complex and large scaled tasks. Kondratenko et al. (2023) mentioned that there is scarcity for skilled talents for AI and it is a risk. Davila et al. (2024) states that LLM has limited capacity in understanding the context, which raises accountability problems for the output. Also, they mention that using LLM is risky for large scaled tasks because of the token limit. Scoccia (2023) highlights the risk of the impact of LLM on software development landscape. It means that extensive use of LLM may lead to a decrease in code qualities, as non-specialized developers rely heavily on code generation without fully understanding the underlying principles. Chen et al. (2023) stated that LLM may generate highly generic goal models which are not specific enough to be useful in a specific domain. Spinellis (2024) mentions that LLM may generate outdated results due to time lag in training data and may generate erroneous codes, which runs without errors but produces incorrect results. The findings of these studies will also be utilized to develop a survey for participants. The survey aims to identify which of the risks identified in these studies have a high level and low level of

awareness among participants. Additionally, it seeks to enhance awareness risks related to the usage of LLM tools in software projects.

Etsenake & Nagappan (2024) conducted a literature survey to explore the role of LLM tools in programming. They explored the effect of LLM on human interaction, user capabilities and task performances of users. They stated that LLM tools improves productivity. However, they also emphasized the need to improve the usability of LLM tools since it is challenging to understand the outputs that LLM generates and to write precise prompts.

Gan et al. (2024) studied on the security, privacy and ethical issues regarding with the use of LLM tools. The study categorizes threats and their effects. They proposed a framework to provide a categorized taxonomy for security, privacy and ethical issues. These issues are analyzed through case studies, which provided perception on how the weaknesses of LLM tools appear in real world scenarios. The case studies were conducted to show the applicability of the proposed framework. In addition to the study by Gan et al. (2024), this study aims to measure the awareness level of legal and ethical issues and enhance awareness among individuals.

He et al. (2024) studied on the security and privacy problems that LLM agents have. The threats were categorized, and case studies were conducted to provide insights into the challenges. The case studies were conducted as conceptual simulations or hypothetical scenarios, rather than real world implementations. Unlike the study conducted by He et al. (2024), this study incorporates real participants through a survey to assess their level of awareness regarding security and privacy concerns. Additionally, it aims to enhance awareness of these issues among participants.

Although the research discussed above focuses on the role, security, privacy, and ethical concerns of LLM tools and explores their effects on human interaction, user capabilities, task performance, and potential threats through literature reviews and case studies, a gap remains in assessing real-world awareness levels of these issues and improving the usability of LLM tools to enhance user understanding and interaction.

# Methodology

In this section, research methodology is provided. The aim of this section is to state the aim of the study and the methodology used to answer the provided research questions.

## Aim of the Study

The purpose of this study is to identify the usage areas of LLM tools in software projects and to raise awareness about the legal and ethical concerns, as well as the potential risks, associated with their use. In line with the purpose of the study, a systematic literature review was conducted and the findings were utilized to develop a survey. Afterwards, a survey was conducted with software development professionals.

## Survey Instrument Development

### Systematic Literature Review (SLR)

In order to prepare the survey questions, a Systematic Literature Review (SLR) was conducted using “(Generative AI OR GAI OR GPT) + ((software development) OR (software project) + (use OR usage OR integration)) + (benefit\* OR risk\* OR challenge\* OR threat\* OR issue\*)” keywords in IEEE Xplore. The number of papers remaining at the end of each iteration is presented in Table 1.

*Table 1: Number of Papers in Each Iteration*

Iteration	Number of Articles
Initial	164
Iteration 1	31
Iteration 2	25

The inclusion criteria were;

- Relevance to the research questions of the study. Abstracts of the papers were analyzed in relation to the research questions, and irrelevant papers were excluded.
- Information extractability. The papers were analyzed through 3 categories, which are the usage areas of LLM tools in software projects, legal/ethical concerns regarding the use of LLM tools and risks of adopting LLM tools. If no information related to any of these three categories could be extracted from a paper, the paper was excluded in the second iteration.

The referenced papers and the extracted information are given in Appendix A.

In order to generate survey statements, extracted answers were classified according to their context.

## Survey Organization

Based on the categories derived from the SLR, the survey statements were organized into four main sections, which are Demographics, Usage Areas of LLM Tools in Software Projects, Legal/Ethical Concerns of Using LLM Tools in Software Projects and Risks of Using LLM Tools in Software Projects. The survey included 115 items in total.

The sections are detailed in the following sections.

## Demographics

Demographics section had nine questions:

- Age
- Gender
- Country they live in
- Role at work
- Industry they work for
- Work model (hybrid, remote, office)
- Years of experience in software projects
- Number of employees in their companies
- Company they work for

## Usage Areas of LLM Tools in Software Projects

This section involved statements about usage areas of LLM tools in eight phases of software projects, which are planning, management, requirements, design, coding, testing, deployment, maintenance.

The number of survey statements each project phase included are given in Table 2.

*Table 2: Number of Survey Statements of Each Project Phase*

Phase of Software Project	Number of Survey Statements
Planning	13
Management	4
Requirements	12
Design	9
Coding	19
Testing	5
Deployment	9
Maintenance	5

The origins of survey statements are given in Appendix B.

## Legal/Ethical Concerns of Using LLM Tools in Software Projects

This section involved statements about the legal/ethical concerns of using LLM tools in software projects. It was separated into six parts, which were the categories for legal/ethical problems.<sup>4</sup>

The number of survey statements each category included are given in Table 3.

*Table 3: Number of Survey Statements of Each Category of Legal/Ethical Problems*

Category of Legal/Ethical Problems	Number of Survey Statements
Ethical Auditing	4
Informational Privacy and Group Privacy	5
Opacity	6
Autonomy	2
Bias	3

<sup>4</sup> (Council of Europe. (n.d.). *Common ethical challenges in AI*. <https://www.coe.int/en/web/human-rights-and-biomedicine/common-ethical-challenges-in-ai>)

Table 3 cont.

Discrimination	3
----------------	---

The origins of survey statements are given in Appendix C.

## Risks of Using LLM Tools in Software Projects

This section involved statements about the risks of using LLM tools in software projects. The origins of survey statements are given in Appendix D. It included nineteen statements in total. At the end of this section, there were two open-ended questions and two 5-point Likert scale items. They are given in Table 4.

Table 4: Open Ended and Likert-Scale Questions

Question Type	Question
Open-Ended Question	112. Is there any other risk that you have witnessed or encountered before, aside from the points mentioned above? If so, could you briefly explain it?
Open-Ended Question	113. Are there any other potential risks you are aware of, aside from the points mentioned above? If so, could you briefly explain them?
5-Likert Scale Question	114. How much has your awareness of legal and ethical issues arising from the use of LLM tools in software projects increased after answering this survey?
5-Likert Scale Question	115. How much has your awareness of the risks arising from the use of LLM tools in software projects increased after answering this survey?

## Data Collection Process

The survey was distributed through several channels. E-mails were sent to all IS students, colleagues and some other people who work in software projects. Also, survey was shared on LinkedIn to reach out to professionals who work in any phase of software projects and use LLM tools.

The survey was accessible to participants via Google Forms. In total, 60 participants participated in survey and survey data analysis part started after closing the survey. The results are mentioned in Results section.

## Ethical Clearance

In order to conduct the survey with the participants working in software projects and using LLM tools, ethical clearance from the board should have been taken. Therefore, approval of all ethical and experimental procedures and protocols was granted by the Human Subjects Ethics Committee of Middle East Technical University with Protocol No: 0048-ODTUIAEK-2025. The ethical clearance approval is presented in Appendix E.

# Results

## Preparation of Data for Analysis

A total of 60 participants completed the survey. The responses were exported for data analysis. To prepare the exported data for analysis, an Excel Macro code was developed. Initially, responses to the statement, 'I am not working on a task related to the ... process due to the nature of my job,' were excluded from the dataset. Subsequently, role and industry responses were categorized to ensure analyzable data. Role responses were classified based on the closest role that participants identified themselves with. The classified role responses are given in Table 5.

*Table 5: Classified Role Responses*

Response Role	Classified Role
Software engineer	Full-stack developer
Software engineer, Backend developer	Backend developer
Software engineer, Frontend developer	Frontend developer
Machine learning engineer, AI engineer, Deep learning engineer	AI engineer
Software engineer, Software architect	Software architect
Software engineer, Backend developer, Frontend developer	Full-stack developer
Full-stack developer, Frontend developer, Backend developer, Data scientist, Machine learning engineer, AI engineer, Data engineer, Deep learning engineer, Natural language processing engineer, Data analyst	Full-stack developer
Software engineer, AI engineer	AI engineer
Software engineer, Software architect, Data scientist, Machine learning engineer, Data engineer, Deep learning engineer	Software architect
Software engineer, Software architect, Embedded software engineer	Software architect
Software engineer, Machine learning engineer, Natural language processing engineer	AI engineer
Software engineer, Machine learning engineer, AI engineer, Data engineer, Deep learning engineer	AI engineer
Business analyst, Project manager	Business analyst
Business analyst, Digital project manager	Business analyst
Software engineer, Devops engineer	Devops engineer
Data scientist, Machine learning engineer, AI engineer, Deep learning engineer, Natural language processing engineer	AI engineer
Machine learning engineer, AI engineer, Deep learning engineer, Natural language processing engineer	AI engineer
AI engineer, Data engineer	AI engineer
Data scientist, Data analyst	Data analyst
Software engineer, Full-stack developer, Backend developer, Devops engineer, Data scientist, Machine learning engineer, AI engineer, Deep learning engineer, Natural language processing engineer, Data analyst	Full-stack developer

Industry responses did not need classification. Only “Consumer durables (white appliances)” response was classified into “Consumer durables”.

## Demographics of the Participants

Among the 60 participants, 8% of them were between 18-24 years old, 68% of them were between 25-34 years old, 22% of them were between 35-44 years old and 2% of them were between 55-64 years old. This distribution indicates that the majority of participants were in the 25-34 age group. The detailed distribution is illustrated in Figure 1.

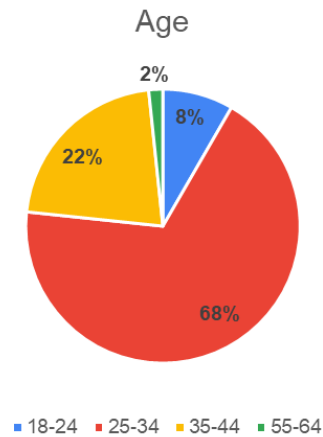


Figure 1: Age Distribution

The gender distribution of participants is presented in Figure 2. A total of 62% of the participants were male, while 38% were female.

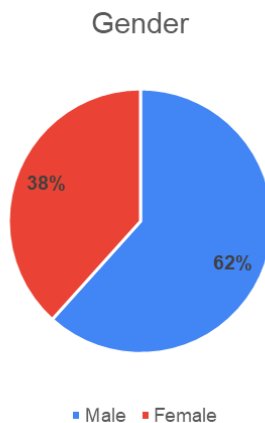


Figure 2: Gender Distribution

The country distribution of participants is presented in Figure 3. The majority of participants resided in Türkiye with 52 out of 60 participants living there. Additionally, 4 participants were from Germany, 2 from the USA, 1 from the Netherlands, and 1 from Portugal.

