Tampere University of Applied Sciences

Ethical Implications of AI assisted Coding in Software Engineering

Hasan Mahmud

BACHELOR'S THESIS
June 2025

Degree Programme
Software Engineering

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Software Engineering

Hasan Mahmud
Ethical Implications of AI Assisted Coding in Software Engineering

Bachelor's thesis 27 pages
June 2025

While the bright light of the age of technology is making our life easier than ever, a new technology is on the rise with even more potential. Artificial Intelligence, shortly AI is increasingly being used in every aspect of our life more so in the creative areas like coding, Software development, software testing and a lot of others. Among the most effective and popular AI-assisted coding tools are ChatGPT, GitHub Copilot and DeepSeek. While these tools are unprecedented in suggesting code, fixing bugs and helping the developer brainstorming ideas real-time, concerns is also being raised in different ways. Developers are now changing their roles from writing code by themselves to supervising the code written by the AI. As writing prompts to AI tools is very easy developers are being over reliant on these tools which is harming their overall creative practice and skills. Very little monitoring is being done in these areas and accessing harmful and vulnerable code is very easy using AI tools. Moreover, issues like ownership, plagiarism, bias and safety are also major concerns.

As an answer to the rising concerns this thesis research aims to explore these ethical implications in AI specially in Software Engineering sector. Through a combination of theoretical analysis, case studies and by examining how developers currently are using this tool in real-time, the study aims to imply that the objective is not to prevent the use of AI but to use it responsibly and more ethically.

Key words: Artificial intelligence, Ethics, Coding, Ownership.

1 INTRODUCTION

Artificial Intelligence has changed the Software Industry in a very short amount of time. Making the developers efficient, productive and helping them in a creative way. Tools like ChatGPT, GitHub Copilot and DeepSeek are making major strides. These tools are trained on massive code repositories and using very advanced machine learning to detect patterns and generate logical and precise answers. Not only they are suggesting code, correcting the errors, finding bugs in the code but also writing the entire code blocks by having one single and very simple prompt instructions. Developers are now so much dependent on AI that in a sense it has changed their role from coder to supervisor. Most of the modern Integrated Development Environments (IDEs) are now including their own version of AI that assists with coding. As a result, developers and professionals now have a companion that saves their time, reduce repetitive tasks and beginners have their own personal teacher who is helping them from scratch.

While the unquestionable benefits that AI tools are bringing are changing the Software Industry for good, developers and institutions are not fully prepared for the ethical consequences of using these tools. One of the major concerns is Code plagiarism while reusing licenced or copyrighted code. The ownership of the code is also a valid problem. Though developers are being more efficient, they are also becoming over reliant on it, so much so that it is affecting their critical thinking. As it is very user friendly and easy to use these tools the risk of AI suggesting insecure and harmful or unethical code is also high. So, it is very clear that if there is no established ethical standard for this regard, the ethical consequences will be very bad in near future.

Seeing the potential that AI has there is no way that the use of AI is going to decrease. From the students of any educational institutions to professionals in IT fields are increasing their use of AI tools day by day. In their education student might use AI to complete their tasks or assignments though it was meant to be solved by their own capabilities. These boundaries and limitations must be set up. In industry over reliance on Ai can cause security risks, licensing violations even unaccountable code generation. Developers need ethical awareness in

these matters. So, this research topic is very relevant now to the Software Industries and educational institutions as it implies the need of guidance to ensure AI tools are used ethically and safely.

The purpose of this thesis is not to limit or oppose the use of AI but to examine their ethical implications in a logical and constructive manner. As these tools will be used increasingly day by day it is going to be difficult to measure the risks it is going to bring If the ethical concerns are overlooked. This thesis will highlight all the ethical concerns and suggest practical solutions of them including transparency in training AI with data, setting clear policies for using AI in educational institute and Software Industries. By this way the goal of using AI responsibly will be met.

As AI tools itself is changing its own functionality and the use of it is also changing dynamically and that is why this thesis focuses on understanding the effect AI is leaving behind human minds and will try to understand the shift that is occurring. It will help to uncover the deeper ethical implications tied to this shift from human generated code to AI generated code. By examining how AI tools are used in real-world contexts and how developers interact with them, this study strives to identify patterns that can inform ethical frameworks, professional guidelines, and educational approaches moving forward.

2 LITERATURE REVIEW

2.1 AI-Assisted Coding Tools

The integration of Artificial Intelligence into Software Engineering changed the way programmers approach problem solving. As the rise of powerful large language models is happening, AI assisted coding tools has become very usefully integrated in Software Engineering Industry. Among a lot of benefits these tools give real time suggestions, help debugging and help explain or understanding complex codes and by this way it is helping improving development speed and productivity.

GitHub Copilot

GitHub Copilot is an AI programming tool created by OpenAI. It is integrated with the Visual Studio IDE, and it helps developers with auto completing code, debugging errors, creating boilerplate and saving their time. After launch it is being used by almost every developer from beginner learners to senior developers. Here in FIGURE 1 the data shows the preferred user of GitHub Copilot.

