Journal of Science Research International (JSRI)
ISSN: 2456-6365 (Online)

OPEN ACCESS

IMPACT OF CHATGPT ON BEGINNER AND EXPERIENCED DEVELOPERS

Dhanya Vinish*, Sanika Chavan, Vaishnavi Thosar and Tanishka Jalvi

Department of Information Technology

Pillai College of Arts, Commerce & Science (Empowered Autonomous), New Panvel, Navi Mumbai, India

**Corresponding author E-mail: dvinish@mes.ac.in*

Received: 09 February 2026	Revised: 21 February 2026	Accepted: 20 March 2026	Published: 31 March 2026

DOI: **https://doi.org/10.5281/zenodo.19557193**

Abstract:

The rapid advancement of artificial intelligence has significantly influenced the software development industry, particularly through tools such as ChatGPT. This study investigates the impact of ChatGPT on beginner and experienced programmers in terms of learning effectiveness, productivity, problem-solving ability, and code quality. A comparative research design was employed, and data was collected from 17 participants using a structured questionnaire. The findings indicate that 94% of participants reported improved learning speed, while 82% observed enhanced productivity. However, 82% of respondents expressed concerns about overdependence, and 88% acknowledged that ChatGPT can generate incorrect or inefficient solutions. The results suggest that ChatGPT acts as a skill accelerator for beginners but may hinder independent problem-solving if overused. For experienced developers, it serves as a productivity-enhancing tool, although careful evaluation of outputs is required. The study concludes that ChatGPT is a supportive tool rather than a replacement for fundamental programming knowledge, and its effectiveness depends on the user's experience and critical thinking ability.

Keywords: *ChatGPT, Artificial Intelligence, AI-Assisted Programming, Programming Education, Developer Productivity, Learning Effectiveness, Problem-Solving Ability.*

1. Introduction

The rapid advancement of Artificial Intelligence (AI) has significantly transformed the software development landscape, introducing intelligent systems that enhance automation, efficiency, and decision-making processes. Among these innovations, ChatGPT has emerged as a widely adopted AI-powered tool capable of generating human-like text, assisting in code development, debugging errors, and explaining complex programming concepts. Its accessibility and ease of use have contributed to its widespread adoption among developers with varying levels of expertise.

For beginner programmers, learning to code often presents challenges related to syntax understanding, logical reasoning, and debugging. AI-assisted tools such as ChatGPT provide immediate feedback, step-by-step explanations, and code suggestions, which can significantly reduce the learning curve and improve confidence levels. However, reliance on such tools may also discourage active problem-solving and limit the development of critical thinking skills if learners depend heavily on generated solutions without understanding the underlying logic.

In contrast, experienced developers utilize AI tools primarily as productivity enhancers rather than learning aids. ChatGPT supports tasks such as code generation, optimization, documentation, and testing, allowing developers to focus on higher-level design and problem-solving. Despite these advantages, concerns remain regarding the accuracy, reliability, and security of AI-generated outputs, necessitating careful validation and human oversight.

While existing research highlights the benefits of AI-assisted programming tools, there is limited comparative analysis of their impact across different experience levels. Most studies focus either on educational benefits or productivity improvements independently. Therefore, this study aims to provide a comparative evaluation of the impact of ChatGPT on beginner and experienced developers, focusing on learning effectiveness, productivity, and problem-solving ability.

2. Literature review

Recent advancements in AI-assisted programming tools have led to increased interest in their impact on both software development and programming education. Studies such as Noy and Zhang (2023) demonstrate that generative AI tools significantly enhance productivity by reducing the time required to complete coding tasks. Similarly, industry research highlights that tools like GitHub Copilot assist developers in code generation and debugging, thereby improving efficiency and reducing development effort.

In educational contexts, AI tools are increasingly used to support learning by providing instant feedback and personalized assistance. These tools help beginners overcome common challenges such as syntax errors and logical misunderstandings, enabling faster acquisition of programming skills. However, several studies caution that overreliance on AI tools may reduce engagement in problem-solving activities, potentially weakening deep learning and conceptual understanding.