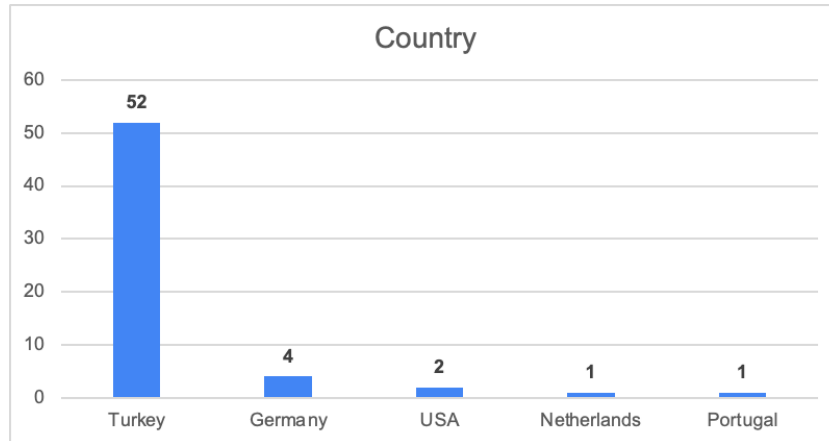


Figure 3: Country Distribution

The work model distribution of participants is presented in Figure 4. Among the 60 participants, 34 worked in an office setting, 20 followed a hybrid work model, and 6 worked remotely.

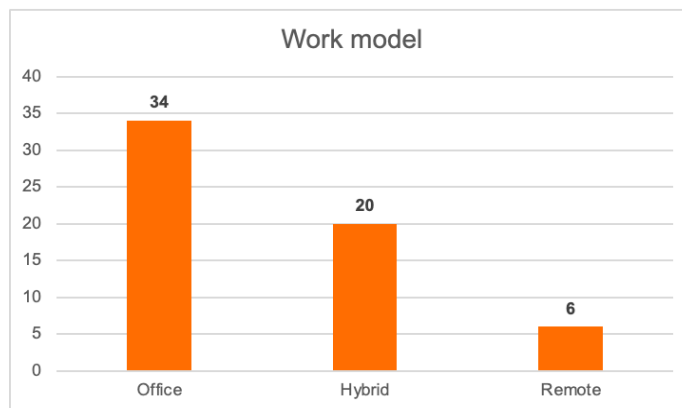


Figure 4: Work Model Distribution

The years of industry experience among participants were also surveyed, and the distribution is presented in Figure 5. Among the 60 participants, 27 had 1-3 years of experience, 17 had 4-6 years, 6 had 7-10 years, and 10 had more than 10 years of experience in the industry.

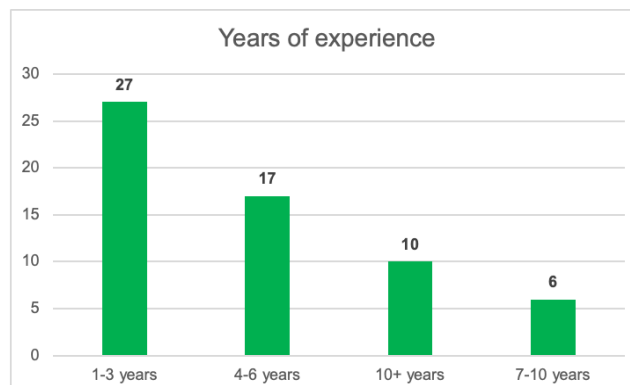


Figure 5: Years of Experience Distribution

The number of employees in the participants' companies was also surveyed, and the distribution is presented in Figure 6. A total of 36 participants were employed in companies with more than 1,000 employees.



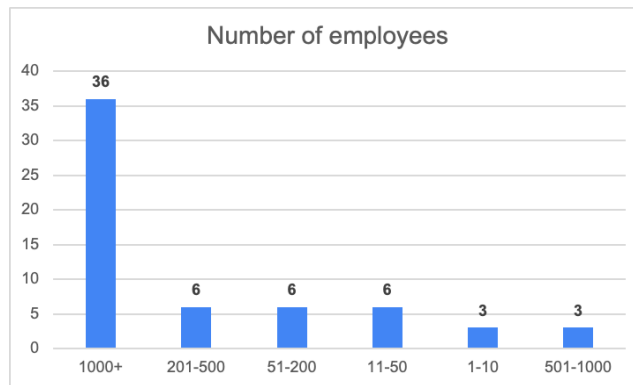


Figure 6: Number of Employees Distribution

Role distribution is illustrated in Figure 7. Among the participants, 26 individuals—nearly half of the total—were full-stack developers. The second-largest role group consisted of AI engineers, with 8 participants. Additionally, the survey included 4 business analysts, 4 data analysts, 4 DevOps engineers, 4 software architects, 2 project managers, 2 software consultants, 1 digital transformation engineer, 1 frontend developer, 1 reliability engineer, 1 backend developer, 1 software test engineer and 1 system engineer, making up the remaining participants.

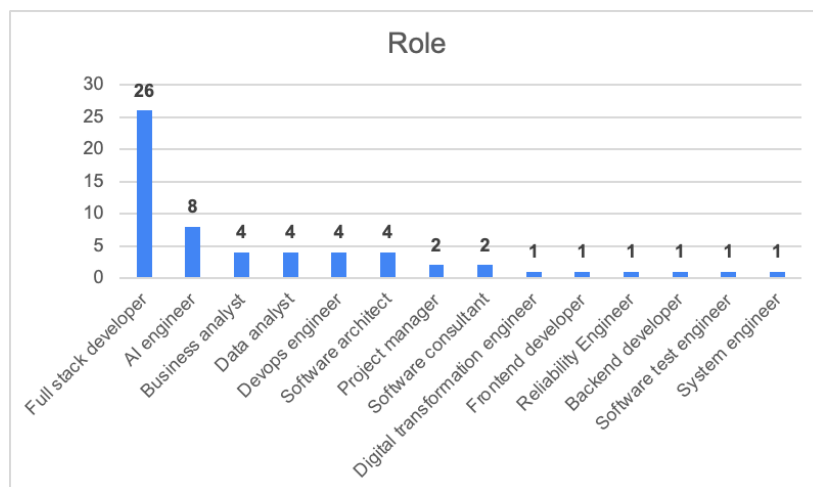


Figure 7: Role Distribution

Industry distribution is illustrated in Figure 8. Among the participants, 31 individuals—accounting for half of the total—were from the defense industry. The remaining participants were distributed across various industries as follows: 3 in automotive, 3 in finance/banking, 2 in retail, 2 in consumer durables, 2 in healthcare, and 2 in automation. Additionally, there was 1 participant each from consulting, academia, e-commerce, education, energy, geographical information systems, online employment, public services, R&D and cybersecurity, SaaS, software, startups, technology, telecom, and various other industries.

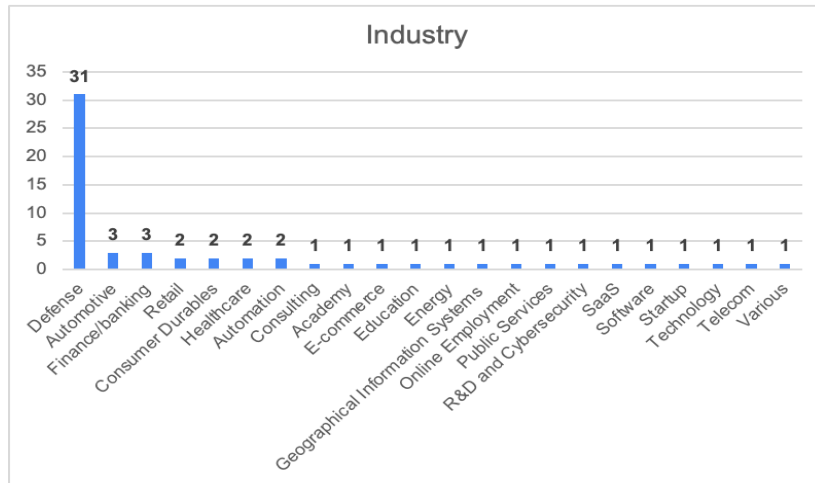


Figure 8: Industry Distribution

## RQ1: What are the usage areas of LLM tools in software projects?

Participants were asked to select the usage areas of LLM tools in their software projects. They had the option to skip specific project phases if they were not involved in them. For instance, participants who were not engaged in planning activities could opt out of the planning phase.

The analysis focused on the most frequently selected usage areas overall, as well as those most frequently selected by participants working in the defense industry. Due to the uneven distribution of participants across industries, with the defense sector being overrepresented, the analysis primarily considered data from the defense industry. A descriptive analysis was conducted to examine these trends. The results are detailed by phases in the following sections.

### Planning

In the planning phase, 31 participants indicated that they were not involved in planning activities. The remaining participants selected the usage areas relevant to their roles. The survey statements and the frequency of their selection are presented in Table 6.

Table 6: Planning Phase - LLM Usage Areas

Planning-Survey Statement	Total
2. Task suggestions for a project plan/reviewing the tasks in the plan.	13
3. Defining the project scope and project constraints.	11
9. Identifying project risks.	8
11. Identifying key milestones in the project.	8
10. Developing mitigation and resolution strategies for identified risks.	7
12. Creating the Project Work Breakdown Structure (WBS).	5
7. Creating a project timeline based on tasks and their estimated durations.	3
6. Estimating the size of functions in the project/iteration (e.g., story points, COSMIC function points).	2
8. Planning a sprint/iteration.	2
4. Creating a resource and budget plan.	1
5. Estimating time and effort for project tasks.	1

Getting ideas for technical solutions of certain phases.	1
--	---

According to the results, the most frequently selected usage area was 'Task suggestions for a project plan/reviewing the tasks in the plan.'

An industry-based analysis revealed that 'Identifying project risks' was the most frequently selected usage area in the defense industry.

## Management

For management phase, 35 participants indicated that they were not involved in management activities.

The remaining participants selected the usage areas relevant to their roles. The survey statements and the frequency of their selection are presented in Table 7.

*Table 7: Management Phase - LLM Usage Areas*

Management-Survey Statement	Total
15. Summarizing project documents.	16
16. Using it as an interactive partner for collaborative problem-solving.	15
17. Developing ethical/legal guidelines for AI systems	1

According to the results, the most frequently selected usage area was 'Summarizing project documents'.

An industry-based analysis revealed that 'Using it as an interactive partner for collaborative problem-solving.' was the most frequently selected usage area in the defense industry Requirements

For requirements phase, 27 participants indicated that they were not involved in requirements activities.

The remaining participants selected the usage areas relevant to their roles. The survey statements and the frequency of their selection are presented in Table 8.

*Table 8: Requirements Phase - LLM Usage Areas*

Requirements-Survey Statement	Total
21. Detailing requirements (e.g., transforming high-level requirements into use cases)	19
27. Ensuring the documentation of requirements (e.g., creating requirement documents, making format adjustments to the documents, and maintaining version control by documenting requirement changes over time)	13
25. Analyzing requirements (e.g., transforming natural language requirements into pseudocode, logical models, or domain-specific languages (DSL))	11
19. Developing user profiles (personas) to understand potential user needs/	9
20. Identifying requirements (e.g., simulating discussions with stakeholders and getting assistance in collecting relevant information, etc.)	9
24. Determining and improving the quality of requirements (e.g., identifying missing requirements, inconsistencies in requirements)	9
26. Gathering feedback on requirements (e.g., simulating different stakeholders to ensure impartial evaluation of the requirements)	9
22. Structuring requirements (e.g., categorizing them as functional, non-functional, etc., and determining relationships between requirements)	7
28. Modifying requirements (e.g., receiving alternative suggestions that meet similar goals when a requirement is changed, automatically updating related requirements or documents to reflect specific changes)	5
29. Prioritizing requirements (e.g., based on importance, complexity, and time constraints).	4
23. Matching user profiles (personas) with requirements.	3

According to the results, the most frequently selected usage area was 'Detailing requirements (e.g., transforming high-level requirements into use cases)'.

