International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 6, Nov-Dec-2026, ISSN (Online): 2395-566X

Impact of Generative AI on Software Development

Dr. Deepak Singh[1], Anil Birappa Mantage[2]
M.Sc. (Computer Science)
Suryadatta College Of Management, Information Research And Technology, Pune (Sppu
Academic Year 2026-27

Abstract- Generative AI tools such as ChatGPT, GitHub Copilot and Google Gemini are now used by many students and developers to write code, find bugs and explain programs. This paper studies the impact of these tools on software development using secondary data, which means data that was already collected and published by others. The main sources are the Stack Overflow Developer Survey 2025 (more than 33,000 responses to its AI questions) and the 2026 survey (more than 30,000 responses), together with two published studies on GitHub Copilot, one on productivity and one on security. The published figures are tested against four simple hypotheses. The results show that most developers use AI tools regularly and find them useful for writing code, and that a controlled experiment found a clear saving of time. At the same time, very few developers highly trust the output, and a security study found that about 40% of the tested Copilot programs were vulnerable. The paper concludes that generative AI is a useful helper for routine work, but its output must always be checked by a person, and developers should keep learning the fundamentals.

Keywords: Generative AI, Software Development, AI Coding Assistants, Developer Productivity, Code Quality, Secondary Data Analysis

I. INTRODUCTION

Writing software has always needed time, practice and a lot of searching for answers. A developer used to depend on books, forums and documentation when a problem came up. In the last few years this has started to change. Generative AI tools can now take a plain English instruction and give back working code, a fix for an error, or a clear explanation of a difficult function.

Because of this, AI assistants are quickly becoming a normal part of the developer's toolbox, inside code editors and in chat tools. Students use them for assignments and projects, and companies use them to speed up daily tasks. At the same time, many people ask if the code is always correct, if it is safe, and if beginners will stop learning properly when a tool does the thinking for them.

This paper studies the impact of generative AI on software development by bringing together what large published developer surveys and earlier research studies have found, and by checking this evidence against a set of clear hypotheses.

1. Background: How Generative Ai Helps In Coding

Generative AI tools for coding are built on large language models. These models are trained on a huge amount of text and program code, and they learn to predict what should come next. The Transformer design introduced by Vaswani et al. (2017) made this kind of model possible, and later work such as Codex (Chen et al., 2021), which was trained on public code, showed that a model can solve many programming problems from a short description.

In everyday software development, these tools are mainly used for:

- writing new code or small functions from a description,
- completing the line or block of code that the developer is typing,
- finding and explaining errors and bugs,
- explaining unknown code or a new technology in simple words,
- writing test cases, comments and documentation,
- converting code from one language or framework to another.

II. LITERATURE REVIEW

Productivity. Peng et al. (2023) ran a controlled experiment in which developers had to implement an HTTP server in JavaScript as quickly as possible. The group that used GitHub Copilot completed the task 55.8% faster than the group that did not. This supports the common belief that AI assistants save time on routine work.

Usability. Vaithilingam et al. (2022) studied how people actually feel while using code generation tools. Participants liked having a starting point, but many found it hard to understand, edit and debug the generated code. This shows that saving typing time does not always mean the whole task becomes easier.

© 2026 IJSRET

Security. Pearce et al. (2022) gave Copilot 89 scenarios based on well-known software weaknesses, which produced 1,689 programs. About 40% of these programs were found to be vulnerable. This is a warning that AI-written code should be reviewed like any other code, and sometimes more carefully.

Developer opinion. The Stack Overflow Developer Survey is one of the largest yearly surveys of developers. Its 2025 and 2026 editions both include detailed questions on AI tools. Their results are used as the main data of this paper and are shown in Section 7.

In short, earlier work agrees that AI tools help with speed, but it also points to doubts about correctness, security and understanding.

1. Objectives, Research Questions And Hypotheses

Objectives

- To find out how widely generative AI tools are used by developers, using published survey data.
- To study the benefits that developers report from these tools.
- To study how much developers trust AI output and what concerns they have.
- To give simple suggestions for responsible use of AI in software projects.