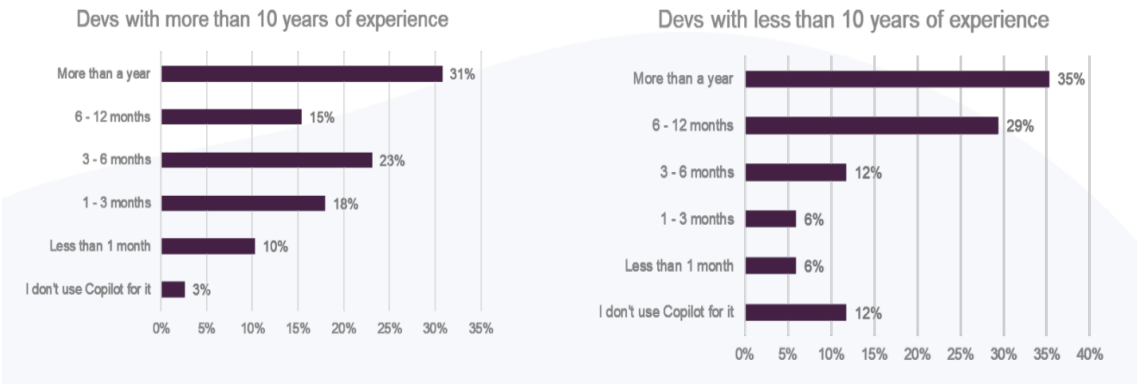

FIGURE 1. Data showing how many professionals prefer using GitHub-Copilot

ChatGPT

In summary, ChatGPT's rapid ascent and widespread adoption underscore its significant impact on how individuals and organizations approach information processing and task execution. Its multifaceted applications and proven benefits position it as a pivotal tool in the ongoing integration of AI into daily workflows. Designed as a conversational AI model, ChatGPT is capable of understanding and generating human-like text, making it versatile for various applications. Users employ ChatGPT for tasks ranging from drafting emails, writing essays, and generating code snippets to providing explanations on complex topics. Its ability to process and generate coherent and contextually relevant responses has made it a valuable tool in education, customer service, software development, and content creation. The advantages of using ChatGPT are supported by empirical data. In the FIGURE 2 bellow it clearly shows the increasing number of users that are using ChatGPT in their workspace.

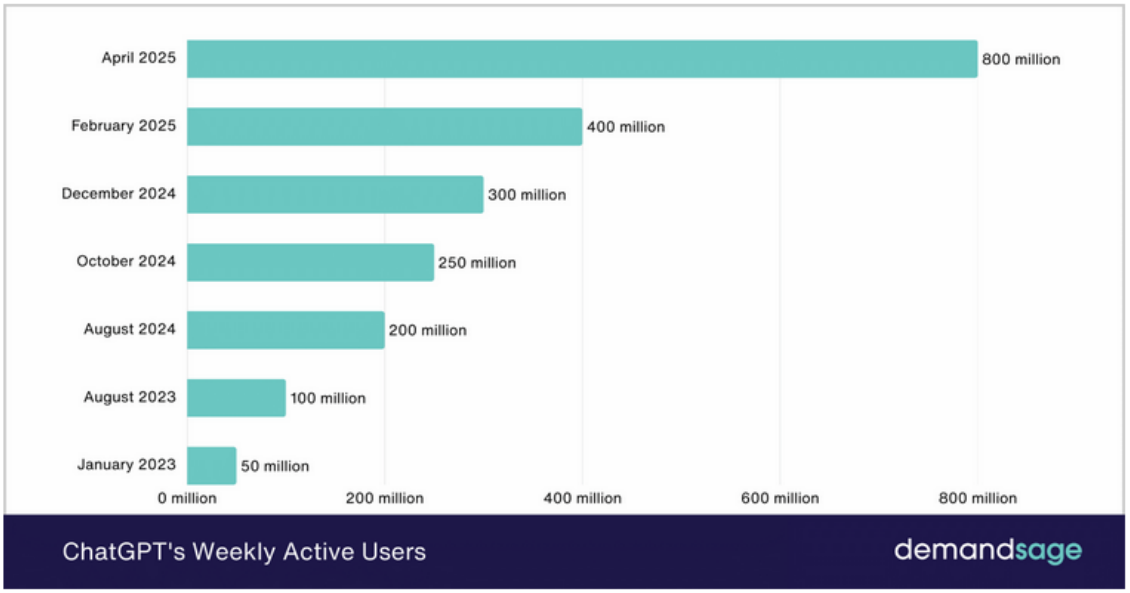

FIGURE 2. Increasing number of users according to Demandsage.

Many professionals and scholars in the tech field have voiced both enthusiasm and caution regarding the use of AI in coding. While some praise the significant time savings and creative support offered by AI assistants, others stress the importance of maintaining human oversight and critical thinking to ensure code quality and ethical use.

With AI increasingly handling repetitive and even complex coding tasks, the role of developers is gradually shifting. Rather than solely writing code, developers are now focusing more on directing, reviewing, and refining AI-generated outputs. This change is influencing not just workflow efficiency, but also how programming skills are learned and practiced.

2.2 Ethical Concerns in Prior Research

As artificial intelligence becomes a more common tool in software engineering, scholars and professionals have raised several ethical questions regarding its use, particularly in AI-assisted coding. Tools like GitHub Copilot, ChatGPT, and similar platforms are increasingly being used into software development. While they increase efficiency, reduce repetitive work, and support idea generation, they also create challenges that researchers have started to document and analyse. This section will provide the most frequently asked concerns in existing literature on the topic.