An industry-based analysis revealed that the same statement was the most frequently selected usage area in the defense industry.

## Design

For design phase, 21 participants indicated that they were not involved in design activities. The remaining participants selected the usage areas relevant to their roles. The survey statements and the frequency of their selection are presented in Table 9.

*Table 9: Design Phase - LLM Usage Areas*

Design-Survey Statement	Total
<b>35. Generating small code prototypes to demonstrate how a software feature or component can be developed (as a starting point for a more detailed design).*</b>	32
38. Receiving suggestions for software design patterns suitable for the project's requirements (e.g., Observer, Factory, Strategy, or Facade design patterns).	15
31. Making software architecture decisions and outlining key points by receiving suggestions on different design types such as monolithic or microservices based on the provided content.	14
32. Creating software design documents (e.g., architectural diagrams, workflows, user stories).	11
34. Describing UML diagrams such as class diagrams, sequence diagrams, etc. (to be later visualized using appropriate tools).	10
37. Checking whether different parts of the code follow a consistent design approach (e.g., whether dependency injection is used consistently across modules, whether error handling is uniform, etc.).	10
33. Defining the software architecture based on the non-functional requirements of the software.	9
36. Reviewing whether the final product aligns with the targeted design principles (e.g., checking for deviations from modularity, Separation of Concerns (SoC), and SOLID principles (Single Responsibility Principle, Open/Closed Principle, Liskov Substitution Principle, Interface Segregation Principle, and Dependency Inversion Principle)).	6

According to the results, the most frequently selected usage area was 'Generating small code prototypes to demonstrate how a software feature or component can be developed (as a starting point for a more detailed design)'.

An industry-based analysis revealed that the same statement was the most frequently selected usage area in the defense industry.

## Coding

For coding phase, 6 participants indicated that they were not involved in coding activities. This finding indicates that the majority of participants actively write code in software projects.

The remaining participants selected the usage areas relevant to their roles. The survey statements and the frequency of their selection are presented in Table 10.

*Table 10: Coding Phase - LLM Usage Areas*

Coding-Survey Statement	Total
<b>40. Writing code. *</b>	37
44. Improving existing code.	36
41. Formatting/reorganizing the code.	34
50. Optimizing existing code in terms of performance, memory usage, and readability.	23
52. Summarizing the code to understand its functionality.	21
53. Improving the reviewability/explainability of the code.	21
49. Converting data in a specific format to be used as input for another program.	15
48. Translating code into natural language or different programming languages.	14

Table 10 cont.

57. Documenting the code (e.g., scripts, etc.).	14
54. Receiving suggestions for secure coding practices.	13
51. Creating code or software components from high-level requirements.	12
46. Automatically completing code snippets.	11
56. Creating code documentation (including code comments etc.) and synchronizing it with code updates.	11
45. Code integration.	8
47. Retrieving relevant information from code repositories.	8
43. Automating code review.	6
42. Analyzing the newly added code in pull requests or commits.	5
55. Detecting/identifying sensitive data and potential vulnerabilities to detect potential cyberattacks or malicious viruses.	5

According to the results, the most frequently selected usage area was 'Writing code'. An industry-based analysis revealed that , the same statement was the most frequently selected usage area in the defense industry.

## Testing

For testing phase, 25 participants indicated that they are were involved in testing activities. The remaining participants selected the usage areas relevant to their roles. The survey statements and the frequency of their selection are presented in Table 11.

Table 11: Testing Phase - LLM Usage Areas

Testing-Survey Statement	Total
<b>60. Writing unit test code.*</b>	24
59. Generating test scenarios from requirements.	17
61. Generating test scenarios for different tests such as regression tests, integration tests, etc.	14
62. Testing software security.	6

According to the results, the most frequently selected usage area was 'Writing unit test code'. An industry-based analysis revealed that the same statement was the most frequently selected usage area in the defense industry.

## Deployment

For deployment phase, 35 participants indicated that they were not involved in deployment activities.

The remaining participants selected the usage areas relevant to their roles. The survey statements and the frequency of their selection are presented in Table 12.

Table 12: Deployment Phase - LLM Usage Areas

Deployment-Survey Statement	Total
<b>65. Writing commands.*</b>	13
67. Receiving improvement suggestions for pipeline configurations.	11
68. Getting assistance in writing or improving IaC scripts, such as Terraform, Ansible, etc.	10
69. Analyzing logs generated during deployment, identifying patterns in errors or warnings, and getting suggestions on troubleshooting steps or fixes.	8

Table 12 cont.

64. Writing code to optimize the deployment process (e.g., sending work packages to the next station in the development pipeline via requests sent through an API).	6
66. Ensuring that deployment environments are consistently defined and reproducible	5
70. Writing test cases that simulate deployment environments.	5
71. Identify vulnerabilities in deployment configurations or missing security best practices, such as improper access control settings or unencrypted data.	4

According to the results, the most frequently selected usage area was 'Writing commands'. An industry-based analysis revealed that the same statement was the most frequently selected usage area in the defense industry.

## Maintenance

For maintenance phase, 35 participants indicated that they were not involved in maintenance activities.

The remaining participants selected the usage areas relevant to their roles. The survey statements and the frequency of their selection are presented in Table 13.

Table 13: Maintenance Phase - LLM Usage Areas

Maintenance-Survey Statement	Total
<b>73. Automating error debugging in the code by generating concrete solutions for potential quality issues.*</b>	15
76. Automating debugging and repair in the code.	15
75. Identifying vulnerabilities during code maintenance and providing solutions to address them, thereby improving code quality and security.	11
74. Code maintenance.	5

According to the results, the most frequently selected usage area was 'Automating error debugging in the code by generating concrete solutions for potential quality issues'.

An industry-based analysis revealed that the same statement was the most frequently selected usage area in the defense industry.

## RQ2: What are the legal/ethical concerns regarding the use of LLM tools in the software industry?

Participants were asked to select the legal/ethical concerns that they were aware of in their software projects.

The analysis examined the most and least frequently selected concerns overall, by category, and by role. A descriptive analysis was conducted, and the results are detailed in the following sections. The results are detailed in the following sections.

The categories of legal and ethical concerns, the corresponding survey statements, the total number of selections, and the number of selections by full-stack developers and AI engineers are presented in Table 14.

The difference between the total count and the counts for full-stack developers and AI engineers arises because the total number represents the selections made by all

participants, while the latter figures specifically reflect selections made by full-stack developers and AI engineers.

Table 14: Legal/Ethical Concerns

Class	Legal/Ethical Concerns-Survey Statement	Total	Full-stack Developer	AI Engineer
Autonomy	89. Outputs obtained using LLM tools in software projects may not be directly usable in decision-making without human intervention, and using them in such a way may raise concerns about reliability.	55	23	7
Informational Privacy and Group Privacy	81. The confidentiality of code generated or analyzed using LLM tools may not be ensured.	52	25	7
Informational Privacy and Group Privacy	80. The privacy of data provided to LLM tools could be violated (due to unauthorized third-party access to user conversations and potential data breaches, the possibility that data provided to GPT can be reused regardless of whether it is public or not, the risk of company employees exposing confidential data by providing company data to GPT, or individuals exposing their own information by submitting it to GPT, etc.)	51	20	7
Informational Privacy and Group Privacy	83. Outputs generated using LLM tools for systems that are critically important in terms of security may be subject to security vulnerabilities or may expose sensitive information.	51	22	8
Bias	91. If the data used in training LLMs is biased, incorrect, or false, the outputs may also be biased and unreliable.	50	21	8
Informational Privacy and Group Privacy	82. Personal information provided to LLM tools may be misused by malicious users.	48	22	6
Bias	90. Outputs obtained using LLM tools may be generated in a biased manner, potentially compromising the accuracy or integrity of the information, leading to biased decision-making.	47	20	7
Opacity	87. The outputs may not be explainable due to LLM tools' inability to understand or explain the underlying logic of complex code.	45	22	7
Discrimination	93. LLM outputs may be based on information generated from intentionally incorrect, false, or discriminatory data, and could be designed to favor certain groups.	44	19	7
Opacity	84. It may be difficult to determine the authenticity and reliability of the generated code or information.	42	18	7
Opacity	85. The outputs may not be reliable because the way they were generated cannot be explained, and in the event of accountability, it may not be clear who should be held responsible or based on what criteria.	42	20	6
Opacity	86. The outputs may be generated incorrectly or in an unverified manner due to hallucinations, making them unreliable.	41	17	7
Discrimination	92. If the LLM tool is trained on biased data, the outputs generated by this tool may contain discriminatory language and reflect harmful stereotypes about different ethnic groups.	41	19	6
Ethical Auditing	77. The outputs may not comply with regulations and standards (e.g., General Data Protection Regulation (GDPR), data privacy principles, EU Artificial Intelligence Act).	40	15	7



Table 15 cont.

Ethical Auditing	78. The outputs may not comply with copyright, intellectual property rights, software licenses, or industry standards.	35	15	7
Ethical Auditing	79. The lack of effective government policies for clarifying the accountability of outputs and ensuring a transparent artificial intelligence research and development process.	31	12	6
Opacity	88. The outputs may contain compromised or malicious content.	25	10	4

### Most Frequently Selected Legal/Ethical Concern

According to

Table 14, the most frequently selected legal/ethical concern statement was “Outputs obtained using LLM tools in software projects may not be directly usable in decision-making without human intervention and using them in such a way may raise concerns about reliability.” It was selected 55 times by participants.

### Least Frequently Selected Legal/Ethical Concern

According to

Table 14, the most frequently selected legal/ethical concern statement was “The outputs may contain compromised or malicious content.” It was selected 25 times by participants.

### Most Frequently Selected Legal/Ethical Concern Classes

According to

Table 14, the “Informational Privacy and Group Privacy” category was the most frequently selected legal/ethical concern, followed by the “Bias” category. The total number of selections for survey statements under the “Informational Privacy and Group Privacy” category was 202, while the statements under the “Bias” category were selected 97 times.

### Most Frequently Selected Legal/Ethical Concern by Roles

According to Figure 7, the majority of participants were full-stack developers, followed by AI engineers. Therefore, the analysis was conducted based on these two roles. According to

Table 14, the most frequently selected legal/ethical concern among full-stack developers is “The confidentiality of code generated or analyzed using LLM tools may not be ensured.”. This concern was selected 25 times by full-stack developers. Among AI engineers, the most frequently selected legal/ethical concerns were: “Outputs generated using LLM tools for systems that are critically important in terms of security may be subject to security vulnerabilities or may expose sensitive information.” and “If the data used in training LLMs is biased, incorrect, or false, the outputs may also be biased and unreliable.”. Each of these concerns was selected 8 times by AI engineers.