For experienced developers, AI-assisted tools are primarily used to automate repetitive tasks, generate boilerplate code, and enhance workflow efficiency. Research indicates that while these tools improve perceived productivity, developers often spend additional time validating and refining AI-generated outputs, particularly in complex scenarios. This suggests that the benefits of AI tools may vary depending on task complexity and user expertise.

Furthermore, existing literature suggests that AI tools provide immediate benefits in terms of efficiency and accessibility but raise concerns regarding long-term skill development and dependency. Despite these insights, there remains a lack of comprehensive studies comparing the impact of AI tools across different experience levels. This gap highlights the need for further research, which this study aims to address.

3. Purpose of study

The primary objective of this research is to examine the impact of ChatGPT on software developers with varying levels of experience, specifically focusing on beginner and experienced professionals. The study aims to evaluate how AI-assisted tools influence key aspects of programming, including learning effectiveness, productivity, problem-solving ability, and overall development practices.

For beginner programmers, the study investigates how ChatGPT supports the learning process by facilitating concept understanding, error correction, and skill development. It also seeks to analyze how the use of AI tools affects the learning curve, confidence levels, and engagement with programming tasks. In addition, the study examines the potential negative effects of overdependence on AI tools, particularly in relation to reduced critical thinking, shallow learning, and limited development of independent problem-solving abilities.

For experienced developers, the study evaluates the role of ChatGPT in improving efficiency, optimizing code, and reducing development time. It further explores how AI tools influence workflow practices, including debugging, documentation, and code refactoring. Additionally, the research investigates challenges associated with AI-generated outputs, such as the need for validation, potential inaccuracies, security concerns, and their impact on code quality and maintainability.

The study also aims to provide a comparative analysis between beginner and experienced developers to understand how experience level influences the effectiveness, usage patterns, and perceived benefits of ChatGPT. This comparison helps in identifying differences in dependency, trust, and application of AI tools across user groups.

Furthermore, the research seeks to contribute to the broader understanding of human–AI interaction in software development by examining how developers integrate AI tools into their workflows. By doing so, the study aims to provide practical insights for educators, learners, and industry professionals on the responsible and effective use of AI-assisted programming tools.

Ultimately, this research aims to highlight both the advantages and limitations of ChatGPT and to support the development of balanced strategies that maximize its benefits while minimizing potential risks in both educational and professional environments.

4. Data analysis and hypothesis testing

4.1 Main research question

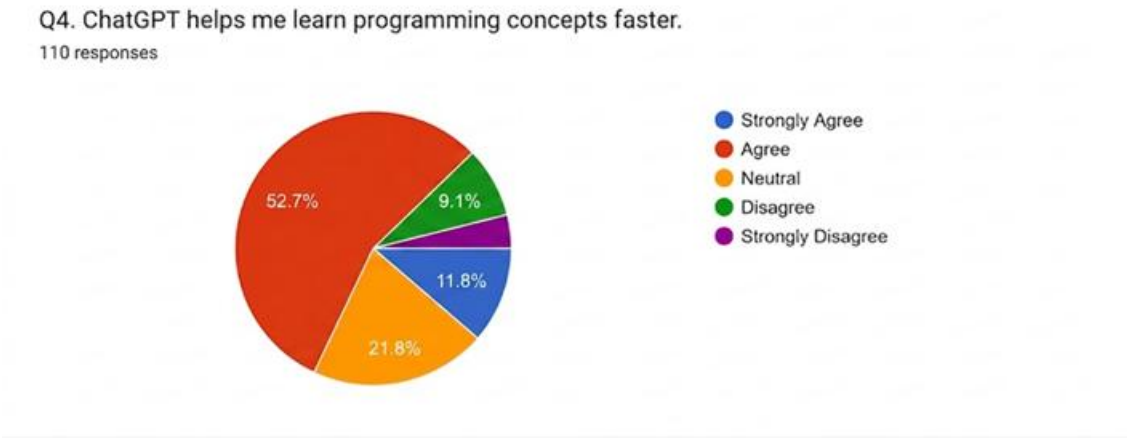

Figure 1: How ChatGpt helps to learn programming concepts faster?

