

The impact of artificial intelligence on software developers and software development companies

Merjem Buza, Ahmed Smaka

Faculty of Traffic and Communications, University of Sarajevo, Zmaja od Bosne 8, 71000 Sarajevo, Bosnia and Herzegovina

Abstract

Almost every few decades, an innovation emerges that completely changes the world, playing a vital role in raising the standard of living, such as the internet. What will be the next definitive moment in history? It's here, and it is artificial intelligence (AI). In this paper, we will discuss how AI technology can enhance programming. We will analyze the current use-cases of AI in this sector and explore possible future applications. We will look at how AI can help developers be more efficient and effective. We have addressed the impact of some of the most well-known software tools available to developers, such as ChatGPT, Devon, Cody, GitHub Copilot, and others. In this study, we have investigated the percentage of developers utilizing artificial intelligence. Furthermore, we have explored whether they employ it solely for business purposes or for day-to-day work as well. All of these aspects are depicted through tables, graphs, and images. The introduction of artificial intelligence (AI) brings both positive and negative aspects. Companies should establish governance standards for using AI tools to ensure that they are used ethically and effectively.

Keywords: Artificial intelligence, programming enhancement, developer efficiency, AI tools in software development.

1 Introduction

In today's rapidly evolving tech landscape, the rise of artificial intelligence (AI) has sparked curiosity and concern among students and aspiring software engineers. As AI and machine learning continue to advance, many wonder: will AI replace software engineers? Before delving into this main question, it's crucial to understand what AI is. Artificial intelligence refers to the simulation of human intelligence processes by machines, particularly computer systems. This encompasses learning, reasoning, problem-solving, perception, and language understanding, which AI achieves through various techniques such as machine learning and deep learning[1].

Introducing innovative tools like ChatGPT and Copilot has created new opportunities to assist and augment software developers across various problems. This paper studies the productivity

effects of AI tools on software development. Furthermore, in this study, we have delineated

both the positive and negative aspects of artificial intelligence, its utilization within software development companies, and the adoption of AI by software developers. We conducted research to ascertain whether they utilize AI tools in their professional environment or solely in day-to-day operations, which tools are most commonly employed by software developers, to what extent, and how much it facilitates their work as a result. With its ability to comprehend context, intention, sentiment, and myriad other elements, AI enables users of all age groups and backgrounds to engage in natural communication across various languages, even without prior knowledge of programming or computer science. It finds application in numerous domains such as customer support, entertainment, education, finance, healthcare, and more[2]. The open-source nature of AI facilitates easy customization, providing developers with flexibility in creating

Corresponding author: Ahmed Smaka (ahmed.smaka@fsk.unsa.ba)

© 2024 The Author(s). This work is licensed under a Creative Commons Attribution 4.0 International License.

applications harnessing its powerful capabilities. However, responsible utilization of these technologies is paramount to mitigate potential harm, whether direct or indirect. This consideration is crucial for maximizing success in integrating advanced technologies into society in the future. The introduction of Generative Artificial Intelligence (GenAI) - a type of artificial intelligence, has brought about rapid changes in software development processes and tools. Furthermore, GenAI has the potential to revolutionize software development by automating repetitive tasks, improving code quality, enhancing collaboration, providing data-driven insights, and ultimately accelerating the development cycle. However, merely possessing good tools is not sufficient[1]. For a company to succeed in software product development, well-functioning teams and efficient processes are crucial. Software engineering is a social activity that emphasizes close cooperation and collaboration among all team members and across teams within the organization. Therefore, it is important to note that while AI offers significant potential, it also presents challenges such as ethical considerations, concerns about data privacy, the need for qualified professionals to manage technology within software teams, and potential changes in team dynamics. However, research on team dynamics is lacking. To understand the impacts on team dynamics, it is also necessary to consider individual work practices to comprehend the effects at the team level. This paper investigates how the work practices of software engineers have transformed and the potential impacts of these transformations on collaboration[1].

2 Comparison of the past and present approaches

Before the advent of artificial intelligence (AI), software developers had to manually write every line of code required for the development of software applications or systems. This typically involved detailed planning, designing, and implementing each function and component of the software. In our research on this topic, we delineated the software development phases as follows:

Manual Planning and Design: Before any code was written, software developers spent a lot of time planning and designing the software. This included defining functionalities, system architectures, databases, and user interfaces, often requiring detailed technical specifications.

Manual Programming: After requirements and architecture were defined, developers manually wrote code for each software functionality. This involved using text editors or integrated development environments (IDEs) to write lines of code in programming languages such as C, C++, Java, Python, etc. Every aspect of functionality, from user interfaces to business logic and databases, required manual programming.

Testing: After the code was written, the testing process followed. Testing involved manually writing tests and verifying the functionality and performance of the software. This often required detailed manual testing of different scenarios to ensure that the software functioned as expected.