Research Questions

- RQ1: How widely and how often do developers use generative AI tools?
- RQ2: What evidence is there that these tools save time and help in coding tasks?
- RQ3: How much do developers trust the output, and what accuracy and security concerns exist?
- RQ4: How are AI tools used for learning, and are workplaces ready with rules for their use?

Hypotheses

- H1: Most developers use generative AI tools regularly.
- H2: Generative AI tools save time on coding tasks.
- H3: Most developers do not highly trust AI output, so they need to check it before using it.
- H4: Developers have real concerns about the security and accuracy of AI-generated code.

III. RESEARCH METHODOLOGY

Design. This is a descriptive study based on secondary data analysis. No new data was collected from people by the author. All figures are taken from published sources and are cited in the reference list with direct links.

Data sources. The following sources were used:

- Stack Overflow Developer Survey 2025 (Ref 4): 33,662 responses to the question on AI tool use and 33,244 responses to the question on trust in the accuracy of AI output.
- Stack Overflow Developer Survey 2026 (Ref 5): more than 30,000 responses from 169 countries. The 2026 figures in this paper were taken from news reports of the survey (Ref 8).
- Peng et al. (2023) (Ref 3): a controlled experiment on GitHub Copilot and speed of work.
- Pearce et al. (2022) (Ref 2): a security test of GitHub Copilot code.
- A news report on the 2025 survey (Ref 9), used for the figures on concerns about AI agents.

Analysis. The percentages are used exactly as published, and no number was changed or estimated. Where the paper adds two published percentages together, this is shown clearly (for example, daily plus weekly use). A hypothesis is treated as supported if the published evidence points the same way and, for questions about "most developers", if the figure is above 50%. The figures were read on 19 November 2026.

Ethics. All data used is public and contains no personal information. Every source is cited.

IV. MAIN IMPACT AREAS

1. Speed And Productivity

AI tools reduce the time spent on repeated work such as writing boilerplate code, simple functions, test cases and comments. A developer can reach a first working version faster and spend the saved time on design and on solving the real problem. The gain is biggest for routine tasks and smaller for complex tasks that need deep knowledge of the project.

2. Code Quality And Correctness

A tool can produce code that looks correct but has a hidden mistake, uses an old method, or does not fit the rest of the project. So the time saved in writing can be lost again in testing and debugging if the code is accepted without reading it. Code review and testing become more important, not less.

3.Security And Privacy

AI-generated code can contain security weaknesses, as shown by Pearce et al. (2022). There is also a privacy risk when developers paste private company code or data into an online tool. Teams should have clear rules on what can and cannot be shared with such tools.

© 2026 IJSRET

4. Learning And Skills

For students, AI works like a patient teacher that explains errors and concepts at any time. But if a beginner copies answers without understanding them, the basic skills of logic, debugging and problem solving do not grow. The tool helps most when it is used to learn, and helps least when it is used to avoid learning.

5. Jobs And The Role Of Developers

Many people worry that AI will reduce entry-level jobs. A more careful view is that the work is changing, less time on typing routine code, and more time on understanding requirements, reviewing code, testing, and working with people. Developers who can use AI well and still understand the basics are likely to stay useful.

6. Ethics And Ownership

Questions remain about who owns AI-generated code, whether it may copy licensed code, and who is responsible when it fails. These points matter for companies and should be handled through clear policies.

V. RESULTS AND ANALYSIS OF PUBLISHED DATA

1. How Widely Are Ai Tools Used? (Rq1)

Table 1 shows the answers to the question "Do you currently use AI tools in your development process?" in the Stack Overflow Developer Survey 2025 (Ref 4). In total, 84% of respondents use or plan to use AI tools, up from 76% in the previous year. Among all respondents, 47.1% use AI tools daily and 17.7% use them weekly, which together is 64.8%.