4.2 Sub-research questions

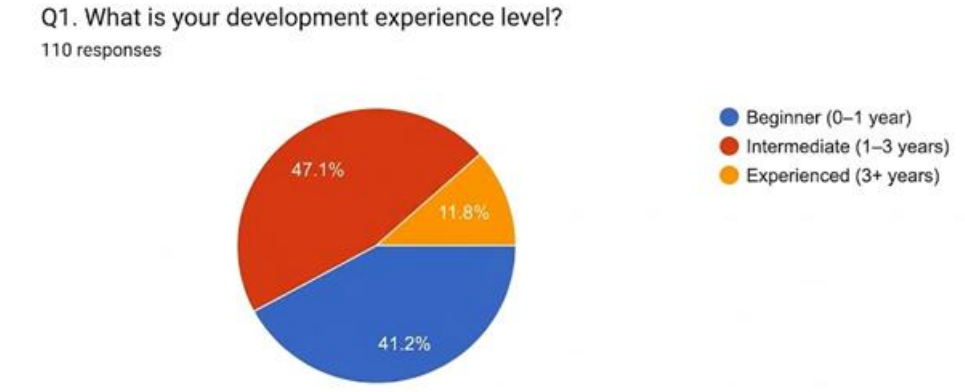

Figure 2: What is your development experience level with ChatGPT?

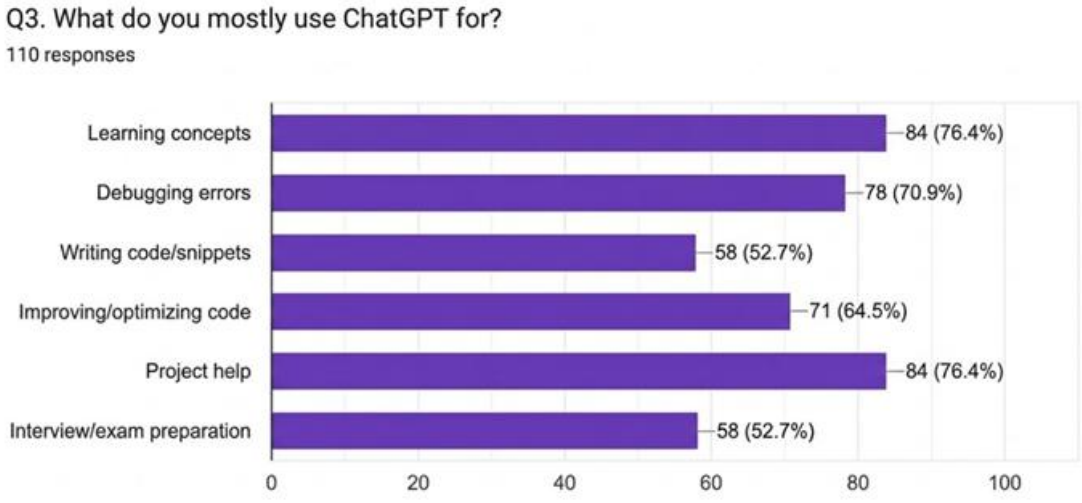

Figure 4: Does ChatGPT help beginners learn programming concepts faster?

