American Journal of Engineering Research (AJER)
e-ISSN: 2320-0847 p-ISSN: 2320-0936
Volume-14, Issue-7, pp-30-33
www.ajer.org
Research Paper Open Access

A Review on Using Artificial Intelligence in Software Development

Anish Bhandi
[1](Freedom High School, California, USA)

ABSTRACT : *Artificial intelligence (AI) is becoming an essential part of modern software engineering, changing how software is designed, developed, and maintained. The growing use of AI-driven tools—such as intelligent code assistants, automated testing platforms, and AI-enhanced development—is speeding up development cycles and improving software quality. This paper reviews how AI is currently applied throughout the software development lifecycle, focusing on areas like code generation, testing automation, predictive analytics, and low-code/no-code development. While these advancements offer clear benefits, including improved efficiency and reduced manual effort, they also present new challenges, such as potential security risks, ethical concerns, and the need for human oversight. This paper intends to examine the balance between developer responsibility and usage of AI systems, highlighting the importance of ethical AI use in software engineering. This paper concludes by exploring strategies for integrating AI into software workflows responsibly and effectively.*
KEYWORDS *Artificial Intelligence, Software Development Cycle, Software Engineering, AI-driven models.*

Date of Submission: 02-07-2025 Date of acceptance: 12-07-2025

I. INTRODUCTION

The software engineering field has seen rapid changes in recent years as new tools and methodologies have been developed to improve the speed, quality, and efficiency of software production. Traditional methods, such as the waterfall model and even agile practices, while useful, have not fully solved common challenges like long development cycles, manual coding errors, and resource-heavy testing processes. As software demands continue to grow, the industry is looking toward artificial intelligence (AI) to offer more advanced solutions. AI is currently a powerful force in the field of software development because of its ability to aid in every task, from code writing to the deployment of applications. Specific technologies like machine learning, natural language processing, and large language models (LLMs) are being used to further the automation of models and optimize certain parts of the software development process. For instance, one such case is GitHub Copilot, which uses AI to provide code in real-time, which helps the developers write faster and more efficiently. AI is widely being used in automated testing tools, which can generate and prioritize test cases, reducing the time required for manual testing and improving software reliability.

AI is influencing the development process by improving the performance of DevOps pipelines and allowing for the proper maintenance of the pipelines, where AI models can correctly predict possible system failures before they occur. Low-code and no-code platforms, which are supported by AI, are bringing the barrier down for entry into software creation, which allows users with limited or no programming background to build a career in software engineering and inspire future generations. All these advancements pose unwanted risks. There is a major dependence on automation owing to AI which results in model bias, security gaps, and loss of developer creativity due to redundant reliance on automation stifling critical thinking. Also, while software development becomes more automated, issues such as lack of transparency and accountability due to ethical ramifications of code produced by AI need scrutiny.

This paper will explore how AI is currently changing the software engineering practices. The second section of this paper analyze the benefits of AI in software development. The third section discusses the challenges or limitations of incorporating AI into software development processes. Finally, provide future discussion points

on how developers and organizations can safely recruit AI online tools using human oversight without compromising the usability of AI in the fourth section. The fifth section discusses the conclusion.

II. BENEFITS OF AI IN SOFTWARE ENGINEERING SYSTEM

The implementation of Artificial Intelligence (AI) in software engineering reveals multiple advantages which transform the current software development process. The software development cycle receives benefits that enhance both efficiency and quality for software engineer developers throughout its various stages. The implementation of these advancements brings forth their own set of risks.

A. The Increase of Productivity

AI-powered tools are used to reduce the time needed for coding, testing, and deploying the software. The automated code generating tools, such as the GitHub Copilot, allows the developers to complete the coding faster by the suggested coding snippets based on the context. This will speed up the entire development processing. This allows the teams to be focused on resolving the complex problems instead of spending the time on repetition of the tasks.

B. The Enhancement of Code Quality

AI detects the software defects and errors in coding at the initial stage of the software development phase by using intelligent prediction models. These tools are used to examine the historical data to make the prediction whether the bugs will appear in future that enables the developers to correct the issues before they escalate. AI contributes to the development of stable and reliable software by minimizing human errors.

C. The Intelligent Automation of Models

AI improves automation of tasks across various stages of software development. Automated testing tools used by artificial intelligence can be used to create, prioritize, and to execute test cases with little human intervention. This improves test effectiveness by enhancing the coverage and test time taken thus making sure the software entities are tested clearly before deployment.

III. CHALLENGES OF AI IN SOFTWARE ENGINEERING

By adapting Artificial Intelligence (AI) in software engineering provides several benefits, though it creates challenges that need to be addressed carefully. The following are the challenges faced while using AI-driven software development that impacts the trust, reliability, and the sustainability of the practices.

A. Reliability and Accuracy of AI Systems

One of the major concerns is whether the code generated by AI or the generated recommendations are reliable consistently. AI models can sometimes produce incorrect or suboptimal code suggestions, especially when trained on biased or incomplete datasets. Critical functional defects along with security issues may occur within the software system if there is no proper validation check.

B. Data Privacy and Security

It is a general practice for AI systems to use large amounts of data, in order to perform with good results. This makes users concerned about data privacy, particularly when AI uses sensitive or proprietary data for training or prediction. Hence making sure that AI systems follow the data protection regulations and protect confidential data is a significant challenge.