Maintenance and Optimization: After the software was released, developers had to maintain and optimize the code to ensure its reliability and performance. This involved fixing bugs, implementing new features, and optimizing the code for faster execution.

Manual Data Analysis: If the software involved data analysis, it often required manual extraction, transformation, and analysis of data without the use of advanced machine learning or data analysis techniques.

Before AI, software development was a complex process that required a lot of manual work and a deep understanding of programming and computer science. Each step of the process, from planning to testing and maintenance, required significant human effort and time. AI has changed this landscape by providing tools and techniques that help automate certain tasks and speed up the software development process. With the use of AI tools, the phases of software development have undergone significant changes and improvements[3]. Here's how these phases look today with the application of AI technologies:

Planning and Design: AI code generation tools can assist in the early planning stage by automatically generating basic software components based on predefined requirements. Data analysis tools can aid in gathering and analyzing data on user preferences and needs, facilitating the definition of software specifications. AI techniques, such as natural language processing (NLP), can analyze communication between teams and stakeholders to identify key requirements and user needs.

Programming: Generative models can automatically generate code for certain parts of the software, such as data processing routines or user interfaces. AI code analysis tools can automatically identify and correct code errors and suggest optimizations, speeding up the programming process and improving code quality.

Testing: AI testing tools can automatically generate test cases based on code analysis and software specifications. They can also automate test execution and analyze results. Machine learning techniques can be used to predict potential software errors before they occur in the real world.

Maintenance and Optimization: AI tools can monitor software performance in real-time and automatically detect and resolve issues to ensure reliability and performance. Generative models can automatically suggest code optimizations to improve execution speed or reduce resource consumption.

Data Analysis: Advanced data analysis and machine learning techniques can uncover patterns and trends in large datasets, enabling a better understanding of user needs and behaviors[4].

With AI tools, the software development phases have become more efficient, faster, and less prone to human errors. AI technologies enable the automation of many tasks that previously required a lot of manual work, freeing up time and resources to focus on creative and complex aspects of software development. Furthermore, a question arises from the perspective of software developers: "Will I be replaced?" Given the current limitations of generative AI, it is not capable of replacing software engineers. However, it could potentially diminish the availability of software engineering positions. For instance, as reported by the Wall Street Journal,

Jonathan Burket, a senior engineering manager at Duolingo Inc., noted a 25% increase in his efficiency since adopting Copilot. This is attributed to the elimination of tasks such as code documentation, test case creation, and information lookup. Present AI tools lack a comprehensive understanding of abstract concepts, contextual comprehension, and the subtleties inherent in human language and requirements. Moreover, they lack the capacity for ethical decision-making and consideration of the societal ramifications of software implementations, as well as the ability to innovate in the manner humans do. Hence, while generative AI can serve as a potent instrument in a software engineer's arsenal, aiding in task automation and enhancing productivity, it is unlikely to supplant software engineers entirely. The replacement of software engineers necessitates artificial intelligence endowed with creativity and problem-solving capabilities on par with or exceeding those of humans. In essence, the advent of artificial general intelligence (AGI) would herald the potential for such replacements. According to IBM, AGI refers to "intelligent machines that are indistinguishable from the human mind." [5]

3 The implications for employees and the company

While generative AI holds immense potential, it also presents significant challenges. One of the primary challenges is the quality and reliability of the generated content. AI models can sometimes produce nonsensical or inappropriate outputs because they lack the deep understanding of context and nuance that humans possess.[6] Another challenge is the ethical and legal implications of AI-generated content. Issues related to copyright infringement, misinformation, and deepfakes are emerging with the rise of generative AI. From a security standpoint, sharing code with artificial intelligence can lead to unintended consequences for software development companies. For instance, if proprietary code belonging to the company is shared with an AI tool, it may result in code leakage. This occurs when the AI tool stores the code submitted for verification or refinement and subsequently offers it as a solution to future queries. Since this code constitutes the

intellectual property of the company, its dissemination in this manner could make innovations accessible to a broader audience. To prevent this, software companies should establish guidelines for the use of artificial intelligence and define the extent to which developers can utilize it. In addition to setting clear guidelines, companies should also invest in educating their employees about the potential risks and ethical considerations associated with generative AI. Training programs can help developers understand the importance of data privacy, the implications of sharing proprietary information, and the ethical use of AI technologies. By fostering a culture of awareness and responsibility, organizations can ensure that their teams are better equipped to navigate the complexities of AI integration.

4 Usage of AI Tools presented through tables

We conducted a survey among software developers to gauge their usage and perceptions of AI tools in the workplace. Through an interview-based approach, we gathered insightful data on the prevalence of AI tool adoption, specific tools utilized, and the perceived ease of their integration into daily workflows. The following tables present a comprehensive overview of our findings.

Table 1: Usage of AI Tools at Work

Usage of AI Tools at work	Percentage(%)
Yes	96.55
No	3.45

Description: This table illustrates the percentage of individuals who use AI tools at their workplace.