2.2.1 Plagiarism Risk

Maybe the most discussed or worrying ethical concern is the possibility of AI generating code that closely resembles or directly replicates existing work. Since these tools are often trained on large public repositories, they can reproduce sections of work that were originally done by others without proper citation. This can lead to developers unknowingly submitting or deploying plagiarized content, which raises issues in both academic and professional aspects

2.2.2 Unclear Code Ownership

Another important topic is who is the original author. When code is produced with the help of an AI tool, questions come around who has the rights to that code. The developer who typed the prompt, the organization that owns the AI tool, and even the authors of the training data could all potentially claim some level of ownership. Legal scholars have seen that current copyright laws do not really

address these complex scenarios, leaving developers with an unsolved question and concern on their mind.

2.2.3 Skill Degradation and Overdependence

AI tool makes coding and problem solving easier, but this advantage is turning into a disadvantage day by day as developers are now over dependent on AI tools as they are not using their own ideas a lot any more it is safe to say that using AI too much while being overdependent their skills are Degrading. Developers are now using automated codes from AI tools without even critical analysis or scrutiny. Over time this is affecting new learners as their basic concepts are not solid enough.

2.2.4 Security Vulnerabilities

AI-generated code may function properly on the surface, but it doesn't always maintain best practices for safety and security even when they are strictly told to do so. Several security audits of AI-assisted code have showed that these tools can introduce bugs, hard-coded credentials, and insecure APIs. Without careful review, such code could be integrated into sensitive systems that needs extra protection, putting users and data at risk.

2.2.5 Misuse and Malicious Applications

There is also the concern that people with bad intentions can exploit AI coding tools to generate harmful or malicious code. For example, someone could ask an AI to write harmful or phishing scripts. Some tools have filters to prevent this, but studies have shown they are not always effective. This creates ethical responsibility for both tool developers and users to prevent harmful use.

2.3 Ethical Theories in Context

To evaluate the ethical implications of AI-assisted coding, it is useful to apply traditional ethical theories that provide structured ways to assess right and wrong. These philosophical frameworks offer different lenses through which we can analyse the use of tools like ChatGPT and GitHub Copilot. The three key ethical theories commonly referenced in technology ethics are **utilitarianism**, **deontology**, and **virtue ethics**. Each theory helps to explore the moral dimensions of AI use in software development from a different angle.

2.3.1 Utilitarianism

Utilitarianism is a form of consequentialism that evaluates the morality of an action based on its outcomes. The central idea, as articulated by Jeremy Bentham and later expanded by John Stuart Mill, is that the best action is the one that produces the greatest amount of happiness or benefit for the greatest number of people. In the context of AI-assisted coding, this means we evaluate these tools by asking: **Do they improve overall productivity, reduce workload, and benefit the majority of developers and end users?**

For example, GitHub Copilot has been reported to boost developer productivity significantly. From a utilitarian point of view, such improvements could justify widespread adoption—*if* the positive outcomes outweigh the possible negative consequences such as plagiarism or security risks.

However, utilitarianism also forces us to weigh those downsides. If AI-generated code increases vulnerabilities or causes developers to lose key skills over time, these harms must be factored in. The theory encourages a balance: maximizing utility while minimizing harm.

2.3.2 Deontology

Deontology, founded by Immanuel Kant, emphasizes that actions must be judged by whether they follow a set of rules or duties, rather than by their consequences. In software development, this translates into questions like: **Is it inherently right or wrong to use code generated by an AI? Does it violate academic or professional codes of conduct, regardless of the benefits it brings?**

From a deontological standpoint, if AI-generated code includes copyrighted or plagiarized content, then using it would be ethically wrong—even if it saves time or increases efficiency. The act itself breaches a duty to uphold intellectual honesty and respect the original work of others. Similarly, deploying AI-generated code without thorough testing might violate a developer's responsibility to write secure and maintainable software.

Professional ethics codes, such as those from the ACM or IEEE, often reflect deontological principles by outlining developer responsibilities like honesty, fairness, and accountability. These codes imply that certain actions—like misrepresenting AI-generated work as original—are unethical regardless of outcomes.

2.3.3 Virtue Ethics

Virtue ethics, rooted in the philosophy of Aristotle, shifts the focus from actions or consequences to the **character of the person acting**. The main question becomes: **Does using AI tools help developers cultivate virtues such as responsibility, honesty, and excellence—or does it promote laziness, dependency, and dishonesty?**

For instance, when a student uses ChatGPT to complete a coding assignment without understanding the code, they might get the job done, but they aren't growing as a competent or honest programmer. This approach can erode technical craftsmanship and personal integrity. On the other hand, if a developer uses AI to enhance their learning, get suggestions, and explore new methods critically, then AI becomes a tool that supports intellectual growth.

Virtue ethics reminds us that ethical practice in software engineering is not just about rule-following or maximizing benefits. It's also about becoming a better, more responsible developer over time. In this sense, the ethical use of AI is deeply personal and contextual.

