



Evaluating The Impact Of Ai Tools On Software Quality And Developer Productivity: A Software Engineering Study

1 Shubham Sunil Shinde, 2 Pavan Bhagwanrao Ghode, 3 Suraj Vasant Sabale, 4 Ashutosh Suresh Varma

1 MCA Student, 2 MCA Student, 3 MCA Student, 4 MCA Student

Pharate Patil Management Institute, Pune, affiliated to Savitribai Phule Pune University, Pune, India

Abstract: The rapid advancement of Artificial Intelligence (AI) has significantly influenced modern software development practices. AI-based development tools such as code completion assistants, automated testing tools, and static code analysis systems are increasingly adopted to enhance software quality and developer productivity. This study evaluates the impact of AI-based development tools on software engineering outcomes, with a specific focus on code quality, development speed, error reduction, and developer efficiency. The research adopts a survey-based methodology involving software developers and computer science students who actively use AI-assisted development tools. Primary data was collected using a structured Google Form questionnaire, and secondary data was gathered from existing research literature. The findings suggest that AI-based tools positively influence developer productivity by reducing coding time, improving code accuracy, and assisting in debugging. Additionally, respondents reported improvements in software quality through early defect detection and code optimization. The study concludes that AI-based development tools play a significant role in modern software engineering practices and are likely to become integral components of future software development environments.

Index Terms - Artificial Intelligence, Software Engineering, Developer Productivity, Software Quality, AI-Based Tools

I. Introduction

Software engineering has continuously evolved with the advancement of new technologies aimed at improving software quality, development speed, and reliability. In recent years, Artificial Intelligence (AI) has emerged as a transformative force in the field of software engineering. AI-based development tools are increasingly being integrated into the software development lifecycle (SDLC) to assist developers in tasks such as code generation, debugging, testing, and maintenance.

AI-powered tools such as intelligent code editors, automated testing systems, and recommendation-based development assistants are designed to reduce manual effort and minimize human errors. These tools analyze large volumes of existing source code and development patterns to provide real-time suggestions and solutions. As a result, developers can focus more on design and logic rather than repetitive coding tasks. One of the major benefits of AI-based development tools is the improvement in developer productivity. By automating routine tasks and offering instant code suggestions, AI tools help developers complete software projects in less time. Additionally, AI-assisted debugging and testing tools contribute to early error detection, thereby enhancing overall software quality.

However, increasing reliance on AI tools also introduces several challenges. Over-dependence on AI-generated solutions may negatively impact developers' problem-solving abilities and learning processes. There are also concerns related to code reliability, ethical usage, intellectual property rights, and security vulnerabilities in AI-generated code.

This study aims to evaluate the impact of AI-based development tools on software quality and developer productivity from a software engineering perspective. By analyzing existing literature and collecting primary data through a survey, this research attempts to identify both the advantages and limitations of AI-assisted development tools and propose balanced approaches for their effective adoption.

II. LITERATURE REVIEW

Several researchers have studied the role of Artificial Intelligence in software engineering and its influence on development practices. Existing literature highlights both productivity improvements and emerging concerns related to AI-assisted development.

GitHub (2023) reported that developers using AI-based code completion tools experienced a significant reduction in task completion time, indicating improved development efficiency. The study emphasized that AI tools are particularly beneficial for repetitive coding and boilerplate generation tasks.

Microsoft Research (2024) examined the impact of AI tools on software teams and found that AI assistance enhances coding speed and consistency. However, the study also noted a decline in manual debugging skills among developers who heavily relied on AI-generated suggestions.

IEEE Spectrum (2022) discussed security risks associated with AI-generated code and highlighted that automated code suggestions may introduce vulnerabilities if not carefully reviewed. The study stressed the importance of human oversight in AI-assisted programming.

Google Research (2023) analyzed the limitations of large language models in software development and concluded that while AI performs well in simple coding tasks, it still struggles with complex algorithmic logic and system-level design decisions.

MIT Technology Review (2024) raised ethical and legal concerns regarding AI-generated code, particularly related to copyright infringement and ownership. The study emphasized the need for clear policies and guidelines to regulate the use of AI tools in professional software development.

Recent academic studies have also indicated that AI tools positively influence beginner developers by providing learning support and reducing initial development barriers. However, excessive reliance on AI tools may reduce creativity and independent thinking among developers.

