

УДК 004.8

IMPACT OF ARTIFICIAL INTELLIGENCE ON SOFTWARE DEVELOPMENT EFFICIENCY

Masloynkin A.A., 1-st year student of group PSb-252

Scientific supervisor: Graborskaya I.V., senior lecturer of the
department of foreign languages

Kuzbass State Technical University named after T.F. Gorbachev, Kemerovo

Introduction

The rapid development of artificial intelligence technology over the past few years has transformed virtually every field of activity worldwide, including the software development industry. The emergence of large language models trained on code has enabled new approaches to software development, including automated code generation and assistance [1]. Large language models (LLMs) and various AI-powered assistants have become tools capable of checking, generating, correcting, and explaining code in seconds. The integration of AI tools into workflows raises an important question regarding efficiency for software developers. Various reports and academic studies indicate significant improvements in development speed and productivity; however, these benefits may be offset by issues related to the correctness and reliability of AI-generated code [2].

The aim of this paper is to analyze the impact of artificial intelligence technologies on software development efficiency and to evaluate their influence on developer productivity.

To achieve this aim, the following objectives are defined:

- to examine the current state of AI implementation in software development;
- to classify modern AI-based development tools;
- to analyze empirical studies on developer productivity;
- to assess the overall effectiveness of AI-assisted development.

This article examines the current state of artificial intelligence implementation in software development and provides a structured analysis of performance metrics. Based on this data, we assess both the potential of this technology and its limitations in software development.

Artificial Intelligence Tools in Modern Software Development

AI-based tools have been rapidly expanding in software development. These tools are broadly divided into four functional groups: 1) code completion and generation, 2) automated testing and error detection, 3) code analysis and verification, and 4) documentation generation. The technological foundation of these systems is based on large language models trained on code corpora, which learn statistical patterns of programming languages [1].

In recent years, several AI programming tools have been introduced, including GitHub Copilot, Amazon CodeWhisperer, Google Gemini Code Assist, and Meta Code Llama, each offering different functionality and implementation approaches.

AI-powered tools are increasingly being used for code analysis, testing, and vulnerability detection in modern development processes.

Table 1

Tool	Developer	Year	Key Capability	Primary Use Case
GitHub Copilot	GitHub / OpenAI	2021	Inline code generation from NL comments and code context	Code completion and generation
Amazon CodeWhisperer	Amazon Web Services	2023	Integrated security vulnerability scanning alongside suggestions	Enterprise and AWS cloud development
Google Gemini Code Assist	Google DeepMind	2024	Multi-file context, 1M-token context window	Full-project AI assistance
Tabnine	Tabnine Ltd.	2019	On-premise deployment, team-trained models	Privacy-focused enterprise environments
Meta Code Llama	Meta AI	2023	Open-source weights, customizable fine-tuning	Research and self-hosted deployments

Comparison of Major AI Coding Assistants

Impact on Developer Productivity

One important aspect of AI's impact on software development is its impact on developer efficiency. A randomized controlled trial by Peng et al. [3] found that developers using GitHub Copilot completed tasks significantly faster, demonstrating a time reduction of more than 50% compared to the control group. The controlled experimental design minimizes the influence of skill level or previous experience, making this method one of the most reliable.

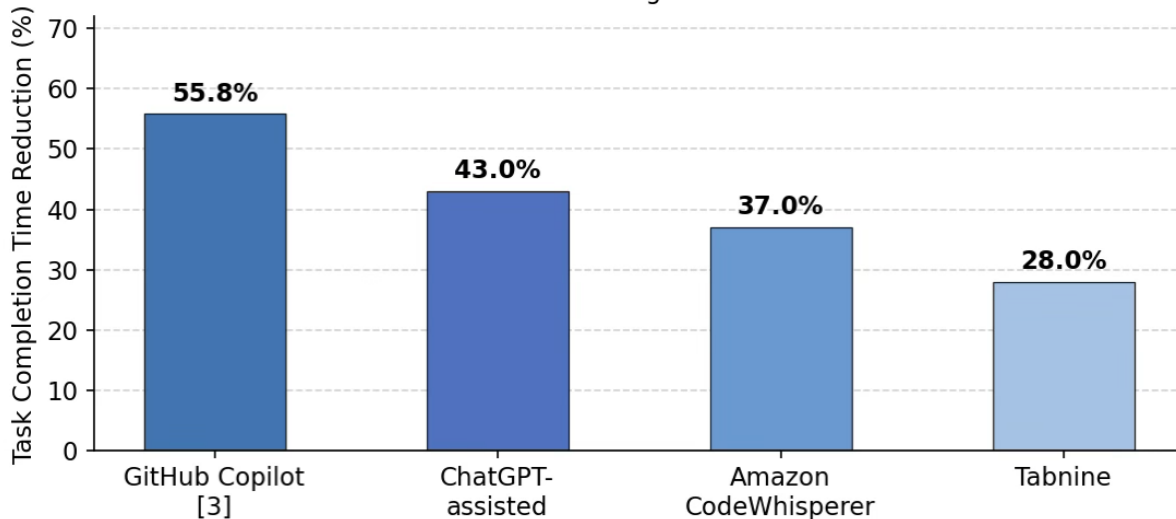
A large-scale industry survey by GitHub [5] reported that 88% of developers felt more productive when using Copilot, while 74% stated they could focus on more meaningful tasks. Ziegler et al. [6] analyzed large-scale usage data and found that a substantial proportion of AI-generated suggestions are accepted by developers, often representing non-trivial multi-token code completions. According to the McKinsey Global Institute [4], generative AI has the potential to accelerate software development tasks by approximately 20–45%, depending on the task and context. McKinsey noted that gains vary significantly by task type, organizational context, and developer seniority.