2.3.4 Justice Ethics (Rawlsian Theory)

John Rawls' theory of justice emphasizes fairness, equal opportunity, and the protection of the least advantaged. Applying this to AI-assisted software development, the ethical concern lies in whether AI tools like ChatGPT or GitHub Copilot—promote or undermine equitable access to opportunities. For example, if proprietary AI tools are only accessible to wealthier firms, this may deepen digital divides in the software industry. Moreover, if AI tools encode biases, they could generate unfair outcomes in systems that impact marginalized communities.

2.4 Topic Definition

This thesis focuses on exploring the ethical implications of using artificial intelligence tools in the process of software development, specifically AI-assisted coding. The central aim is to understand how tools like GitHub Copilot and ChatGPT are affecting the ethical responsibilities of software engineers. These AI tools are now being used widely for writing code, solving bugs, generating ideas, and even suggesting whole program structures. As their use grows, there is also a rising concern about their ethical impact.

The scope of this thesis includes key ethical areas such as **code ownership**, **bias in AI-generated suggestions**, **safety and security of the code**, and **the responsibility of developers when using AI tools**. It also considers **professional ethics** and how the role of a software engineer is changing in this new environment. For example, one important question is: Who is responsible if AI-generated code contains a bug that causes harm? Another concern is whether

AI tools are unintentionally promoting biased or unsafe coding practices, especially when their training data is not fully transparent.

This thesis does **not** focus on AI in general life, such as AI used in smart homes, personal assistants, or self-driving cars. It also does not cover the use of AI in creative design, digital art, or multimedia. Another area left out is low-code or no-code platforms, where users create applications with little or no programming knowledge. The focus stays on AI that supports or writes **code** in software engineering projects.

By narrowing the topic in this way, the thesis allows a deeper and more meaningful discussion of the ethical challenges faced by professional developers who use AI tools. The goal is not to criticize AI use but to promote awareness and encourage responsible use within software engineering.

2.5 Objectives and Purpose

The main goal of this thesis is to explore and understand the ethical concerns that come with using AI tools in software engineering. As more developers rely on AI-assisted coding platforms like GitHub Copilot and ChatGPT, it becomes important to think carefully about how these tools are changing the way software is developed. The objective is not just to describe these tools but to examine their deeper impact on professional values, responsibilities, and the development process itself.

One of the key objectives is to find out how software developers are currently using these AI tools in real work situations. This includes looking at whether they use AI mostly for suggestions, bug fixing, or even full code generation. Understanding this helps to set the foundation for identifying the ethical risks involved.

Another important goal is to analyse the possible risks and benefits that come with AI-assisted coding. For example, while these tools save time and boost productivity, they also raise serious questions about ownership of code, security

flaws, and the fading role of human judgment in decision-making. It is also necessary to ask whether using AI tools makes developers too dependent and weakens their original coding skills over time.

This thesis also aims to suggest best practices and ethical guidelines for developers and companies that use these tools. These suggestions will be based on existing research, case studies, and ethical theories. The goal is to offer ideas that support ethical and responsible use of AI in real-life coding work.

Finally, it is important to note that the purpose of this research is not to criticize or stop the use of AI. Instead, it is to encourage smarter, safer, and more thoughtful use of these tools. If used correctly, AI can help improve the work of software engineers without replacing their judgment or creativity.

3 AI Tools in Practice – Benefits, Risks, and Ethical Considerations

3.1 Introduction to The AI Tools

Artificial Intelligence has come up as very useful and powerful tool in modern software development. Using GitHub Copilot and ChatGPT developers are now integrating AI into their majority of work starting from writing code to debugging testing and a lot other. In this evolutionary time of AI this technical shift in the software industry is not only technical but also ethical.

Among the most widely adopted AI tools in programming are ChatGPT and GitHub Copilot. ChatGPT supports a variety of development tasks, from refining algorithms to generating code snippets on demand. GitHub Copilot, developed in collaboration with OpenAI, assists developers by integrating directly into Visual Studio and other IDEs. These tools are increasingly being used to enhance coding efficiency, streamline workflows, and drive significant changes across the software development industry. As the FIGURE 3 implies the increasing amount of usage and the impact it is having on software industry.

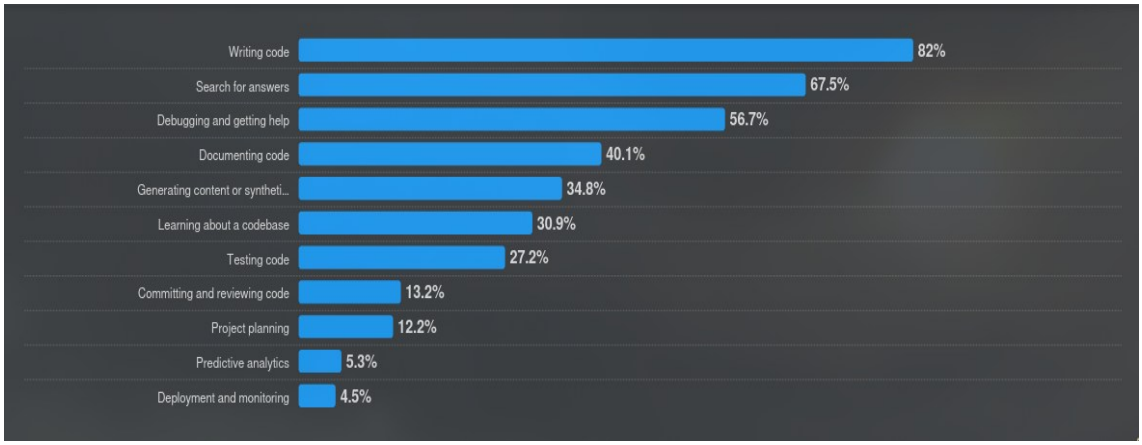

FIGURE 3. From the 2024 stack overflow developers survey showcasing the impact of AI in the development workflow.