### RQ3: What are the risks of adopting LLM tools for software development?

Participants were asked to select the risks they are aware of in their software projects.

The analysis examined the most and least frequently selected risks overall and by role. A descriptive analysis was conducted to explore these findings. The results are detailed in the following sections.

The risks survey statements, along with the total number of times they were selected and the number of times they were selected by full-stack developers and AI engineers, are presented in Table 16.

*Table 16: Risks of Using LLM Tools in Software Projects*

Risks-Survey Statement	Total	Full-stack Developer	AI Engineer
94. Coding decisions made by LLM tools may be insufficient, and as a result, code suggestions or shortcuts may need to be reviewed by an expert developer.	51	24	6
99. LLM tools may generate code that runs without errors but does not produce the desired result.	51	21	8
96. Incorrectly phrasing the prompt in LLM tools may result in failure to obtain the correct answer.	50	22	7
95. LLM tools may generate content that appears relevant but is actually unrelated, or produce outputs that are disconnected from the context and lack sufficient structure.	49	24	7
98. LLM tools may generate meaningless and nonsensical responses due to hallucinations (e.g., suggesting API function names that do not actually exist).	46	20	7
100. LLM tools may produce outdated results due to the possibility of training data being outdated.	43	23	7
110. Since less experienced developers may write code with the help of LLM tools without fully understanding the principles behind the algorithms, it could negatively impact the software development ecosystem.	43	20	7
109. The quality of code generated by LLM tools (e.g., meeting requirements, understandability and readability, consistency, memory usage, and optimization of processing time) may not be sufficient.	42	19	7
111. LLM tools may generate models that are too general to be useful for domain-specific applications.	40	20	6
101. Debugging the code generated by LLM tools can be challenging.	38	19	5
104. Due to the limitations of LLM tools in understanding the provided content, the outputs generated for content they do not fully comprehend may be irrelevant and inaccurate.	37	18	7
107. If LLM security is not ensured, cyberattacks (e.g., phishing, hacking, etc.) could manipulate LLM algorithms and data.	36	18	5
97. LLM tools may offer inadequate or no support for programming languages that are not widely used, compared to more common languages.	36	18	5
105. The review outputs generated by LLM tools may, in some cases, mislead the reviewer.	35	16	6
108. Malicious third parties could manipulate the responses generated by LLM tools by performing prompt injection attacks.	34	14	5
102. Understanding and maintaining complex code generated by LLM tools can be difficult.	33	18	5
106. Due to the token limit, operations on large code chunks may not be possible or may be interrupted. To avoid interruptions, the required cost may be high.	33	19	5
103. Due to the lack of a standardized approach and regulations in AI research and applications, issues may arise regarding security, ethics, and privacy, and there could be inconsistencies between different organizations and countries.	31	15	3

## Most Frequently Selected Risk

According to Table 16, the most frequently selected risk statements were: “Coding decisions made by LLM tools may be insufficient, and as a result, code suggestions or shortcuts may need to be reviewed by an expert developer.” and “LLM tools may generate code that runs without errors but does not produce the desired result.” Each of these statements was selected 51 times by participants.

## Least Frequently Selected Risk

According to Table 16, the least frequently selected risk statement was “Due to the lack of a standardized approach and regulations in AI research and applications, issues may arise regarding security, ethics, and privacy, and there could be inconsistencies between different organizations and countries.” It was selected 31 times by participants.

## Most Frequently Selected Risk by Roles

According to Figure 7, the majority of participants were full-stack developers, followed by AI engineers. Therefore, the analysis was conducted based on these two roles. According to Table 16, the most frequently selected risks among full-stack developers were: “Coding decisions made by LLM tools may be insufficient, and as a result, code suggestions or shortcuts may need to be reviewed by an expert developer.” and “LLM tools may generate content that appears relevant but is actually unrelated, or produce outputs that are disconnected from the context and lack sufficient structure.”. Each of these statements was selected 24 times by full-stack developers.

The most frequently selected risk was “LLM tools may generate code that runs without errors but does not produce the desired result.” among AI engineers. It was selected 8 times by AI engineers.

## Open-Ended Questions

At the end of the risk section of the survey, the respondents were asked two open-ended questions given in Table 4. The responds to the questions are presented in Table 17.

*Table 17: Responses to Open-Ended Questions*

Question	Response
112. Is there any other risk that you have witnessed or encountered before, aside from the points mentioned above? If so, could you briefly explain it?	Prompt manipulation: Although the LLM tool is designed with good intentions, it is possible to access unethical answers or responses the LLM tool does not want to provide, by asking manipulative questions.
112. Is there any other risk that you have witnessed or encountered before, aside from the points mentioned above? If so, could you briefly explain it?	It makes people lazy. I tend to get help from the tool rather than working on the algorithms myself.
112. Is there any other risk that you have witnessed or encountered before, aside from the points mentioned above? If so, could you briefly explain it?	Even if the LLM tool's token limit is sufficient, it still cannot give reasonable answers because of its inability due to chain-of-thoughts and context reasoning.
113. Are there any other potential risks you are aware of, aside from the points mentioned above? If so, could you briefly explain them?	Since the awareness of data privacy risks is high in our company, we use the LLM tool in our local network that our company's AI department developed.

## 5-Likert Scale Questions

Following the open-ended questions at the end of the risk section of the survey, participants were asked to rate the increase in their awareness of legal and ethical concerns and risks on a scale from 1 to 5, where 1 represents the least increase and 5 represents the highest increase in awareness. According to responses to the question “114. How much has your awareness of legal and ethical issues arising from the use of LLM tools in software projects increased after answering this survey?”, the distribution of awareness increase is presented in Figure 9. The results indicate that awareness has increased among 60% of participants.

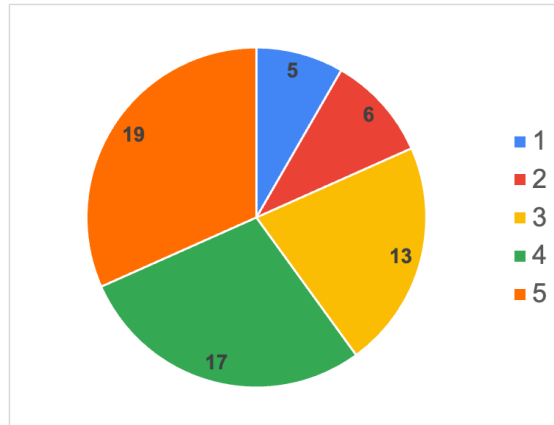


Figure 9: Distribution of Increase in Awareness of Legal and Ethical Issues

According to responses to the question “115. How much has your awareness of the risks arising from the use of LLM tools in software projects increased after answering this survey?”, the distribution of awareness increase is presented in Figure 10. The results indicate that awareness of risks has increased among 58% of participants.

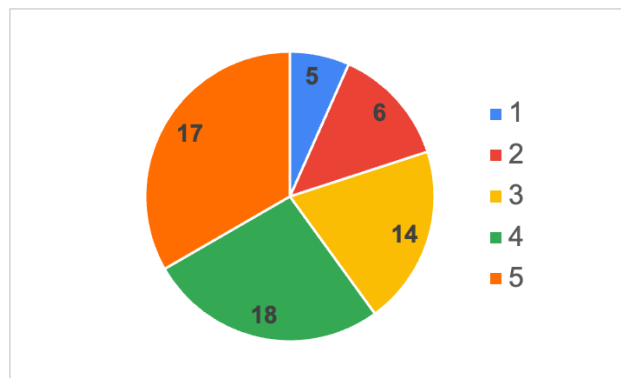


Figure 10: Distribution of Increase in Awareness of Risks

# Discussions

## LLM Usage Areas

According to the findings given in Results section, the most frequently usage area of LLM tools in planning phase of software projects is receiving task suggestions for a project plan/reviewing the tasks in the plan. This aligns with the findings of Zhenfeng et al. (2024), who also identified this as a common application of LLM tools in software projects, thereby supporting their study. For defense industry, LLM is mostly used for identifying the project risks, according to the results. This usage area was not identified in the systematic literature review, so it is a new finding regarding the usage of LLM tools in planning phase of software projects.

For management phase, the most frequently usage area is summarizing project documents. Ebert & Louridas (2023) and Hu & Chen (2023) also stated that summarizing project documents is an observed usage area of LLM. In defense industry, LLM is mostly used for using it as an interactive partner for collaborative problem-solving, according to the results It supports the study of Neyem et al. (2024).

For requirements phase, LLM tools are mostly used for detailing requirements, such as transforming the high-level requirements into use cases, etc. It is the same for defense industry, too. Notably, this specific usage area was not identified in any of the studies reviewed in the systematic literature review, making it a novel finding regarding the application of LLM tools in the requirements phase of software projects. For design phase, the most frequently usage area is generating small code prototypes to demonstrate how a software feature or component can be developed, which will be used as a starting point for a more detailed design. It is the same for defense industry, too. Particularly, this usage area was also not identified in any of the studies reviewed in the systematic literature review. Therefore, it can be considered as a novel finding regarding the application of LLM tools in the design phase of software projects.

For coding phase, writing code and improving the existing code are the most frequently used areas of LLM. Writing code is the most frequently usage area in defense industry. Fan et al. (2024), Ebert & Louridas (2023), Xiao et al. (2024), Hu & Chen (2023) and Scoccia (2023) identified writing code as a key application in their studies. Fan et al. (2024), Ebert & Louridas (2023), Elvira et al. (2024) and Hu & Chen (2023) identified improving the existing code as a key application in their studies. Consequently, the survey results align with and support the findings of the systematic literature review. For testing phase, writing unit test code is the most used area of LLM both in defense industry and other industries. Writing unit test code as a usage area was also not identified in any of the studies reviewed in the systematic literature review, making it a novel finding regarding the application of LLM tools in the testing phase of software projects.

For deployment phase, writing commands is the most frequently used area both in defense industry and other industries. It was not identified in any of the studies reviewed in the systematic literature review. Therefore, it is also a new finding regarding the application of LLM tools in the deployment phase of software projects.

For maintenance phase, automating error debugging in the code by generating concrete solutions for potential quality issues, and automating debugging and repair in the code are the two areas that are most frequently used both in defense industry and other industries. Automating error debugging in the code by generating concrete solutions for potential quality issues was also identified as a specific usage area of LLM tools in software projects by Ebert & Louridas (2023) and Xiao et al. (2024). Additionally, the systematic literature review

referenced automating debugging and repair in code from five studies. Therefore, the survey results align with and support the findings of the systematic literature review.

## Legal/Ethical Concerns of Using LLM

The legal/ethical concern which has the highest level of awareness is that the outputs obtained using LLM tools in software projects may not be directly usable in decision-making without human intervention and using them in such a way may raise concerns about reliability. The result supports the findings of systematic literature review, since it was also identified as one of the legal and ethical concerns in the studies of Kondratenko et al. (2023) and Davila et al. (2024). Other than this statement, people are mostly aware of the legal/ethical concerns in “Informational Privacy and Group Privacy” and “Bias” categories. It means that people are highly aware that LLM tools are not trusted in terms of data privacy and the outputs are not completely reliable because of the biased nature of LLM.