Answer	All respondents	Professional developers	Learning to code	Early career (1-5 years)
Use AI tools daily	47.1	50.6	39.5	55.5
Use AI tools weekly	17.7	17.4	18.7	18.1
Use monthly or less often	13.7	12.8	15.1	11.5
Do not use, but plan to soon	5.3	4.6	7.2	2.5
Do not use and do not plan to	16.2	14.7	19.5	12.3
Number of responses	33,662	26,004	2,843	6,360

Source: Ref 4.

The 2026 survey shows the same direction. Coding assistants or coding agents are used at work by 66% of respondents, general-purpose chat tools by 63%, and only 17.2% say they do not use AI tools (Table 3).

2. Do Ai Tools Save Time And Help In Coding? (Rq2)

The strongest evidence on time comes from the controlled experiment of Peng et al. (2023), where the group using Copilot completed the same task 55.8% faster than the group without it (Table 4). In the 2026 Stack Overflow survey, 74% of respondents found AI useful or very useful for implementation, which is the work closest to writing code (Table 3). The experiment covers only one task, so the result should not be taken as a fixed saving for all work.

3. How Much Do Developers Trust The Output? (Rq3)

Table 2 shows the 2025 answers on trust in the accuracy of AI output. Only 3.1% highly trust it. Adding the published figures, 32.7% trust it at least somewhat, while 45.7% somewhat or highly distrust it. So more developers distrust the accuracy of AI than trust it.

Level of trust	% of respondents
Highly trust	3.1
Somewhat trust	29.6
Somewhat distrust	26.1
Highly distrust	19.6
Trust (highly + somewhat)	32.7
Distrust (somewhat + highly)	45.7

Source: Ref 4. The two total rows were added by the author from the rows above.

The 2026 survey adds that only 6.6% would trust AI for important decisions, while nearly half say they trust it when they can verify the work easily. Also, 78.7% say that knowing the source of AI-generated code or answers is important (Table 3).

Figure	Result	Source
Use coding assistants or coding agents at work (2026)	66%	Ref 5, 8
Use general-purpose chat tools at work (2026)	63%	Ref 5, 8
Use automated agent workflows (2026)	26.2%	Ref 5, 8

© 2026 IJSRET

Figure	Result	Source
Do not use AI tools (2026)	17.2%	Ref 5, 8
Find AI useful or very useful for implementation (2026)	74%	Ref 5, 8
Would trust AI for important decisions (2026)	6.6%	Ref 5, 8
Say source attribution is important for AI code or answers (2026)	78.7%	Ref 5, 8
Employer has published approved AI tools or guidance (2026)	24%	Ref 5, 8
Learners who use AI code-generation tools to learn (2026)	52.7%	Ref 5, 8
Concerned about accuracy of AI agents (2025)	87%	Ref 9
Concerned about security and data privacy of AI agents (2025)	81%	Ref 9

Study	What was tested	Main result
Peng et al. (2023), Ref 3	Developers implemented an HTTP server in JavaScript, with and without Copilot	Group with Copilot completed the task 55.8% faster
Pearce et al. (2022), Ref 2	89 security-related scenarios, producing 1,689 programs	About 40% of the programs were vulnerable

4. Learning And Workplace Rules (Rq4)

Among people learning to code, 39.5% use AI tools daily and 58.2% use them at least weekly (Table 1, 39.5 + 18.7). In the 2026 survey, 52.7% of learners say they use AI code-generation tools as a way to learn. On the workplace side, only 24% say their employer has published approved tools or official guidance on AI. This shows that use of AI is growing faster than the rules for it.

5. Testing The Hypotheses

Hypothesis	Published evidence	Result
H1: Most developers use AI tools regularly	64.8% use daily or weekly (2025); 84% use or plan to use; only 17.2% do not use AI tools at work (2026)	Supported
H2: AI tools save time on coding tasks	Copilot group was 55.8% faster in a controlled experiment; 74% find AI useful for implementation (2026)	Supported
H3: Most developers do not highly trust AI output	Only 3.1% highly trust it (2025); only 6.6% would trust it for important decisions (2026)	Supported
H4: Developers have real concerns about security and accuracy	87% concerned about accuracy and 81% about security and privacy of AI agents (2025); about 40% of tested Copilot programs were vulnerable	Supported