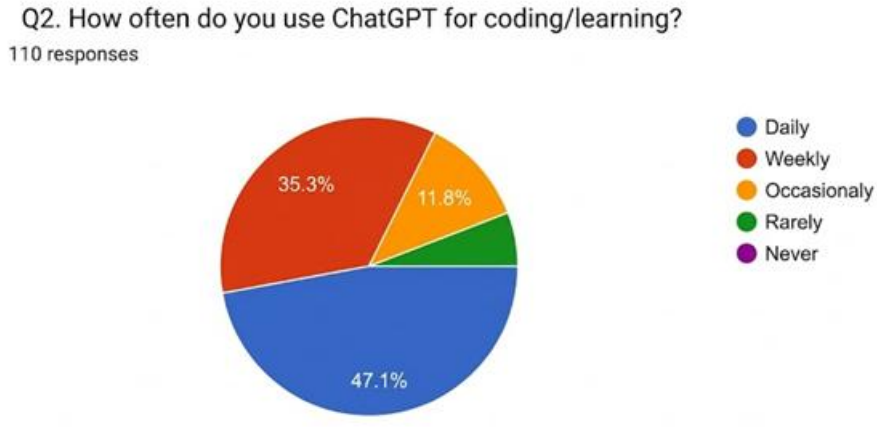

Figure 5: How often do you use ChatGpt for coding/learning?

4.3 Hypothesis

The data for this study was collected using a structured questionnaire from participants with varying levels of programming experience. The sample predominantly consists of early-career developers, which may introduce a slight bias toward beginner perspectives.

Hypothesis 1: Learning impact

The results show that 94% of participants agreed that ChatGPT helps in learning programming concepts faster. This indicates a strong positive impact on learning among beginners.

H1 is supported

Hypothesis 2: Productivity impact

Approximately 82% of respondents reported improved productivity using ChatGPT, particularly in debugging and code generation.

H2 is supported

Hypothesis 3: Problem-solving ability

Responses regarding problem-solving improvement were mixed, with many participants selecting neutral options.

This suggests that while ChatGPT assists in solving problems, it does not significantly enhance independent thinking.

H3 is partially supported

Hypothesis 4: Experience-Based Differences

The findings indicate that beginners primarily use ChatGPT for learning, while experienced developers use it for productivity and efficiency tasks.

H4 is supported

Overall finding

The analysis shows that ChatGPT significantly improves learning and productivity but raises concerns about dependency and limited impact on independent problem-solving.

5. Methodology

This study adopts a quantitative and comparative research design to analyze the impact of ChatGPT on developers with different levels of programming experience. The research focuses on evaluating variations in learning effectiveness, productivity, and problem-solving ability between beginner and experienced developers.

Data was collected using a structured questionnaire distributed through an online survey platform. A convenience sampling method was employed to gather responses from participants with diverse backgrounds, including students, early-career developers, and experienced professionals. Participants were categorized into three groups—beginners, intermediate developers, and experienced developers—based on their years of programming experience, enabling a comparative analysis across experience levels.

The questionnaire consisted of both multiple-choice and Likert-scale questions, with responses measured on a five-point scale ranging from “strongly agree” to “strongly disagree.” The survey instrument was designed to evaluate key variables such as learning effectiveness, productivity improvement, problem-solving ability, and dependency on AI tools. Each variable was represented through multiple questions to ensure consistency and

improve the reliability of the collected data. The questionnaire was structured in alignment with the research objectives and hypotheses to facilitate systematic analysis.

A total of 110 valid responses were collected, with a majority representing early-career developers. While this provides a broader dataset compared to smaller exploratory studies, the distribution of participants across experience levels may influence the overall findings and interpretation.

The collected data was analyzed using descriptive statistical techniques, including percentage distribution, frequency analysis, and comparative analysis between groups. The analysis focused on identifying patterns and trends in how ChatGPT influences different aspects of programming across varying experience levels. Visual tools such as bar charts and pie charts were used to represent the data and support interpretation.

In addition to descriptive analysis, a comparative approach was applied to evaluate differences in responses between beginner and experienced developers, particularly in relation to learning outcomes, productivity gains, and problem-solving capabilities. This approach enabled the validation of the proposed hypotheses and provided insights into experience-based variations in AI tool usage.

Despite the relatively larger sample size, the study primarily relies on self-reported data, which may introduce response bias. Furthermore, the use of convenience sampling limits the generalizability of the findings. Therefore, the results should be interpreted as indicative trends rather than definitive conclusions, providing a foundation for further research in this area.