Full-stack developers are mostly aware that the confidentiality of code generated or analyzed using LLM tools may not be ensured, which is also about data privacy. Particularly, this specific concern was identified as one of the legal and ethical concerns in the studies of Hamer et al. (2024), Davila et al. (2024) and Ebert & Louridas (2023). Therefore, the most frequently selected concern among full-stack developers aligns with and supports the findings of the systematic literature review.

AI engineers are mostly aware that the outputs generated using LLM tools for systems that are critically important in terms of security may be subject to security vulnerabilities or may expose sensitive information. It is also about data privacy. This specific concern was derived as a survey statement from the study of Scoccia (2023). Since it was the most frequently selected legal/ethical concern by AI engineers, the finding supports the specific study.

AI engineers also are mostly aware that if the data used in training LLMs is biased, incorrect, or false, the outputs may also be biased and unreliable. It is related to the biased nature and the unreliability of LLM outputs. The studies of Kondratenko et al. (2023), Xia et al. (2024), and Ebert & Louridas (2023) were used as references in formulating this survey statement. Since this concern was also among the most frequently selected legal and ethical issues by AI engineers, the finding aligns with and supports these studies.

The legal/ethical concern which has the lowest level of awareness is that the outputs may contain compromised or malicious content. Apart from this statement, people are not very aware of the legal/ethical concerns in “Ethical Auditing” category. The category is primarily about the regulations, government policies, intellectual property rights and standards, indicating that these topics are not perceived as major concerns by the participants. One possible explanation for this lack of awareness is the absence of well-defined regulations or policies on LLM usage. However, awareness of these issues may increase if governments implement concrete policies. This finding shows that the specific legal and ethical concern identified by Zhenfeng et al. (2024) and Ebert & Louridas (2023) is not as widely recognized compared to other identified legal and ethical concerns."

## Risks of Using LLM

The risks which have the highest level of awareness among participants are, first, the coding decisions made by LLM tools may be insufficient, and as a result, code suggestions or shortcuts may need to be reviewed by an expert developer. Second, LLM tools may generate code that runs without errors but does not produce the desired result. The first result supports the findings of Davila et al. (2024), and the second result supports the findings of Spinellis (2024), as these issues were identified as risks in their studies. Notably, these findings indicate that participants recognize the importance of using LLM tools as a starting point rather than relying on their outputs directly. Additionally, there is an awareness that LLM-generated outputs require verification, as they may appear sufficient but may not function as intended.

Full-stack developers are highly aware that coding decisions made by LLM tools may be insufficient, and as a result, code suggestions or shortcuts may need to be reviewed by an expert developer, as all other roles. In addition, they are also highly aware that the LLM tools may generate content that appears relevant but is actually unrelated or produce outputs that are disconnected from the context and lack sufficient structure. This finding indicates that full-stack developers are highly aware that LLM-generated outputs cannot be directly used for a task without thorough review and verification. Additionally, they recognize that these outputs may require modifications or additions to fully meet the requirements of a given task. These findings also support the study of Davila et al. (2024), since the specific risks they identified are widely recognized among full-stack developers. AI engineers are highly aware that LLM tools may generate code that runs without errors but does not produce the desired result. This finding further indicates that they recognize the possibility of outputs appearing sufficient while potentially not functioning correctly, reinforcing the need for thorough review and validation. . This finding also aligns with the study of Spinellis (2024), since the specific risk he identified is widely recognized among AI engineers.

The risk which has the lowest level of awareness is that due to the lack of a standardized approach and regulations in AI research and applications, issues may arise regarding security, ethics, and privacy, and there could be inconsistencies between different organizations and countries. It is a compatible result with the legal/ethical concerns that have least awareness level. It means that people are not aware that there is not a standardized approach and regulations due to LLM tool usage and there are issues that may arise as a result. This finding shows that the specific risk identified by Kondratenko et al. (2023) is not as widely recognized compared to other identified risks.

## Limitations

In this study, the number of participants was 60. A larger sample size would have resulted in a more balanced distribution across industries and roles, facilitating more detailed analyses based on these factors, as well as on years of experience or age. The industry distribution of participants was not suitable for conducting statistical analysis or drawing industry-specific conclusions, as 52% of the participants were from the defense industry. Expanding the participant pool to include more professionals from other industries would improve the generalizability of the findings.

Similarly, the role distribution was also uneven, limiting the possibility of role-based statistical analysis or conclusions. Notably, 43% of the participants were full-stack developers. Including a more diverse range of roles in the survey would allow for meaningful comparisons between different roles.



## Conclusions

The objective of this study was to identify the usage areas of LLM tools in software projects and to raise awareness of the legal/ethical concerns, as well as the risks associated with their use. To achieve this goal, a systematic literature review (SLR) was conducted, and a survey was developed based on the SLR findings. The survey was conducted to 60 participants, and the results were analyzed.

The findings revealed that the legal and ethical concern with the highest level of awareness was the use of LLM outputs in decision-making processes without human intervention. Additionally, participants were highly aware of the inadequacies of LLM tools in making independent coding decisions, emphasizing the necessity for expert review. Another widely recognized risk was the potential for LLM-generated code to execute without errors but fail to deliver the desired results.

The study also highlighted variations in awareness across different classes of legal and ethical issues. Informational privacy and group privacy were the most recognized categories, whereas awareness of ethical auditing—primarily related to regulations and government policies—was significantly lower. These results underscore the need to increase awareness about the lack of regulations and the potential legal and ethical problems that may arise from using LLM-generated outputs. Corporate training programs could be effective to address this gap.

To enhance the study, future research could aim to increase the sample size and achieve a more balanced distribution across industries, roles, and age groups. Additionally, further studies could investigate the relationship between the quality of products and the use of LLM tools in software development. For instance, the quality of products developed using LLM tools during specific stages of the software development lifecycle could be systematically analyzed.

## References

- Adapa, C., Avulamanda, S.S., Anjana, A.R.K., Victor, A. (2024, April 23). *AI-Powered Code Review Assistant for Streamlining Pull Request Merging*. In *2024 IEEE International Conference for Women in Innovation, Technology & Entrepreneurship (ICWITE)*.<https://doi.org/10.1109/ICWITE59797.2024.10503540>
- Basole, R.C., Major, T. (2024, March 25). *Generative AI for Visualization: Opportunities and Challenges*. In *IEEE Computer Graphics and Applications*.<https://doi.org/10.1109/MCG.2024.3362168>
- Chen, B., Chen, K., Hassani, S., Yang, Y., Amyot, D., Lessard, L., Mussbacher, G., Sabetzadeh, M., Varrault, D. (2023, September 28). *On the Use of GPT-4 for Creating Goal Models: An Exploratory Study*. In *2023 IEEE 31st International Requirements Engineering Conference Workshops (REW)*.<https://doi.org/10.1109/REW57809.2023.00052>
- Davila, N., Wiese, I., Steinmacher, I., Da Silva, L. L., Kawamoto, A., Peres Favaro, G. J., & Nunes, I. (2024, June 18). *An industry case study on adoption of AI-based programming assistants*. In *2024 IEEE/ACM 46th International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*. <https://doi.org/10.1145/3639477.3643648>
- Dhar, R., Vaidhyathan, K., Varma, V. (2024, July 24). *Can LLMs Generate Architectural Design Decisions? - An Exploratory Empirical Study*. In *2024 IEEE 21st International Conference on Software Architecture (ICSA)*.<https://doi.org/10.1109/ICSA59870.2024.00016>
- Davila, N., Melegati, J., Wiese, I. (2024, July 18). *Tales from the Trenches: Expectations and Challenges from Practice for Code Review in the Generative AI Era*. In *IEEE Software*.<https://doi.org/10.1109/MS.2024.3428439>
- Ebert, C., Louridas, P. (2023, July 07). *Generative AI for Software Practitioners*. In *IEEE Software*.<https://doi.org/10.1109/MS.2023.3265877>
- Etsenake, D., & Nagappan, M. (2024). Understanding the human-LLM dynamic: A literature survey of LLM use in programming tasks. *Proceedings of ACM Conference (Conference'17)*.
- Elvira, T., Procko, T.T., Couder, J.O., Ochoa, O. (2024, April 09). *Digital Rubber Duck: Leveraging Large Language Models for Extreme Programming*. In *2023 Congress in Computer Science, Computer Engineering, & Applied Computing (CSCE)*.<https://doi.org/10.1109/CSCE60160.2023.00051>
- Fu, M., Tantithamthavorn, C. (2022, March 10). *GPT2SP: A Transformer-Based Agile Story Point Estimation Approach*. In *IEEE Transactions on Software Engineering*.<https://doi.org/10.1109/TSE.2022.3158252>

Fan, A., Gokkaya, B., Harman, M., Lyubarskiy, M., Sengupta, S., Yoo, S., Zhang, J.M. (2024, March 04). *Large Language Models for Software Engineering: Survey and Open Problems*. In *2023 IEEE/ACM International Conference on Software Engineering: Future of Software Engineering (ICSE-FoSE)*.<https://doi.org/10.1109/ICSE-FoSE59343.2023.00008>

Gupta, M., Akiri, C., Aryal, K., Parker, E., Praharaj, L. (2023, August 01). *From ChatGPT to ThreatGPT: Impact of Generative AI in Cybersecurity and Privacy*. In *IEEE Access*.<https://doi.org/10.1109/ACCESS.2023.3300381>

Gan, Y., Yang, Y., Ma, Z., He, P., Zeng, R., Wang, Y., Li, Q., Zhou, C., Li, S., Wang, T., Gao, Y., Wu, Y., & Ji, S. (2024). Navigating the risks: A survey of security, privacy, and ethics threats in LLM-based agents. *arXiv*. <https://arxiv.org/abs/2411.09523>

He, F., Zhu, T., Ye, D., Liu, B., Zhou, W., & Yu, P. S. (2024). The emerged security and privacy of LLM agent: A survey with case studies. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 37(4), Article 111.

Hamer, S., d'Amorim, M., Williams, L. (2024, July 04). *Just another copy and paste? Comparing the security vulnerabilities of ChatGPT generated code and StackOverflow answers*. In *2024 IEEE Security and Privacy Workshops (SPW)*.<https://doi.org/10.1109/SPW63631.2024.00014>

Hu, C., Chen, J. (2023, October 24). *A Dimensional Perspective Analysis on the Cybersecurity Risks and Opportunities of ChatGPT-Like Information Systems*. In *2023 International Conference on Networking and Network Applications (NaNA)*.<https://doi.org/10.1109/NaNA60121.2023.00061>

Kondratenko, Y., Shevchenko, A., Zhukov, Y., Kondratenko, G., Striuk, O. (2023, December 21). *Tendencies and Challenges of Artificial Intelligence Development and Implementation*. In *2023 IEEE 12th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*.<https://doi.org/10.1109/IDAACS58523.2023.10348800>

Naveed, H., Khan, A.U., Qiu, S., Saqib, M., Anwar, S., Usman, M., Akhtar, M., Barnes, N., Mian, A. (2024, October 18). *A Comprehensive Overview of Large Language Models*. arXiv preprint arXiv:2303.18223.

Neyem, A., Gonzalez, L.A., Mendoza, M., Alcocer, J.P.S., Centellas, L., Paredes, C. (2024, May 03). *Toward an AI Knowledge Assistant for Context-Aware Learning Experiences in Software Capstone Project Development*. In *IEEE Transactions on Learning Technologies*.<https://doi.org/10.1109/TLT.2024.3396735>

Olson, L. (2024, June 18). *Custom Developer GPT for Ethical AI Solutions*. In *2024 IEEE/ACM 3rd International Conference on AI Engineering ,À Software Engineering for AI (CAIN)*.