The benefits of AI and AI assisted tools are widely monitored and studied but in this changing of time being very concern about the ethical use is also important. Developers are not only using AI for productivity they are literally adapting to it,

changing their workflow, being over-dependent to it relying on key decision-making responsibility. As a result, this shift is raising the ethical concerns even more. By looking into how these tools are being used, in which cases the developers and the users are making this shift – we will have a clear view of the danger of the ethical misuse that is happening and where the potential risks are. The ethical issues are also shown in the bellow FIGURE 4.

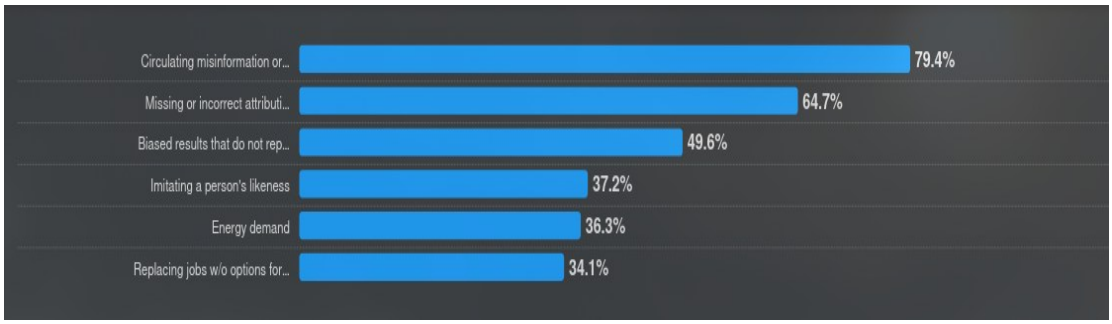

FIGURE 4. From the 2024 stack overflow developers survey for the ethical issues for AI assisted tools.

This section will explore the practical implications of AI-assisted development by examining the benefits, limitations, and ethical concerns tied to tools like ChatGPT and GitHub Copilot. Through real-world examples, tool comparisons, and referenced data, it aims to highlight how developers are using these tools and what responsibilities and challenges come with them.

3.2 Advantages of Using AI-assisted Coding Tools

AI tools have significantly changed developers work in software industry. From coding to debugging, testing and many other areas. Junior developers now have their personal helper that they never had before. From researching to thinking and analysing in every area AI have helped developers and personal in software industry so much that now they became over dependent on using AI in many cases. While using AI can guide the rise of next revolution in the development of Software it is also more relevant now than ever to be very cautious about the ethical use of AI. The advantages and the benefits of using AI assisted tools will be highlighted in this section. In the bellow FIGURE 5, it is clear that AI is impacting on various sectors of developer's workflow.

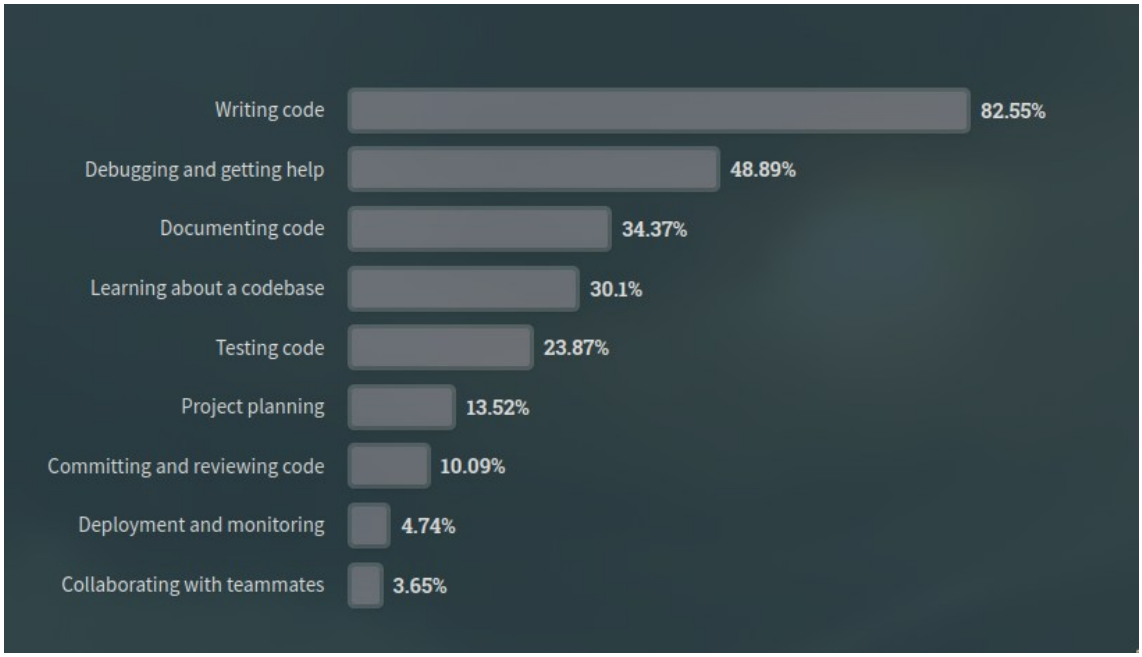

FIGURE 5. From the 2023 Stack overflow developers survey for the benefits of using AI

Here is another survey showing different areas of using AI by developers from the Stack overflow survey 2023 clearly shown in FIGURE 6.