Overall, the literature suggests that AI-based development tools offer measurable benefits in terms of productivity and code quality, but they must be used responsibly to avoid long-term negative impacts on software engineering skills and practices.

III. RESEARCH METHODOLOGY

3.1 Research Design

This research adopts a descriptive and exploratory research design to analyze the impact of AI-based development tools on software quality and developer productivity.

3.2 Data Sources

- **Primary Data:** Collected through a structured online questionnaire distributed to MCA students and software developers.
- **Secondary Data:** Research articles, journals, conference papers, and online technical reports related to AI and software engineering.

3.3 Sample Size

The study includes responses from 25 participants consisting of MCA students and software developers.

3.4 Data Collection Tool

Google Forms was used to design and distribute the questionnaire. The responses were collected and analyzed using basic statistical methods.

3.5 Data Analysis Technique

The data collected was analyzed using descriptive statistics such as percentages and graphical representation to interpret trends and patterns in developer responses.

Survey Questionnaire (Primary Data)

The primary data was collected using a structured questionnaire consisting of ten questions. The questionnaire focused on respondents' familiarity with AI-based development tools, frequency of usage, impact on productivity, error reduction, dependency concerns, security implications, and overall satisfaction. The questions were designed using a mix of multiple-choice and Likert-scale formats to ensure clarity and ease of response.

IV. RESULTS AND DISCUSSION

The survey was conducted among 25 respondents, including MCA students and early-career software developers who actively use AI-based development tools. The collected responses were analyzed using descriptive statistics to understand the impact of AI tools on software quality and developer productivity.

4.1 Awareness and Usage of AI-Based Development Tools

The results indicate that 88% of respondents are familiar with AI-based development tools, while 12% reported limited familiarity. Frequently used tools include AI-powered code editors, code completion assistants, and automated debugging tools.

4.2 Impact on Developer Productivity

A significant proportion of respondents (76%) agreed that AI-based tools improve coding productivity by reducing development time and minimizing repetitive tasks. Around 16% reported moderate improvement, while 8% observed little or no impact on productivity.

4.3 Error Reduction and Code Quality

Approximately 72% of participants stated that AI tools help reduce coding errors through real-time suggestions and automated debugging. However, 20% emphasized that manual review is still necessary to ensure code reliability, and 8% expressed concerns regarding inaccurate or incomplete AI-generated code.

4.4 Dependency and Learning Impact

About 44% of respondents acknowledged partial dependency on AI tools during development activities. Meanwhile, 56% believed that AI tools support learning when used responsibly but should not replace core programming skills and logical thinking.

4.5 Security and Reliability Concerns

Nearly 36% of participants expressed concerns about potential security vulnerabilities and reliability issues in AI-generated code. The remaining respondents felt confident using AI tools provided that proper validation and testing are performed.

4.6 Overall Satisfaction

Overall satisfaction levels were positive, with 80% of respondents expressing satisfaction with AI-based development tools as coding assistants. The findings indicate that AI tools significantly enhance productivity and code quality when used as supportive technologies rather than complete replacements for developers.

Discussion

The survey results demonstrate that AI-based development tools contribute positively to software engineering practices by improving development efficiency and supporting error detection. However, concerns related to over-dependence, security, and reduced problem-solving skills highlight the importance of balanced usage. These findings align with existing literature, reinforcing the view that AI tools are best utilized as assistive technologies under human supervision.

V. CONCLUSION

This study concludes that AI-based development tools play a significant role in improving software quality and developer productivity in modern software engineering. While AI tools offer speed, efficiency, and consistency, their limitations highlight the importance of responsible usage and human supervision. Future research can focus on long-term impacts of AI tools on developer skills, advanced AI-driven testing frameworks, and ethical governance models for AI-assisted software development.

VI. ACKNOWLEDGMENT

The authors express sincere gratitude to their research guide and the Department of Computer Applications Pharate Patil Management Institute, Pune Affiliated to Savitribai Phule Pune University, Pune, India for their guidance and support throughout this research work.

VII. REFERENCES

1. GitHub. (2023). *The impact of AI tools on developer productivity.*
2. Microsoft Research. (2024). *AI-assisted software engineering: Benefits and challenges.*
3. IEEE Spectrum. (2022). *Security risks in AI-generated code.*
4. Google Research. (2023). *Limitations of AI models in software development.*
5. MIT Technology Review. (2024). *Ethical concerns of AI-generated software.*