6. Limitations

Despite providing valuable insights, this study has several limitations that should be considered when interpreting the results. First, although the sample size (n=110) is moderate, it may still limit the statistical generalizability of the findings to the broader population of software developers. A larger and more diverse sample across different regions and professional backgrounds would provide more robust and representative results.

Second, the study relies on self-reported data collected through questionnaires, which may introduce response bias. Participants may overestimate or underestimate the impact of ChatGPT on their learning and productivity. Additionally, subjective perceptions may not always accurately reflect actual performance or skill improvement.

Third, the sample is predominantly composed of early-career developers, which may skew the findings toward beginner perspectives. This imbalance may affect the comparative analysis, as the experiences and usage patterns of highly experienced professionals may not be fully represented.

Furthermore, the use of convenience sampling limits the external validity of the study. Since participants were not selected through random sampling, the results may not be fully generalizable to the wider developer community.

Another limitation is the lack of experimental validation. The study is based on survey responses rather than controlled experiments, which restricts the ability to establish causal relationships between ChatGPT usage and improvements in learning or productivity.

Additionally, the rapid evolution of AI technologies presents a challenge, as tools such as ChatGPT are continuously updated with new features and capabilities. As a result, the findings of this study may be time-sensitive, and future versions of AI tools may produce different outcomes.

Finally, the study does not account for variations in programming languages, development environments, project complexity, or domain-specific requirements. These factors may significantly influence the effectiveness of AI-assisted tools and should be considered in future research for a more comprehensive analysis.

Conclusion

This study examined the impact of ChatGPT on both beginner and experienced developers, focusing on key factors such as learning effectiveness, productivity, and problem-solving ability. Based on the analysis of 110 responses, the findings indicate that ChatGPT significantly enhances learning speed and confidence among beginners by providing immediate support, explanations, and debugging assistance. However, excessive reliance on AI tools may hinder the development of independent problem-solving skills and critical thinking, highlighting the need for balanced usage.

For experienced developers, ChatGPT serves as an effective productivity-enhancing tool, enabling faster code generation, debugging, and documentation. While it improves efficiency and reduces development time, the necessity of verifying AI-generated outputs introduces additional cognitive effort and raises concerns related to accuracy, security, and maintainability. This suggests that the benefits of AI tools are influenced by the complexity of tasks and the expertise of the user.

The comparative analysis further reveals that the impact of ChatGPT varies significantly based on experience level. Beginners primarily utilize the tool as a learning aid, whereas experienced developers leverage it for workflow optimization and task automation. This distinction emphasizes the importance of user expertise in determining the effectiveness and appropriate use of AI-assisted tools.

Although the study is supported by a moderate sample size, the reliance on self-reported data and convenience sampling limits the generalizability of the findings. Therefore, the results should be interpreted as indicative trends rather than definitive conclusions.

Overall, the study demonstrates that ChatGPT is a valuable support tool rather than a replacement for fundamental programming knowledge. Its effectiveness depends on the user's ability to critically evaluate generated outputs and integrate AI assistance into the development process responsibly.

The findings highlight the importance of promoting responsible and balanced use of AI tools in both educational and professional settings. Future research should focus on larger and more diverse samples, experimental research designs, and longitudinal studies to better understand the long-term impact of AI-assisted programming on skill development, productivity, and software quality.

References

1. Bubeck, S., et al. (2023). *Sparks of artificial general intelligence: Early experiments with GPT-4*. arXiv.
2. GitHub. (2022). *Quantifying GitHub Copilot's impact on developer productivity* (Research report).
3. Noy, S., & Zhang, W. (2023). *Experimental evidence on the productivity effects of generative artificial intelligence* (Working Paper No. 31161). National Bureau of Economic Research.
4. Association for Computing Machinery. (2023). AI-assisted programming and software development productivity. *Communications of the ACM*.
5. Institute of Electrical and Electronics Engineers. (2023). AI tools in software engineering education. *IEEE Xplore Digital Library*.