A study by Avdeev and Belyaeva [7] reported efficiency gains in routine development tasks, while also noting an adaptation period required for effective use of AI tools. The study highlights that AI productivity gains may be partially offset in the short term by onboarding and workflow adjustment costs.

Table 2 – Summary of Empirical Studies on AI-Assisted Developer Productivity

Study	Method	Sample Size	Key Finding
Peng et al. [3], 2023	Randomized controlled trial	95 professional developers	More than 50% in task completion time with GitHub Copilot
GitHub Survey [5], 2022	Online developer survey	2,000+ developers	88% feel more productive; 74% can focus on more meaningful work
Ziegler et al. [6], 2022	Telemetry analysis	Millions of Copilot interactions	A substantial proportion of suggestions are accepted, often representing multi-token code completions [6]
McKinsey Global Institute [4], 2023	Industry analysis	Multiple enterprises globally	20–45% acceleration of software development lifecycle tasks
Авдеев, Беляева [7], 2023	Questionnaire survey (Russia)	124 software developers	Reported efficiency gains in routine tasks and a required adaptation period [7]

Figure 1. Developer Task Completion Time Reduction with AI Coding Assistants



Conclusion

This paper has examined the impact of artificial intelligence on software development, focusing on its role in transforming development workflows and improving developer productivity. The analysis shows that AI-powered tools, particularly those based on large language models, have become an integral part of modern software engineering practices.

Empirical evidence demonstrates that AI-assisted development can significantly accelerate programming tasks, with controlled experiments and industry reports consistently indicating substantial time savings and improved efficiency [3; 5]. In addition, large-scale surveys suggest that developers perceive these tools as beneficial, enabling them to focus on more complex and meaningful aspects of their work [6].

Overall, the integration of AI into software development represents a major shift toward automation and augmentation of human capabilities. While the degree of impact varies depending on context and task type, current evidence suggests that AI tools have strong potential to enhance productivity and reshape software engineering processes in the coming years.

References:

1. Chen M., Tworek J., Jun H. et al. Evaluating Large Language Models Trained on Code [Electronic resource] // arXiv preprint arXiv:2107.03374. — 2021. — URL: <https://arxiv.org/abs/2107.03374> (accessed 15.03.2026).

2. Dakhel A.M., Majdinasab V., Nikanjam A. et al. GitHub Copilot AI pair programmer: Asset or liability? // Journal of Systems and Software. — 2023. — Vol. 203. — Article 111734. — DOI: 10.1016/j.jss.2023.111734 (accessed 15.03.2026)

3. Peng S., Kalliamvakou E., Cihon P., Demirer M. The impact of AI on developer productivity: Evidence from GitHub Copilot [Electronic resource] // arXiv preprint arXiv:2302.06590. — 2023. — URL: <https://arxiv.org/abs/2302.06590> (accessed 17.03.2026).

4. McKinsey Global Institute. The economic potential of generative AI: the next productivity frontier [Electronic resource]. — McKinsey & Company, 2023. — 68 p. — URL: <https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/the-economic-potential-of-generative-ai-the-next-productivity-frontier> (accessed 19.03.2026).

5. GitHub, Inc. Research: quantifying GitHub Copilot's impact on developer productivity and happiness [Electronic resource] // GitHub Blog. — 2022. — URL: <https://github.blog/2022-09-07-research-quantifying-github-copilots-impact-on-developer-productivity-and-happiness/> (accessed 19.03.2026).

6. Ziegler A., Kalliamvakou E., Li X.A. et al. Productivity assessment of neural code completion // Proceedings of the 6th ACM SIGPLAN International

Symposium on Machine Programming (MAPS 2022). — New York: ACM, 2022. — P. 21–29. — DOI: 10.1145/3520312.3534864 (accessed 19.03.2026).

7. Авдеев В.В., Беяева И.С. Анализ эффективности применения инструментов на основе искусственного интеллекта в процессах разработки программного обеспечения // Молодой учёный. — 2023. — № 45 (440). — С. 44–49. — URL: <https://moluch.ru/archive/440/96345/> (accessed 21.03.2026).