Table 2: Usage of Specific AI Tools

AI Tool	Percentage
Chat GPT	65.52
GitHub Copilot	31.03
Cody	31.03
Devon	20.69
Code Guru	17.24

Description:

This table displays the percentage of individuals using specific AI tools at work.

Table 3: Rating of Ease of AI Tool Usage

Scale (1-5)	Column A (t)
1	0.00
2	6.90
3	27.59
4	34.48
5	24.14

Description:

This table shows the distribution of ratings on the ease of using AI tools, rated on a scale from 1 to 5.

Based on the tables presented, we can draw the following conclusions:

1. High Adoption Rate: The data reveals a significant adoption rate of AI tools among software developers, with 96.55% of respondents indicating their use of such tools in their work environments.
2. Popular Tools: Among the AI tools surveyed, Chat GPT emerges as the most widely utilized, with a usage rate of 65.52%, followed by GitHub Copilot and Cody, each with a usage rate of 31.03%.
3. Mixed Perceptions on Ease of Use: While a majority of respondents utilize AI tools, there is a variance in perceptions regarding the ease of integrating these tools into daily workflows. While 34.48% of respondents find them moderately beneficial (rated 4 on a scale of 1-5), 27.59% rate their usefulness as average (rated 3). However, a notable proportion (24.14%) rates them highly beneficial (rated 5), indicating a positive outlook on their impact.

Overall, the findings suggest a widespread acceptance and utilization of AI tools in the software development domain, with room for further exploration and optimization in enhancing their seamless integration and effectiveness in various workflows.

5 Conclusion

This paper presents evidence on the productivity effects of generative AI tools in software development. To the best of our knowledge, it is the first controlled experiment to measure the productivity of AI tools in professional software development. In this paper, we describe the state of programming before the advent of artificial intelligence and compare it to contemporary practices. We conducted a survey within a software company, targeting software developers, and the results revealed that over 95% of them utilize AI tools both in their professional tasks and daily work. These tools significantly ease their workload. Furthermore, we demonstrate the impact of artificial intelligence on companies and their employees. At present, using AI tools primarily facilitates the research process by aiding in the search for potential problem solutions. Additionally, these tools are employed to propose solutions for minor issues and to assist in the learning process. These AI tools are perceived to have substantial potential in the work environment. However, it is acknowledged that they will not be able to fully replace individuals with specific work experience. Moreover, our study highlights the transformative effects of AI on workplace efficiency and productivity. The integration of AI tools has led to faster problem-solving and more efficient handling of routine tasks. Despite these advancements, the need for human expertise remains paramount, particularly in complex and nuanced situations where experience and intuition play a crucial role. Looking forward, it is essential for companies to strike a balance between leveraging AI tools and maintaining a skilled workforce. Continuous training and upskilling of employees will ensure that they can effectively collaborate with AI technologies, thereby maximizing the benefits of both human intelligence and artificial intelligence. In conclusion, while AI tools provide significant advantages and facilitate many aspects of software development and other professional activities, they are not a substitute for the invaluable insights and expertise that experienced professionals bring to their work. The future of work will likely see a harmonious integration of AI tools and human skills, leading to more innovative and efficient outcomes.

6 References

- [1] R. Ulfsnes and N. B. Moe, "Transforming Software Development with Generative AI: Empirical Insights on Collaboration and Workflow," Researchgate.Net, no. February, pp. 0–15, 2024, [Online].

Available:

https://www.researchgate.net/profile/Rasmus-Ulfsnes/publication/378567262_Transforming_Software_Development_with_Generative_AI_Empirical_Insights_on_Collaboration_and_Workflow/links/65e0541e7670d36abe62051/Transforming-Software-Development-with-Generativ

- [2] A. Bucaioni, H. Ekedahl, V. Helander, and P. T. Nguyen, "Programming with ChatGPT: How far can we go?," Mach. Learn. with Appl., vol. 15, no. December 2023, p. 100526, 2024, doi: 10.1016/j.mlwa.2024.100526.
- [3] S. Chinthapatla, "Unleashing the Future : A Deep Dive into AI-Enhanced Productivity for Developers," no. March, 2024.
- [4] "6 Impacts of Artificial Intelligence on Software Development." Accessed: May 26, 2024. [Online].

Available:

<https://www.intelivita.com/blog/impact-of-ai-on-software-development/>

- [5] "Will AI Replace Software Developers? | IT Career Center | CompTIA." Accessed: May 27, 2024. [Online].

Available: <https://www.comptia.org/blog/will-ai-replace-software-developers>

- [6] "Breaking down the advantages and disadvantages of artificial intelligence - IBM Blog." Accessed: May 26, 2024. [Online].

Available: <https://www.ibm.com/blog/breaking-down-the-advantages-and-disadvantages-of-artificial-intelligence/>