Oswal, J.U., Kanakia, H.T., Suktel, D. (2024, March 22). *Transforming Software Requirements into User Stories with GPT-3.5 -: An AI-Powered Approach*. In *2024 2nd International*

*Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT)*.<https://doi.org/10.1109/IDCIoT59759.2024.10467750>

Scoccia, G.L. (2023, November 02). *Exploring Early Adopters' Perceptions of ChatGPT as a Code Generation Tool*. In *2023 38th IEEE/ACM International Conference on Automated Software Engineering Workshops (ASEW)*.<https://doi.org/10.1109/ASEW60602.2023.00016>

Spinellis, D. (2024, April 05). *Pair Programming With Generative AI*. In *IEEE Software*.<https://doi.org/10.1109/MS.2024.3363848>

Vito, G.D., Lambiase, S., Palomba, F., Ferrucci, F. (2024, January 01). *Meet C4SE: Your New Collaborator for Software Engineering Tasks*. In *2023 49th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*.<https://doi.org/10.1109/SEAA60479.2023.00044>

Wuisang, M.C., Kurniawan, M., Santosa, K.A.W., Gunawan, A.A.S., Saputra, K.E. (2023, October 30). *An Evaluation of the Effectiveness of OpenAI's ChatGPT for Automated Python Program Bug Fixing using QuixBugs*. In *2023 International Seminar on Application for Technology of Information and Communication (iSemantic)*.<https://doi.org/10.1109/iSemantic59612.2023.10295323>

Xia, B., Lu, Q., Zhu, L., Lee, S.U., Liu, Y., Xing, Z. (2024, June 18). *Towards a Responsible AI Metrics Catalogue: A Collection of Metrics for AI Accountability*. In *2024 IEEE/ACM 3rd International Conference on AI Engineering, Software Engineering for AI (CAIN)*.

Xiao, T., Treude, C., Hata, H., Matsumoto, K. (2024, June 18). *DevGPT: Studying Developer-ChatGPT Conversations*. In *2024 IEEE/ACM 21st International Conference on Mining Software Repositories (MSR)*.

Zhang, X., Liu, L., Wang, Y., Liu, X., Wang, H., Ren, A., Arora, C. (2023, September 28). *PersonaGen: A Tool for Generating Personas from User Feedback*. In *2023 IEEE 31st International Requirements Engineering Conference (RE)*.<https://doi.org/10.1109/RE57278.2023.00048>

Zhenfeng, W., Quanwang, L., Huihui, G., Xiehua, Y. (2024, April 25). *Legal Risks and Governance Paths for Generative AI--A Case Study of ChatGPT*. In *2023 13th International Conference on Information Technology in Medicine and Education (ITME)*.<https://doi.org/10.1109/ITME60234.2023.00023>

# APPENDICES

## Appendix A

ID	Reference	LLM Usage Areas	Legal/ethical Concerns	Risks
1	Olson, L. (2024, June 18). <i>Custom Developer GPT for Ethical AI Solutions</i> . In <i>2024 IEEE/ACM 3rd International Conference on AI Engineering ,Ai Software Engineering for AI (CAIN)</i> .	-providing developers with practical application on how to comply with legal frameworks -providing alternate ethical perspectives	complying with GDPR and EU AI Act	-
3	Neyem, A., Gonzalez, L.A., Mendoza, M., Alcocer, J.P.S., Centellas, L., Paredes, C. (2024, May 03). <i>Toward an AI Knowledge Assistant for Context-Aware Learning Experiences in Software Capstone Project Development</i> . In <i>IEEE Transactions on Learning Technologies</i> . <a href="https://doi.org/10.1109/TLT.2024.3396735">https://doi.org/10.1109/TLT.2024.3396735</a>	-code maintenance -generating documentation, syncing code updates and documentation -retrieving knowledge -task recommendation -collaborative problem solving	-	-
5	Zhenfeng, W., Quanwang, L., Huihui, G., Xiehua, Y. (2024, April 25). <i>Legal Risks and Governance Paths for Generative AI--A Case Study of ChatGPT</i> . In <i>2023 13th International Conference on Information Technology in Medicine and Education (ITME)</i> . <a href="https://doi.org/10.1109/ITME60234.2023.00023">https://doi.org/10.1109/ITME60234.2023.00023</a>	-	data privacy and security copyright	copyright infringement generating infringing content and inability to foresee which specific infringing content will be generated
7	Zhang, X., Liu, L., Wang, Y., Liu, X., Wang, H., Ren, A., Arora, C. (2023, September 28). <i>PersonaGen: A Tool for Generating Personas from User Feedback</i> . In <i>2023 IEEE 31st International Requirements Engineering Conference (RE)</i> . <a href="https://doi.org/10.1109/RE57278.2023.00048">https://doi.org/10.1109/RE57278.2023.00048</a>	-generating personas , which helps in understanding potential user needs and defining software features -Cleaning, Integration, Prediction, and Analysis of User Feedback , processes user feedback data -Addressing Non-Functional Requirements (NFRs) through personas	-	-
12	Xia, B., Lu, Q., Zhu, L., Lee, S.U., Liu, Y., Xing, Z. (2024, June 18). <i>Towards a Responsible AI Metrics Catalogue: A Collection of Metrics for AI Accountability</i> . In <i>2024 IEEE/ACM 3rd International Conference on AI Engineering ,Ai Software Engineering for AI (CAIN)</i> .	-	-data privacy breaches -accountability issues because of lack of transparency -erosion of informational integrity. mostly concerned for	-

			biased decision making -complying to regulations and standards	
14	Adapa, C., Avulamanda, S.S., Anjana, A.R.K., Victor, A. (2024, April 23). <i>AI-Powered Code Review Assistant for Streamlining Pull Request Merging. In 2024 IEEE International Conference for Women in Innovation, Technology &amp; Entrepreneurship (ICWITE)</i> . <a href="https://doi.org/10.1109/ICWITE59797.2024.10503540">https://doi.org/10.1109/ICWITE59797.2024.10503540</a>	code formatting code review-analyzing newly added code in pull requests or commits automating code review	-	-
16	Fan, A., Gokkaya, B., Harman, M., Lyubarskiy, M., Sengupta, S., Yoo, S., Zhang, J.M. (2024, March 04). <i>Large Language Models for Software Engineering: Survey and Open Problems. In 2023 IEEE/ACM International Conference on Software Engineering: Future of Software Engineering (ICSE-FoSE)</i> . <a href="https://doi.org/10.1109/ICSE-FoSE59343.2023.00008">https://doi.org/10.1109/ICSE-FoSE59343.2023.00008</a>	-code generation -code optimization, generating optimized versions of existing codes -automatically repair, fix bugs etc. -synthesize programs from high-level requirements that meet specific criteria -code summarization -code refactoring, to improve structure, readability, maintainability -help to enable developers to interact with code using natural language	-	-
20	Oswal, J.U., Kanakia, H.T., Suktel, D. (2024, March 22). <i>Transforming Software Requirements into User Stories with GPT-3.5 -: An AI-Powered Approach. In 2024 2nd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCloT)</i> . <a href="https://doi.org/10.1109/IDCloT59759.2024.10467750">https://doi.org/10.1109/IDCloT59759.2024.10467750</a>	transforming software requirements into user stories	-	-
21	Ebert, C., Louridas, P. (2023, July 07). <i>Generative AI for Software Practitioners. In IEEE Software</i> . <a href="https://doi.org/10.1109/MS.2023.3265877">https://doi.org/10.1109/MS.2023.3265877</a>	automating repetitive tasks (testing, req.traceability) creating test suites from requirements (improves software quality) automating workflows by routing work products to the next suitable step in a production pipeline to enhance creativity automating tasks (testing, debugging, deployment) assisting in code generation, refactoring, improving existing code Maintaining legacy	cybersecurity (malicious code or code containing backdoors) privacy of the generated or analyzed code difficult to determine the authenticity and reliability of the generated code or information misinformation and fake content potential risk of bias or discrimination (GAI can inherit biases from data they are trained	neither deterministic nor explainable-challenges in understanding and validating the output quality and reliability issues security and privacy concerns misinformation and fake content ethical implications-AI systems making decisions that could have significant consequences without a true understanding of the real world.

		software by assisting in code maintenance, integration, and understanding Summarizing documentation, reviews, interviews, and meeting minutes. Assisting in the identification of missing requirements, inconsistencies, and potential risks	on,leading to biased outputs) lack of explainability (unable to explain how it reached to the output)	
22	Davila, N., Wiese, I., Steinmacher, I., Da Silva, L. L., Kawamoto, A., Peres Favaro, G. J., & Nunes, I. (2024, June 18). <i>An industry case study on adoption of AI-based programming assistants</i> . In <i>2024 IEEE/ACM 46th International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)</i> . <a href="https://doi.org/10.1145/3639477.3643648">https://doi.org/10.1145/3639477.3643648</a>	auto-completing code snippets to reduce typing effort asking questions (reducing online searches) finding alternatives for unfamiliar problems building solutions	-	inadequate coding decisions lack of context in suggestions copyrights, IP, software licenses, industry standards limited support for non-popular programming languages trustworthiness - because of black box process of GAI
25	Fu, M., Tantiathamthavorn, C. (2022, March 10). <i>GPT2SP: A Transformer-Based Agile Story Point Estimation Approach</i> . In <i>IEEE Transactions on Software Engineering</i> . <a href="https://doi.org/10.1109/TSE.2022.3158252">https://doi.org/10.1109/TSE.2022.3158252</a>	agile story point estimation	-	-
27	Vito, G.D., Lambiase, S., Palomba, F., Ferrucci, F. (2024, January 01). <i>Meet C4SE: Your New Collaborator for Software Engineering Tasks</i> . In <i>2023 49th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)</i> . <a href="https://doi.org/10.1109/SEAA60479.2023.00044">https://doi.org/10.1109/SEAA60479.2023.00044</a>	code suggestions code review GitHub API operations-retrieving information from repositories or managing tasks unit and acceptance test generation	-	-
28	Xiao, T., Treude, C., Hata, H., Matsumoto, K. (2024, June 18). <i>DevGPT: Studying Developer-ChatGPT Conversations</i> . In <i>2024 IEEE/ACM 21st International Conference on Mining Software Repositories (MSR)</i> .	-code generation -code completion -code summarization -program repair -code review -debugging	-	-
29	Kondratenko, Y., Shevchenko, A., Zhukov, Y., Kondratenko, G., Striuk, O. (2023, December 21). <i>Tendencies and Challenges of Artificial Intelligence Development and Implementation</i> . In <i>2023 IEEE 12th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)</i> . <a href="https://doi.org/10.1109/IDAACS58523.2023.10348800">https://doi.org/10.1109/IDAACS58523.2023.10348800</a>	-	- -transparency -bias -accountability	-The lack of interpretability in decision-making due to the use of artificial neural networks (ANN) -The scarcity of skilled talent in the field of AI -transparency -bias -accountability -need for effective governance frameworks to ensure responsible and transparent AI