6. Discussion

The data shows a clear gap between use and trust. AI tools are now used by a large majority of developers, and many use them every day, yet only a very small share highly trust the output. Developers seem to use AI as a fast helper and then check the results themselves, which matches the "nearly half trust it when they can verify it" finding of the 2026 survey.The saving of time is real in the controlled experiment, but the security study shows that faster code can also carry hidden weaknesses. This means the time saved in writing should be partly spent on review and testing.For learning, many students and beginners already use AI tools. Since only a small share of workplaces have official AI guidance, colleges and companies need clear rules on how AI may be used.

7. Suggestions For Responsible Use

Read, test and understand every piece of AI-generated code before using it.

- Do not paste private, company or personal data into public AI tools.
- Students should first try to solve a problem on their own and then use AI to check or explain.
- Use AI to learn concepts and not only to get final answers.
- Colleges and companies should publish a clear policy about where and how AI tools may be used.
- Keep human code review and security testing, even for code that looks correct.

8. Limitations

This study uses secondary data, so the author could not choose the questions or the respondents. The Stack Overflow survey is answered by people who choose to take part, mostly developers active in online communities, so it may not represent all developers or all students in India. The 2026

© 2026 IJSRET

figures were read from news reports and should be checked against the survey pages. The surveys used different questions and different groups of people, so their numbers are compared in direction only and are not combined. The Copilot experiments were done on single tasks and on an earlier version of the tool, and AI tools are changing very fast. Future work can collect primary data from students and developers in India through a new survey.

VI. CONCLUSION

Published data from large developer surveys and earlier studies shows that generative AI is now widely used in software development and can save real time on routine coding. At the same time, very few developers highly trust the output, and security tests found many weaknesses in AI-written code. All four hypotheses were supported by the published evidence. The best way forward is to treat AI as a helper and not as a replacement for the developer. People who keep learning the fundamentals and always review what the tool gives them will get the most benefit from it.

REFERENCES

1. Chen, M., Tworek, J., Jun, H., et al. (2021). Evaluating Large Language Models Trained on Code. arXiv:2107.03374. https://arxiv.org/abs/2107.03374
2. Pearce, H., Ahmad, B., Tan, B., Dolan-Gavitt, B., and Karri, R. (2022). Asleep at the Keyboard? Assessing the Security of GitHub Copilot's Code Contributions. IEEE Symposium on Security and Privacy (arXiv:2108.09293). https://arxiv.org/abs/2108.09293
3. Peng, S., Kalliamvakou, E., Cihon, P., and Demirer, M. (2023). The Impact of AI on Developer Productivity: Evidence from GitHub Copilot. arXiv:2302.06590. https://arxiv.org/abs/2302.06590
4. Stack Overflow (2025). 2025 Developer Survey, AI section. https://survey.stackoverflow.co/2025/ai
5. Stack Overflow (2026). 2026 Developer Survey. https://survey.stackoverflow.co/2026/
6. Vaithilingam, P., Zhang, T., and Glassman, E. L. (2022). Expectation vs. Experience: Evaluating the Usability of Code Generation Tools Powered by Large Language Models. CHI Conference on Human Factors in Computing Systems. Extended Abstracts. https://doi.org/10.1145/3491101.3519665
7. Vaswani, A., Shazeer, N., Parmar, N., et al. (2017). Attention Is All You Need. Advances in Neural Information Processing Systems (arXiv:1706.03762). https://arxiv.org/abs/1706.03762
8. IT Brief (October 2026). Stack Overflow survey shows developers trust AI with caveats (report on the 2026 Developer Survey). https://itbrief.co.uk/story/stack-overflow-survey-shows-developers-trust-ai-with-caveats
9. Waters, J. K. (2026). Developers Lean on AI More, But Report Growing Doubts About Accuracy, Stack Overflow Survey Says. ADT Magazine. https://adtmag.com/Blogs/WatersWorks/2026/01/Stack-Overflow-Survey.aspx

© 2026 IJSRET