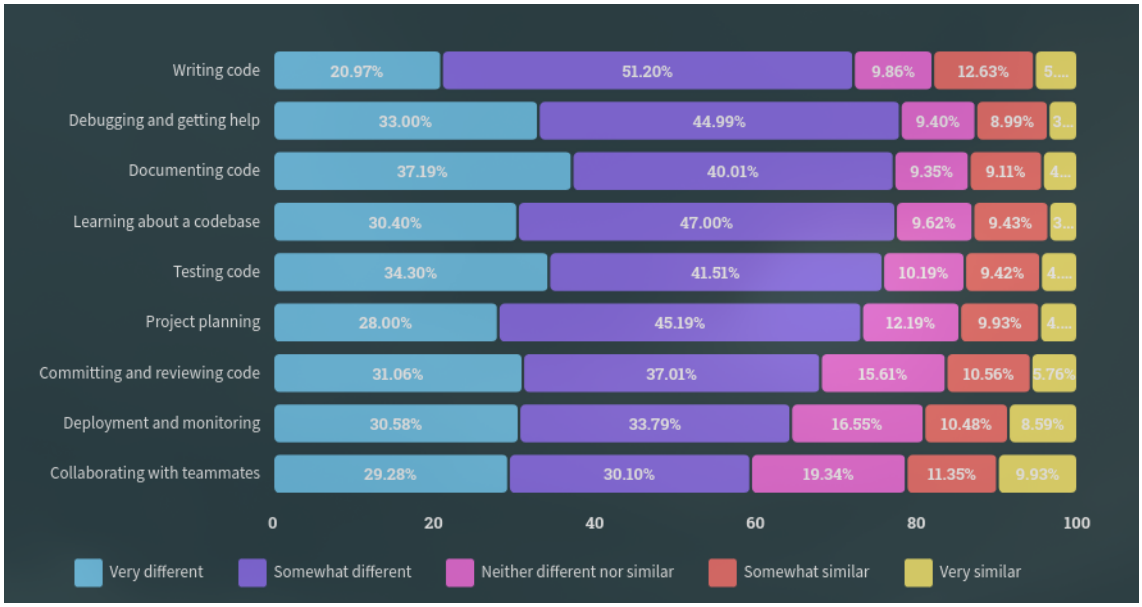

FIGURE 6. AI changing development workflow by helping in these sectors.

3.2.1 Increase in Productivity

AI-assisted tools like ChatGPT and GitHub Copilot have contributed significantly to enhancing developer productivity. These tools help automate routine coding tasks, provide real-time suggestions, and assist with problem-solving directly within development environments. By streamlining workflows and reducing the need for manual coding, developers are able to complete tasks more efficiently and focus on higher-level challenges. As a result, teams are experiencing faster development cycles and greater overall efficiency.

The bellow FIGURE 7 showing that the percentage of people impacted by its positive productive benefits.

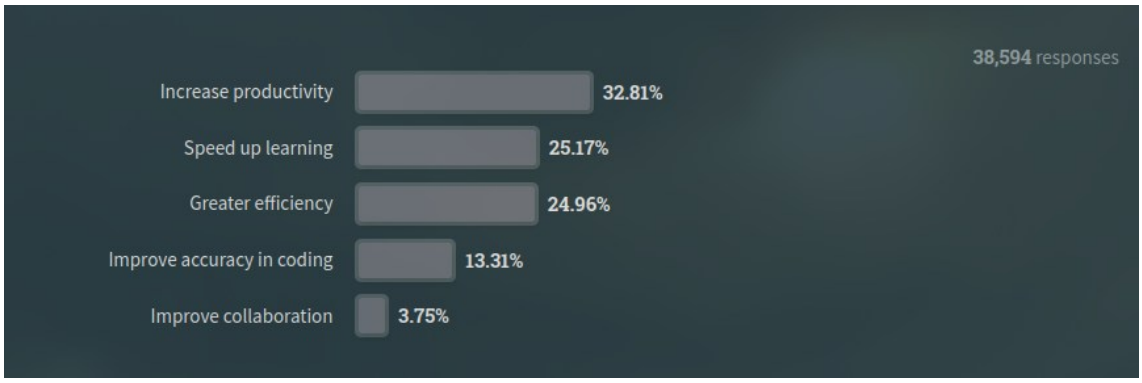

FIGURE 7. From the 2023 Stack overflow developers survey showing highest amount of people appreciating for the increased productivity.

3.2.2 Reduction in Repetitive Tasks

AI-assisted development tools have proven effective at automating repetitive programming tasks. Developers now rely on these tools to quickly generate standard code patterns such as getter and setter methods, boilerplate React components, and configuration files. What once required significant manual effort can now often be accomplished in a fraction of the time with minimal input. However, while these tools offer substantial time savings, they also introduce the need for careful review. Generated code may include unnecessary dependencies or minor inaccuracies that require correction before integration into production systems. Therefore, while automation reduces upfront effort, it still demands

thoughtful oversight to maintain code quality and prevent long-term technical debt. Bellow FIGURE 8 mentions the time saving benefit of AI tools.