				research and development -limitations and reliance on human intervention in current AI systems -need for standardized approaches and regulations in AI research and implementation
30	Dhar, R., Vaidhyanathan, K., Varma, V. (2024, July 24). <i>Can LLMs Generate Architectural Design Decisions? - An Exploratory Empirical Study. In 2024 IEEE 21st International Conference on Software Architecture (ICSA).</i> <a href="https://doi.org/10.1109/ICSA59870.2024.00016">https://doi.org/10.1109/ICSA59870.2024.00016</a>	automatic code documentation and summarization generating architectural design decisions from a given context	privacy concern for private data slowness of adoption of Architecture Decision Records (ADRs)	privacy concern for private data slowness of adoption of Architecture Decision Records (ADRs)-due to challenges such as time constraints, inconsistent uptake, inadequate tool support, effort needed to capture Architecture Knowledge (AK), interruptions to the design process caused by documenting AK, and uncertainty regarding which AK needs documentation.
32	Davila, N., Melegati, J., Wiese, I. (2024, July 18). <i>Tales from the Trenches: Expectations and Challenges from Practice for Code Review in the Generative AI Era. In IEEE Software.</i> <a href="https://doi.org/10.1109/MS.2024.3428439">https://doi.org/10.1109/MS.2024.3428439</a>	-code review -code understanding, visualization and easing code comprehension -improve reviewability of the code -feedback automation, including identifying violations of standards, antipatterns, and performance bottlenecks, and suggesting actionable code fixes	-trustworthiness of generative AI models without human intervention -confidentiality of private code -compliance with security and privacy policies	-trustworthiness because of black box structure -limitations of GAI in understanding the context, which can affect the accuracy and relevance of their suggestions -Misleading Review Comments -confidentiality of private code -compliance with security and privacy policies -token limit, makes it difficult to process large code review changes
33	Basole, R.C., Major, T. (2024, March 25). <i>Generative AI for Visualization: Opportunities and Challenges. In IEEE Computer Graphics and Applications.</i> <a href="https://doi.org/10.1109/MCG.2024.3362168">https://doi.org/10.1109/MCG.2024.3362168</a>			
37	Gupta, M., Akiri, C., Aryal, K., Parker, E., Praharaj, L. (2023, August 01). <i>From ChatGPT to ThreatGPT: Impact of Generative AI in Cybersecurity and Privacy. In IEEE</i>	-automated code review -suggest secure coding practices by providing alternative solutions that comply with secure coding standards -analyze security-	-Misuse of personal information by malicious users -Controversy over data ownership and rights, potential lack of legal rights to the	-vulnerabilities of ChatGPT that can be exploited by malicious users -Jailbreak attacks, where malicious entities can



	Access: <a href="https://doi.org/10.1109/ACCESS.2023.3300381">https://doi.org/10.1109/ACCESS.2023.3300381</a>	related data, such as network logs and security event alerts, to identify potential attack patterns and behaviors -developing ethical guidelines for AI systems by generating natural language explanations and recommendations based on existing ethical frameworks and principles	information used by ChatGPT, regardless of whether the information is public or not -Unauthorized access to user conversations and data breaches -users have received biased outputs reflecting harmful stereotypes when using ChatGPT for gathering data or writing articles/essays -Misuse by organizations and employees, samsung case where employees put confidential information about company to chatGPT and make it publicly accessible -Hallucinations and misinformation, raising concerns about accuracy and reliability	manipulate the model to generate unauthorized content -Reverse psychology attacks, malicious users can use reverse psychology techniques to trick ChatGPT into generating harmful or misleading content -Prompt injection attacks, malicious users can inject specific prompts to manipulate the generated responses -Cyber offense, cyber offenders can develop various cyber attacks such as social engineering attacks, phishing attacks, automated hacking, attack payload generation, malware creation, and polymorphic malware
44	Hamer, S., d,ÃdAmorim, M., Williams, L. (2024, July 04). <i>Just another copy and paste? Comparing the security vulnerabilities of ChatGPT generated code and StackOverflow answers. In 2024 IEEE Security and Privacy Workshops (SPW).</i> <a href="https://doi.org/10.1109/SPW63631.2024.00014">https://doi.org/10.1109/SPW63631.2024.00014</a>	-code summarization -code translation -automated program repair -vulnerability detection -security code reviews -software security testing	-	-security vulnerabilities -Insecure code propagation -Lack of awareness of developers -Software supply chain risk
48	Elvira, T., Procko, T.T., Couder, J.O., Ochoa, O. (2024, April 09). <i>Digital Rubber Duck: Leveraging Large Language Models for Extreme Programming. In 2023 Congress in Computer Science, Computer Engineering, &amp; Applied Computing (CSCE).</i> <a href="https://doi.org/10.1109/CSCE60160.2023.00051">https://doi.org/10.1109/CSCE60160.2023.00051</a>	-Code refactoring -Code walkthroughs, explaining code line-by-line in natural language -Coverage testing	-	-
73	Wuisang, M.C., Kurniawan, M., Santosa, K.A.W., Gunawan, A.A.S., Saputra, K.E. (2023, October 30). <i>An Evaluation of the Effectiveness of OpenAI's ChatGPT for Automated Python Program Bug Fixing using QuixBugs. In 2023 International Seminar on Application for Technology of Information and Communication (iSemantic).</i> <a href="https://doi.org/10.11">https://doi.org/10.11</a>	-automated bug fixing	-	-

	09/iSemantic59612.2023.10295323			
74	Hu, C., Chen, J. (2023, October 24). <i>A Dimensional Perspective Analysis on the Cybersecurity Risks and Opportunities of ChatGPT-Like Information Systems</i> . In <i>2023 International Conference on Networking and Network Applications (NaNA)</i> . <a href="https://doi.org/10.1109/NaNA60121.2023.00061">https://doi.org/10.1109/NaNA60121.2023.00061</a>	<ul style="list-style-type: none"> <li>-Checking for bugs in code segments</li> <li>-Refactoring code expression</li> <li>-Converting meeting recordings to text and summarizing with ChatGPT</li> <li>-Generating code in multiple programming languages to meet user needs</li> <li>-Writing software documentation, such as scripts, reports, contracts, and papers</li> </ul>	<ul style="list-style-type: none"> <li>-Data Security Risks, data leakage and privacy breaches</li> <li>-ChatGPT can be used to generate false information, spread fake news, and impersonate the language style of specific individuals or groups, reliability of information generated by generative AI and its potential for defamation or spreading misinformation</li> <li>-Network Security Risks, generative AI can be used to generate realistic text, making it a powerful tool for criminal activities</li> <li>-Regulatory Compliance, GDPR</li> <li>-Lack of Transparency and Explainability, concerns about their trustworthiness and potential vulnerabilities</li> </ul>	-
79	Scoccia, G.L. (2023, November 02). <i>Exploring Early Adopters' Perceptions of ChatGPT as a Code Generation Tool</i> . In <i>2023 38th IEEE/ACM International Conference on Automated Software Engineering Workshops (ASEW)</i> . <a href="https://doi.org/10.1109/ASEW60602.2023.00016">https://doi.org/10.1109/ASEW60602.2023.00016</a>	<ul style="list-style-type: none"> <li>-generate code snippets</li> <li>-assist developers in their programming tasks</li> </ul>	-	<ul style="list-style-type: none"> <li>-quality of Generated Code</li> <li>-Trust in Generated Code</li> <li>-Safety and Security Risks, Generated code may contain vulnerabilities or expose sensitive information, especially in safety-critical systems or applications</li> <li>-Impact on Software Development Landscape, It may lead to a decrease in code quality, as non-specialized developers rely heavily on code generation without fully understanding the underlying principles</li> </ul>
117	Chen, B., Chen, K., Hassani, S., Yang, Y., Amyot, D., Lessard, L., Mussbacher, G., Sabetzadeh, M., Varrault, D. (2023, September 28). <i>On the Use of GPT-4 for</i>	-requirement engineering (POTENTIAL, NOT USED)	-	-Hallucination, may generate plausible-looking goal models that contain syntactic or semantic errors

	<p><i>Creating Goal Models: An Exploratory Study. In 2023 IEEE 31st International Requirements Engineering Conference Workshops (REW).</i><a href="https://doi.org/10.1109/REW57809.2023.00052">https://doi.org/10.1109/REW57809.2023.00052</a></p>			<p>-Generic Outputs, may generate highly generic goal models with elements that are not specific enough to be useful in the domain</p>
16 4	<p>Spinellis, D. (2024, April 05). <i>Pair Programming With Generative AI. In IEEE Software.</i><a href="https://doi.org/10.1109/MS.2024.3363848">https://doi.org/10.1109/MS.2024.3363848</a></p>	<p>IDE assistants (like github copilot) chatbots (like chatgpt) translating code (to modernize legacy code or work with code in different languages) transforming data</p>	-	<p>answers can be stupidly wrong erroneous code (produces code that runs but produces incorrect results) generates outdated results (due to timelag in training data) generates confusing code (hard to debug and maintain) generates code with security vulnerabilities inability to reasoning for complex algorithm and understanding the logic of the lying code</p>

## Appendix B

Survey Statement ID	Related Phase of Software Project	Survey Statement	Referenced Paper ID
1	Planning	I am not working on a task related to the planning process due to the nature of my job.	-
2	Planning	Task suggestions for a project plan/reviewing the tasks in the plan.	3
3	Planning	Defining the project scope and project constraints.	Self-developed
4	Planning	Creating a resource and budget plan.	Self-developed
5	Planning	Estimating time and effort for project tasks.	Self-developed
6	Planning	Estimating the size of functions in the project/iteration (e.g., story points, COSMIC function points).	25
7	Planning	Creating a project timeline based on tasks and their estimated durations.	Self-developed
8	Planning	Planning a sprint/iteration.	Self-developed
9	Planning	Identifying project risks.	Self-developed
10	Planning	Developing mitigation and resolution strategies for identified risks.	Self-developed
11	Planning	Identifying key milestones in the project.	Self-developed
12	Planning	Creating the Project Work Breakdown Structure (WBS).	Self-developed
13	Planning	Evaluating whether the project complies with ethical values and legal regulations (e.g., GDPR compliance).	1
14	Management	I am not working on a task related to the management process due to the nature of my job.	-
15	Management	Summarizing project documents.	21, 74
16	Management	Using it as an interactive partner for collaborative problem-solving.	3
17	Management	Developing ethical/legal guidelines for AI systems	37

