

УДК 004.8:004.42

Ihor Shkurin¹, Nataliia Zhukova²

¹ student of group CST-114 NU “Zaporizhzhia Polytechnic”

² PhD (Philology), Associate Professor NU “Zaporizhzhia Polytechnic”

THE RISE OF AI-POWERED CODE GENERATION

Artificial Intelligence (AI) has changed software development in the last 2-3 years. How developers create code now is revolutionized by AI-powered code generators like GitHub Copilot, ChatGPT, or OpenAI Codex, which may increase productivity and minimize the human element.

AI coding has both benefits and challenges.

One major benefit is AI-assisted code generation. These SM assistants help programmers by suggesting code, creating whole blocks of functions, and even helping troubleshoot already written code. AI models may predict and suggest code snippets by examining large datasets and open-source repositories, enabling programmers to create software faster and with fewer bugs. AI-powered code generators are also flexible and useful for inexperienced and experienced engineers, as they support many popular programming languages.

Another advantage is the automation of repetitive coding tasks. Automating repetitive and boring processes allows developers to focus on more complex problem-solving aspects. Additionally, by sticking to best-known practices and quickly detecting any vulnerabilities, these tools help to improve the quality of the final code. AI can also enhance collaboration among team members by offering suggestions in real time and reducing the time spent on debugging and testing.

AI tools contribute to learning and skill improvement. Beginners can use AI-powered assistants as educational tools, gaining insights into coding best practices and learning new programming techniques. They provide instant feedback, helping developers refine their skills while working on real projects.

However, AI-generated code has its challenges and faults. Problems such as biased training data, a lack of innovation, and potential security vulnerabilities are some of the issues that may arise. AI models are trained on existing code, which might reinforce outdated or inefficient coding practices. Additionally, AI-generated code may lack proper documentation and comments, making it difficult to understand and maintain in the long run. Even though AI can assist with coding, human oversight is needed to guarantee accuracy and security.

Another drawback is over-reliance on AI tools, which may lead to a decline in problem-solving skills among developers. If programmers become too dependent on AI-generated suggestions, they may struggle to write complex code without assistance. Additionally, AI may sometimes generate incorrect or suboptimal solutions, requiring developers to review and test the output carefully.

Despite these obstacles, the development of AI-powered tools is positively influencing the direction of global software engineering. These tools will become even more sophisticated as technology advances, further boosting programming productivity and creativity. AI-driven development is expected to evolve with improved accuracy, better understanding of context, and stronger integration with development workflows, making software engineering more efficient and innovative.