Task	Manual Time	AI Time	Revision Needed
Unit Test Creation	45 mins	10 mins	25% of cases
Error Handling	60 mins	15 mins	32% of cases
API Boilerplate	120 mins	25 mins	18% of cases

FIGURE 8. Time savings vs. required revisions for AI-automated tasks (Synthesized from JetBrains 2023, IEEE TSE 2023, Google 2023).

3.2.3 AI's Dual Role in Junior Developer Education

New programmers often use AI tools like ChatGPT and GitHub Copilot as helpful assistants while learning to code. These tools provide quick explanations and can help with debugging, offering support that is available anytime. This instant guidance can make learning smoother and more efficient. However, relying too much on AI tools can sometimes prevent learners from fully developing their own problem-solving and coding skills. It may also reduce their ability to remember basic coding concepts or handle new challenges independently. Because of this, educators emphasize the importance of balancing AI assistance with practicing foundational skills without aid.

AI tools provide real-time explanations and support that often surpass what traditional textbooks can offer. They help clarify programming concepts quickly, assist in debugging by identifying and fixing simple errors more efficiently, and are available around the clock, making them especially valuable for learners working at unconventional hours.

There is a growing concern that overreliance on AI tools can lead to gaps in essential programming skills. Some learners may find it difficult to recall or write basic code independently after depending heavily on AI assistance. Additionally, those using AI support sometimes take longer to troubleshoot new or unfamiliar

problems on their own. Issues with originality have also been observed, as AI-generated code can blur the lines of academic honesty. Because of these challenges, some educators have introduced assessments that limit or prohibit AI use to ensure students develop their fundamental coding abilities without external aid.

3.3 Risks and Limitations of AI-Assisted Development

3.3.1 Over Reliance and Skill Decreasing

While AI has had a significant impact across many areas, its widespread use also raises important concerns about long-term effects on skill development. The increasing accessibility and integration of AI tools in software development have led to a growing dependence on these technologies. This reliance may contribute to a decrease in fundamental coding abilities and problem-solving skills as developers lean more on AI assistance rather than building expertise independently.

The integration of AI into software development workflows is transforming how developers work, bringing both opportunities and challenges. While AI tools like code generators and assistants can boost productivity by automating routine tasks, they also risk creating gaps in developers' foundational coding skills. Many developers are still adapting to working alongside AI, which can lead to overreliance on these tools and potential decreases in problem-solving abilities and code comprehension. Balancing AI assistance with continued skill development is essential to ensure long-term proficiency and innovation in software engineering (Janssen, 2024). This is similar to the "calculator dependency" observed in mathematics education during the 1980s.

While AI-powered dialogue systems offer students quick support and personalized learning experiences, excessive dependence on them may actually harm long-term learning outcomes. According to a systematic review by Zhai, Wibowo, and Li (2024), students who frequently rely on AI tools tend to show

reduced engagement in critical thinking and independent problem-solving. The ease of receiving instant answers often discourages learners from actively analysing concepts or seeking deeper understanding, which in turn weakens cognitive development and decision-making skills. This trend suggests that unbalanced AI use in education could shift learning from active exploration to passive consumption.

3.3.2 Code Quality Issue

A recent study by GitClear looked at over 153 million lines of code and found some worrying trends linked to AI coding tools. The research showed that AI often creates a lot of copy-pasted code instead of code that fits well with the rest of the project. This can cause problems for keeping the software stable and easy to maintain over time. AI also seems to increase technical debt — meaning quick fixes now could cause more work later. This “AI-induced technical debt” means teams need to be careful, use better testing, and find new ways to measure developer productivity beyond just how much code they write. Even though AI can help developers work faster, it’s important to use these tools wisely, with clear guidelines and good training, to keep code quality high (DevOps.com, 2025).

3.3.3 Code Ownership Problems

AI tools often generate code by copying from existing sources, which can lead to challenges in maintaining the software. Since these tools rely on analysing data from various places, the risk of unintentionally using code that may have ownership or legal concerns is significant. It’s important to carefully manage and review the data used to train AI systems to avoid potential problems with intellectual property rights down the line.

3.4 Ethical Implication of Using AI

3.4.1 Plagiarism

Recent investigations reveal troubling patterns about code plagiarism:

Key Findings
During my Internship at BizBondIT Limited we noticed Copilot sometimes regurgitated code from Stack Overflow posts which is a clear violation of plagiarism.

Real-World Example
When building a REST API last semester, Copilot suggested JWT code that exactly matched a 2018 GitHub gist (later confirmed via Codequiry scan). This forced our team to:

1. Rewrite the entire authentication layer
2. Implement mandatory similarity checks
3. Add /*AI-REVIEWED*/ tags to all suspect code

3.4.2 Bias in Suggestion

ChatGPT and other AI assisted tools are very much biased as they show specific traits which reveal the biased pattern.