18	Requirements	I am not working on a task related to the requirements process due to the nature of my job.	-
19	Requirements	Developing user profiles (personas) to understand potential user needs.	7
20	Requirements	Identifying requirements (e.g., simulating discussions with stakeholders and getting assistance in collecting relevant information, etc.)	20
21	Requirements	Detailing requirements (e.g., transforming high-level requirements into use cases)	Self-developed
22	Requirements	Structuring requirements (e.g., categorizing them as functional, non-functional, etc., and determining relationships between requirements)	Self-developed
23	Requirements	Matching user profiles (personas) with requirements.	7
24	Requirements	Determining and improving the quality of requirements (e.g., identifying missing requirements, inconsistencies in requirements)	21
25	Requirements	Analyzing requirements (e.g., transforming natural language requirements into pseudocode, logical models, or domain-specific languages (DSL))	Self-developed
26	Requirements	Gathering feedback on requirements (e.g., simulating different stakeholders to ensure impartial evaluation of the requirements)	Self-developed
27	Requirements	Ensuring the documentation of requirements (e.g., creating requirement documents, making format adjustments to the documents, and maintaining version control by documenting requirement changes over time)	Self-developed
28	Requirements	Modifying requirements (e.g., receiving alternative suggestions that meet similar goals when a requirement is changed, automatically updating related requirements or documents to reflect specific changes)	Self-developed
29	Requirements	Prioritizing requirements (e.g., based on importance, complexity, and time constraints).	Self-developed
30	Design	I am not working on a task related to the	-

		requirements process due to the nature of my job.	
31	Design	Making software architecture decisions and outlining key points by receiving suggestions on different design types such as monolithic or microservices based on the provided content.	30
32	Design	Creating software design documents (e.g., architectural diagrams, workflows, user stories).	Self-developed
33	Design	Defining the software architecture based on the non-functional requirements of the software.	Self-developed
34	Design	Describing UML diagrams such as class diagrams, sequence diagrams, etc. (to be later visualized using appropriate tools).	Self-developed
35	Design	Generating small code prototypes to demonstrate how a software feature or component can be developed (as a starting point for a more detailed design).	Self-developed
36	Design	Reviewing whether the final product aligns with the targeted design principles (e.g., checking for deviations from modularity, Separation of Concerns (SoC), and SOLID principles (Single Responsibility Principle, Open/Closed Principle, Liskov Substitution Principle, Interface Segregation Principle, and Dependency Inversion Principle)).	Self-developed
37	Design	Checking whether different parts of the code follow a consistent design approach (e.g., whether dependency injection is used consistently across modules, whether error handling is uniform, etc.).	Self-developed
38	Design	Receiving suggestions for software design patterns suitable for the project's requirements (e.g., Observer, Factory, Strategy, or Facade design patterns).	Self-developed
39	Coding	I am not working on a task related to the coding process due to the nature of my job.	-
40	Coding	Writing code.	16,21,28,74,79
41	Coding	Formatting/reorganizing the code.	14,21
42	Coding	Analyzing the newly added code in pull requests or commits.	14

43	Coding	Automating code review.	14,27,28,32,37
44	Coding	Improving existing code.	16,21,48,74
45	Coding	Code integration.	21
46	Coding	Automatically completing code snippets.	22,27,28,164
47	Coding	Retrieving relevant information from code repositories.	27
48	Coding	Translating code into natural language or different programming languages.	164,16,44,48
49	Coding	Converting data in a specific format to be used as input for another program.	164
50	Coding	Optimizing existing code in terms of performance, memory usage, and readability.	16
51	Coding	Creating code or software components from high-level requirements.	16
52	Coding	Summarizing the code to understand its functionality.	16,28,32,44
53	Coding	Improving the reviewability/explainability of the code.	32
54	Coding	Receiving suggestions for secure coding practices.	37
55	Coding	Detecting/identifying sensitive data and potential vulnerabilities to detect potential cyberattacks or malicious viruses.	37,44
56	Coding	Creating code documentation (including code comments etc.) and synchronizing it with code updates.	3,30
57	Coding	Documenting the code (e.g., scripts, etc.).	74
58	Testing	I am not working on a task related to the testing process due to the nature of my job.	-
59	Testing	Generating test scenarios from requirements.	21
60	Testing	Writing unit test code.	Self-developed
61	Testing	Generating test scenarios for different tests such as regression tests, integration tests, etc.	27,44
62	Testing	Testing software security.	44

63	Deployment	I am not working on a task related to the deployment process due to the nature of my job.	-
64	Deployment	Writing code to optimize the deployment process (e.g., sending work packages to the next station in the development pipeline via requests sent through an API).	21
65	Deployment	Writing commands.	Self-developed
66	Deployment	Ensuring that deployment environments are consistently defined and reproducible	Self-developed
67	Deployment	Receiving improvement suggestions for pipeline configurations.	Self-developed
68	Deployment	Getting assistance in writing or improving IaC scripts, such as Terraform, Ansible, etc.	Self-developed
69	Deployment	Analyzing logs generated during deployment, identifying patterns in errors or warnings, and getting suggestions on troubleshooting steps or fixes.	Self-developed
70	Deployment	Writing test cases that simulate deployment environments.	Self-developed
71	Deployment	Identify vulnerabilities in deployment configurations or missing security best practices, such as improper access control settings or unencrypted data.	Self-developed
72	Maintenance	I am not working on a task related to the maintenance process due to the nature of my job.	-
73	Maintenance	Automating error debugging in the code by generating concrete solutions for potential quality issues.	21,28
74	Maintenance	Code maintenance.	21
75	Maintenance	Identifying vulnerabilities during code maintenance and providing solutions to address them, thereby improving code quality and security.	3,164
76	Maintenance	Automating debugging and repair in the code.	16,28,44,73,74



## Appendix C

Survey Statement ID	Category of Legal/Ethical Problem	Survey Statement	Referenced Paper ID
77	Ethical Auditing	The outputs may not comply with regulations and standards (e.g., General Data Protection Regulation (GDPR), data privacy principles, EU Artificial Intelligence Act).	1,12,32,74
78	Ethical Auditing	The outputs may not comply with copyright, intellectual property rights, software licenses, or industry standards.	5,22
79	Ethical Auditing	The lack of effective government policies for clarifying the accountability of outputs and ensuring a transparent artificial intelligence research and development process.	Self-developed
80	Informational privacy and group privacy	The privacy of data provided to LLM tools could be violated (due to unauthorized third-party access to user conversations and potential data breaches, the possibility that data provided to GPT can be reused regardless of whether it is public or not, the risk of company employees exposing confidential data by providing company data to GPT, or individuals exposing their own information by submitting it to GPT, etc.)	5,12,30,37,44,74,164
81	Informational privacy and group privacy	The confidentiality of code generated or analyzed using LLM tools may not be ensured.	21,32,44
82	Informational privacy and group privacy	Personal information provided to LLM tools may be misused by malicious users.	37,74
83	Informational privacy and group privacy	Outputs generated using LLM tools for systems that are critically important in terms of security may be subject to security vulnerabilities or may expose sensitive information.	79
84	Opacity	It may be difficult to determine the authenticity and reliability of the generated code or information.	21
85	Opacity	The outputs may not be reliable because the way they were generated cannot be explained, and in the event of accountability, it may not be clear who should be held	12,21,22,29,32,74,79





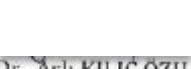



		responsible or based on what criteria.	
86	Opacity	The outputs may be generated incorrectly or in an unverified manner due to hallucinations, making them unreliable.	37
87	Opacity	The outputs may not be explainable due to LLM tools' inability to understand or explain the underlying logic of complex code.	164
88	Opacity	The outputs may contain compromised or malicious content.	5,21
89	Autonomy	Outputs obtained using LLM tools in software projects may not be directly usable in decision-making without human intervention, and using them in such a way may raise concerns about reliability.	29,32
90	Bias	Outputs obtained using LLM tools may be generated in a biased manner, potentially compromising the accuracy or integrity of the information, leading to biased decision-making.	12
91	Bias	If the data used in training LLMs is biased, incorrect, or false, the outputs may also be biased and unreliable.	12, 21,29
92	Discrimination	If the LLM tool is trained on biased data, the outputs generated by this tool may contain discriminatory language and reflect harmful stereotypes about different ethnic groups.	37
93	Discrimination	LLM outputs may be based on information generated from intentionally incorrect, false, or discriminatory data, and could be designed to favor certain groups.	37,74

## Appendix D

Survey Statement ID	Survey Statement	Referenced Paper ID
94	Coding decisions made by LLM tools may be insufficient, and as a result, code suggestions or shortcuts may need to be reviewed by an expert developer.	22
95	LLM tools may generate content that appears relevant but is actually unrelated, or produce outputs that are disconnected from the context and lack sufficient structure.	22
96	Incorrectly phrasing the prompt in LLM tools may result in failure to obtain the correct answer.	Self-developed
97	LLM tools may offer inadequate or no support for programming languages that are not widely used, compared to more common languages.	22
98	LLM tools may generate meaningless and nonsensical responses due to hallucinations (e.g., suggesting API function names that do not actually exist).	117,164
99	LLM tools may generate code that runs without errors but does not produce the desired result.	164
100	LLM tools may produce outdated results due to the possibility of training data being outdated.	164
101	Debugging the code generated by LLM tools can be challenging.	164
102	Understanding and maintaining complex code generated by LLM tools can be difficult.	164
103	Due to the lack of a standardized approach and regulations in AI research and applications, issues may arise regarding security, ethics, and privacy, and there could be inconsistencies between different organizations and countries.	29
104	Due to the limitations of LLM tools in understanding the provided content, the outputs generated for content they do not fully comprehend may be irrelevant and inaccurate.	32
105	The review outputs generated by LLM tools may, in some cases, mislead the reviewer.	32
106	Due to the token limit, operations on large code chunks may not be possible or may be interrupted. To avoid interruptions, the required cost may be high.	32
107	If LLM security is not ensured, cyberattacks (e.g.,	21,37

	phishing, hacking, etc.) could manipulate LLM algorithms and data.	
108	Malicious third parties could manipulate the responses generated by LLM tools by performing prompt injection attacks.	37
109	The quality of code generated by LLM tools (e.g., meeting requirements, understandability and readability, consistency, memory usage, and optimization of processing time) may not be sufficient.	79
110	Since less experienced developers may write code with the help of LLM tools without fully understanding the principles behind the algorithms, it could negatively impact the software development ecosystem.	79
111	LLM tools may generate models that are too general to be useful for domain-specific applications.	117

## Appendix E

<b>UYGULAMALI ETİK ARAŞTIRMA MERKEZİ</b> <b>APPLIED ETHICS RESEARCH CENTER</b>	 <b>ORTA DOĞU TEKNİK ÜNİVERSİTESİ</b> <b>MIDDLE EAST TECHNICAL UNIVERSITY</b>
DUMLUPINAR BULVARI 06800 ÇANKAYA ANKARA/TURKEY T: +90 312 210 22 91 F: +90 312 210 79 59 seam@metu.edu.tr www.usam.metu.edu.tr	
	21 OCAK 2025
<b>Konu:</b> Değerlendirme Sonucu	
<b>Gönderen:</b> ODTÜ İnsan Araştırmaları Etik Kurulu (İAEK)	
<b>İlgi:</b> İnsan Araştırmaları Etik Kurulu Başvurusu	
<b>Sayın Dr. Öğr. Üyesi Özden Özcan Top</b>	
Danışmanlığımı yürüttüğünüz İpek Çobanoğlu'nun " <i>LLM Modellerinin Yazılım Projelerinde Kullanım Alanları, Kullanımından Doğan Yasal/Etik Sorunlar ve Riskler</i> " başlıklı araştırmanız İnsan Araştırmaları Etik Kurulu tarafından uygun görülerek 0048-ODTÜİAEK-2025 protokol numarası ile onaylanmıştır	
Bilgilerinize saygılarımla sunarım	
 Prof. Dr. Ş. Halil TURAN Başkan	 Doç. Dr. Ali Emre Turgut Üye
 Prof. Dr. İ. Semih AKÇOMAK Üye	
 Doç. Dr. Aslı KILIÇ OZHAN Üye	 Doç. Dr. Murat Perit ÇAKIR Üye
 Dr. Öğretim Üyesi Süreyya OZCAN KABASAKAL Üye	 Dr. Öğretim Üyesi Müge GÜNDÜZ Üye