C. Ethical and Bias Issues

A well-known drawback about AI models is that it can unknowingly develop biases from training data. This can lead to unfair or discriminatory results in software functionality. Also, the critical thinking and problem-solving skills of developers can be affected when they overuse AI systems and its automated suggestions. These ethical considerations are important and need to be handled diligently in order to ensure the AI system is used responsibly.

IV. FUTURE DIRECTIONS AND RECOMMENDATIONS

To fully harness the power of Artificial Intelligence (AI) in software engineering, it is essential to develop strategies that address the current challenges while promoting responsible, efficient, and ethical AI integration.

The following listed recommendations are important for shaping up the future of AI adoption in software engineering.

A. Implementing Ethical Guidelines

It is crucial to establish clear ethical guidelines for AI use in software engineering. These guidelines should address concerns about bias, developer dependency, and the social impact of AI-driven automation. Additionally, training programs can help developers become more aware of potential ethical risks and teach them how to use AI responsibly.

B. Strengthening Data Privacy and Security

Future AI solutions must prioritize data protection by using secure training methods, such as federated learning, which allows AI models to learn without directly accessing sensitive data. This can help preserve privacy while still benefiting from AI-driven insights.

C. Encouraging Human-AI Collaboration

Instead of fully automating software development, future approaches should emphasize collaboration between human developers and AI systems. AI should be used to assist and enhance human decision-making, not to replace it. This balance can preserve essential software engineering skills while still benefiting from AI's speed and accuracy.

V. CONCLUSION

Artificial Intelligence (AI) is reshaping the software engineering landscape by offering new solutions that significantly improve productivity, code quality, and development efficiency. Through applications such as automated code generation, intelligent testing, and predictive defect detection, AI-driven tools have introduced transformative improvements across various stages of the software development lifecycle. However, as this paper has highlighted, several challenges still need to be addressed, including issues surrounding trust and explainability, ethical concerns, and the risk of human skill loss.

The future of AI in software engineering depends on the responsible and strategic integration of these technologies. Strengthening data privacy and developing simple AI systems will help bridge the existing gaps. At the same time, creating a collaborative environment where AI tools augment rather than replace human developers is key to maintaining essential software engineering expertise.

By following these recommended actions, the software engineering community can successfully balance the benefits of AI with its risks, ensuring that AI-driven development leads to sustainable innovation and long-term growth. AI will undoubtedly continue to evolve as a critical component of modern software engineering, and its responsible adoption will be essential for shaping the future of the field.

REFERENCES

[1]. Finio, Matthew.: AI in software development, IBM (2024).
[2]. Sida Peng, Eirini Kalliamvakou, Peter Cihon, Mert Demirer.: The Impact of AI on Developer Productivity. In: Evidence from GitHub Copilot, Microsoft.com (2023).
[3]. Kalliamvakou, Eirini.: Research - quantifying GitHub Copilot's impact on developer productivity and happiness. In: GitHub Blog (2024).
[4]. S. Martínez-Fernández.: Software Engineering for AI-Based Systems. In: A Survey, ACM Transactions on Software Engineering and Methodology, vol. 31, no. 2, pp. 1–58, doi: 10.1145/3487043 (2022).
[5]. P. Jain.: Interaction between Software Engineering and Artificial Intelligence- A Review. In: International Journal on Computer Science and Engineering (IJCSE), vol. 3, no. 12, pp. 3774–3779 (2011)
[6]. A. Arpteg, P. Ab, P. Ab, and J. Bosch.: Software Engineering Challenges of Deep Learning. In: 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), pp. 50–59, 2018, doi: 10.1109/SEAA.2018.00018 (2018)
[7]. D. Sculley.: Hidden technical debt in machine learning systems. In: Advances in Neural Information Processing Systems, vol. 2015-January, pp. 2503–2511 (2015).
[8]. Jörg Rech, Klaus-Dieter Althoff.: Artificial intelligence and software engineering: Status and future trends, SE, KI, vol. 3, pp. 5–11, (2004)
[9]. T. Chen.: ChatGPT and other artificial intelligence applications speed up scientific writing. In: Journal of the Chinese Medical Association, 86(4):351-353. doi: 10.1097/JCMA.0000000000000900. Epub 2023 Feb 14. PMID: 36791246 (2023).
[10]. Min, Alfonso.: Artificial Intelligence and Bias: Challenges, Implications, and Remedies. In: Journal of Social Research. 2. 3808-3817. 10.55324/josr.v2i11.1477, (2023).

[11]. Marar, Hazem.: Advancements in software engineering using AI. In: Computer Software and Media Applications. 6. 3906. 10.24294/csma. v6i1.3906, (2024).
[12]. J. Sauvola, S. Tarkoma, M. Klemettinen, J. Riekki, D. Doermann, Future of software development with generative AI. In: Automated Software Engineering, Vol. 31, No. 1, (2024).
[13]. O. Elazhary.: Investigation of the interplay between developers and automation. In: Proc. 43rd Int. Conf. Software Engineering: Companion Proc. (ICSE '21), pp. 153-155. Doi: 10.1109/MS.2023.3265877, (2021).