Shocking Numbers

- Python suggestions outnumbered Rust 3**:1** in my personal log of 50 Copilot prompts
- A recent study by Huang, Zhu, Xing, Jin, Wang, and Xu (2023) introduced a novel AI-based approach called PCR-Chain to resolve fully qualified names (FQNs) and fix syntax errors in partial code snippets. This method leverages large language models to outperform previous state-of-the-art techniques by approximately 5%, achieving over 80% accuracy,

particularly in Python. The study highlights the potential of prompt-based AI solutions in improving software engineering processes without relying on traditional program analysis.

3.4.3 Security and Privacy Risks

AI-powered coding tools, while increasing productivity, introduce serious security vulnerabilities and privacy issues. A lot of documented cases and studies highlight these risks, making it crucial for developers to exercise caution.

Exposure of Sensitive Data

AI tools can leak confidential information time and time again which makes it very vulnerable for the developers and for any purpose of use.

- **Samsung's ChatGPT Leak (2023):** Engineers pasted proprietary semiconductor code into ChatGPT for debugging. The code was stored on OpenAI's servers, violating Samsung's internal policies. For all this reason Samsung banned AI tools for sensitive projects (Ray, 2023).
- **Copilot leaking data:** GitHub Copilot has been found to occasionally expose sensitive data, including active API keys and credentials, by suggesting snippets copied from public code repositories. This leakage poses significant security risks, especially for cloud services, as about 12% of cloud-related prompts returned high-risk secrets (GitGuardian, 2023)

3.5 Summary and Reflection

3.5.1 Key Takeaways

As I lookback to the investigation there is three important points that I noted both from the data and the developer interview

1. Productivity Paradox

    - Yes, AI saves a lot of time specially in the boilerplate task

- On the other hand, we pay for using it in the debugging time as shown in the interview.

2. Skill decreasing

    - Yes, AI helps beginners and even experts on their coding journey immensely
    - We also see that they become overdependent of using it and their skills decrease day by day

3. Security Issues

    - The reason we use AI is to make something usable and secure though it helps initially to make it happen
    - The security issues by AI generated code is worrisome

3.5.2 Reflection

Looking back to everything it is clear that AI have both very good and also some concerning side effects If it is misused. So many of time saving tasks were done by AI tools that now developers cannot think any other alternative of it for the same amount of productivity but at the same time it is also noticed that the ethical concerning issue of AI is also there. While GitHub Copilot auto fills boring database code, on the other hand it is also creating problems when suggesting others licensed code and the coder have no idea about it later being in a grave problematic situation westing a lot of precious time. For this reason, developers still must check everything like a teacher evaluating homework after even finishing the project doesn't matter how greatly it is working. One of the other major issues is the rules regarding AI tools is not yet that much clear which we showed by several cases before. For this reason, it is better for companies and other using AI implementing better policies regarding AI usages and the legal parameters.

Finally, as a tool AI is not going to be avoided just for these issues as the positive influence of AI is so much. What is needed is the right implementation of using it, being very cautious of the ethical implementations and implications of using it. As time goes by more and more advanced ways of handling it should be arise and then it will be less of the issue but for now, we should be as much careful about the issues mentioned and many more.

4 DISCUSSION

AI tools like GitHub Copilot and ChatGPT offer undeniable productivity gains automating repetitive tasks, reducing development time, and assisting junior programmers. However, these benefits come with ethical issues, including skill decrease (over-reliance leading to weaker problem-solving abilities). Without proper safeguards, widespread AI adoption risks normalizing code plagiarism, security vulnerabilities, and biased outputs—issues already observed in real-world cases like Samsung's ChatGPT data leak and Stack Overflow's temporary ban on AI-generated answers.

Moving forward, the software industry must balance innovation with responsibility. Developers should treat AI as a helper not a replacement by rigorously reviewing suggestions, documenting AI use, and staying accountable for final code quality. Meanwhile, organizations need clear policies on AI-generated code ownership, security audits, and ethical training. Without these measures, AI risks doing more harm than good in software engineering.

REFERENCES

DevOps.com. 2025. AI in software development: Productivity at the cost of code quality? https://devops.com/ai-in-software-development-productivity-at-the-cost-of-code-quality/

GitGuardian. 2023. Yes, GitHub Copilot can leak secrets. *GitGuardian Blog*. https://blog.gitguardian.com/yes-github-copilot-can-leak-secrets

Huang, Q., Zhu, J., Xing, Z., Jin, H., Wang, C., & Xu, X. 2023. A chain of AI-based solutions for resolving FQNs and fixing syntax errors in partial code. *arXiv*. https://arxiv.org/abs/2306.11981

Janssen, J. 2024. The disruptive impact of AI on software development. *LinkedIn*. https://www.linkedin.com/pulse/disruptive-impact-ai-software-development-jackie-janssen-cf7ke

Ray, S. 2023. Samsung bans ChatGPT and other chatbots for employees after sensitive code leak. *Forbes*. https://www.forbes.com/sites/siladityaray/2023/05/02/samsung-bans-chatgpt-and-other-chatbots-for-employees-after-sensitive-code-leak/

Zhai, C., Wibowo, S., & Li, L. D. 2024. The effects of over-reliance on AI dialogue systems on students' cognitive abilities: A systematic review. *Smart Learning Environments*, 11(28). https://slejournal.springeropen.com/articles/10.1186/s40561-024-00316-7