

OPEN ACCESS

Volume: 5

Issue: 4

Month: October

Year: 2026

ISSN: 2583-7117

Published: 22.10.2026

Citation:

Fiza Nazir Shaikh, Prof. Rekha Kalpesh Kakade "The Impact of Artificial Intelligence in Software Development: Opportunities, Challenges, and Future Directions" International Journal of Innovations in Science Engineering and Management, vol. 5, no. 4, 2026, pp. 206-213.

DOI:

10.69968/ijisem.2026v5i4206-213



This work is licensed under a Creative Commons Attribution-Share Alike 4.0 International License

The Impact of Artificial Intelligence in Software Development: Opportunities, Challenges, and Future Directions

Fiza Nazir Shaikh¹, Prof. Rekha Kalpesh Kakade²

¹Student, Department of B.Voc (Software Development), Prof. Ramkrishna More Autonomous College, Pune, India

²Professor, Department of B.Voc (Software Development), Prof. Ramkrishna More Autonomous College, Pune, India

Abstract

With the rise of Artificial Intelligence (AI), more and more people start implementing AI tools in their software development process. The current study investigates the effect of AI on software development paying special attention to the opportunities of AI-assisted software development, its challenges and perspectives. The aim of the paper is to analyze the ways AI tools may be applied in software development and their limitations. The research includes literature review and survey of 91 participants. The survey addresses the issues of AI tool utilization, productivity increase, the most common benefits, and challenges related to code generated by AI. As a result, 77 respondents (84.6%) claimed to have experienced the increase in productivity. Increased productivity became the most frequently chosen benefit, followed by improved code documentation and learning how to code with the help of AI. However, there were some challenges, including dependence on AI tools, privacy issues, and difficulties with comprehension of the generated code. Thus, there is evidence that AI-assisted software development is possible and allows improving software development process in many aspects, but the output of AI should be carefully evaluated. Human knowledge becomes necessary for requirements understanding, code assessment, security issues detection and decision-making.

Keywords: Artificial Intelligence (AI), Generative AI, Large Language Models (LLMs), Software Development, AI-Assisted Coding, Human-AI Collaboration, Code Generation, Software Testing, Debugging, Software Engineering

INTRODUCTION

How AI is shaping software development AI is transforming how software is built by enabling teams to use AI helps teams write code, identify bugs and prepare test cases, and accelerate software development. In particular, generative AI and large language models are frequently leveraged to assist, as they can also eliminate some duplicate steps within the software development life cycle.

At present there are questions. Code generated by AI can omit specifications, and introduce issues with reliability, security and reliable operation. So human evaluation remains important for quality and validation.

Past work in this space also highlights the role of people in software engineering, requiring expertise, ingenuity, care and oversight.

For this reason AI should not be regarded as an end to the developer, but as an aide to which humans can work with.

The paper examines both gains and pains of using AI in software development: it discusses opportunities to improve productivity, risks associated with AI-written code and the need for human judgment. It is based on prior literature and analysis of survey results. The aim is to demonstrate how both AI and human expertise can help generate value responsibly.

RESEARCH OBJECTIVES

1. To study the impact of AI tools on software development productivity.
2. Elaborate on the benefits and the major drawbacks of work using AI.
3. To give reasons why AI generated code needs done checks.
4. To explore the manner in which human expertise and AI can complement each other in development.

SCOPE OF THE STUDY

This study includes AI aided software development. We can also include generation of code, testing & debugging and review of the output.

LITERATURE REVIEW

The scope of Artificial Intelligence in software work has attracted significant scientific attention. The majority of research papers concentrate on the effectiveness of AI or its application in specific circumstances. Others try to determine its technical limitations and practical implementation barriers. A few authors also try to address the paradigm shifts in developer obligations. Several works also emphasize the importance of proper qualifications for participants in the software development process and the need for human oversight of AI. Each of the following papers addresses some of the issues related to the utilization of AI in software design.

Peres et al. (2020) explore the role of AI in the framework of the fourth industrial revolution. The authors highlight the challenges related to the implementation of this technology in the corporate environment. They suggest that there is a shepherding effect when it comes to adopting such innovations on a large scale. The pilot projects do not necessarily lead to transformations in the realms outside of the realm where they were initially implemented.

Vijayakumar et al. (2026) analyze the peculiarities of AI-native software systems that are based on Large Language Models and utilize AI-assisted programming. The authors describe how natural language prompts can be leveraged to generate code. They also suggest that such an approach could revolutionize the development of software systems.

Schedlberger and Böhm (2024), in turn, focus on the potential of AI in software engineering and its effectiveness in different contexts. They underline the importance of domain-specific knowledge and address such concepts as variability management and model-driven software development. The article also highlights the impact of this emerging field on the development of new curricula and the evolution of the competencies that software engineers need to acquire.

Finally, Madupati (2024) takes a closer look at the role of AI in various stages of software development. The author compiles the benefits that developers could see in terms of increased productivity and efficiency, as well as the ethical concerns and additional limitations that may arise. The areas that should be automatized are coding, testing, debugging, and deployment.

In short, each of the papers listed above provides valuable insights into the different aspects of AI utilization in software development. They also highlight the technical challenges that are associated with the broader adoption of this technology and the necessity of human oversight. By building upon the existing research, the current one aims to further examine the utilization of AI in this field, the sense of its implementation from the perspective of workers, its key benefits, limitations, and the importance of code reviews.

RESEARCH METHODOLOGY

The current research explores the impact of Artificial Intelligence on software development through a literature review and a survey. The information gathered from the study will highlight the benefits, limitations, and human elements of applying artificial intelligence in the field.

Figure 1 below represents the research plan. The process begins with a literature review to identify the related works and existing knowledge about the subject. This step is followed by data collection through a survey to generate primary information. The next stage is data analysis to examine the collected data. Finally, the results of the study will be evaluated and interpreted.

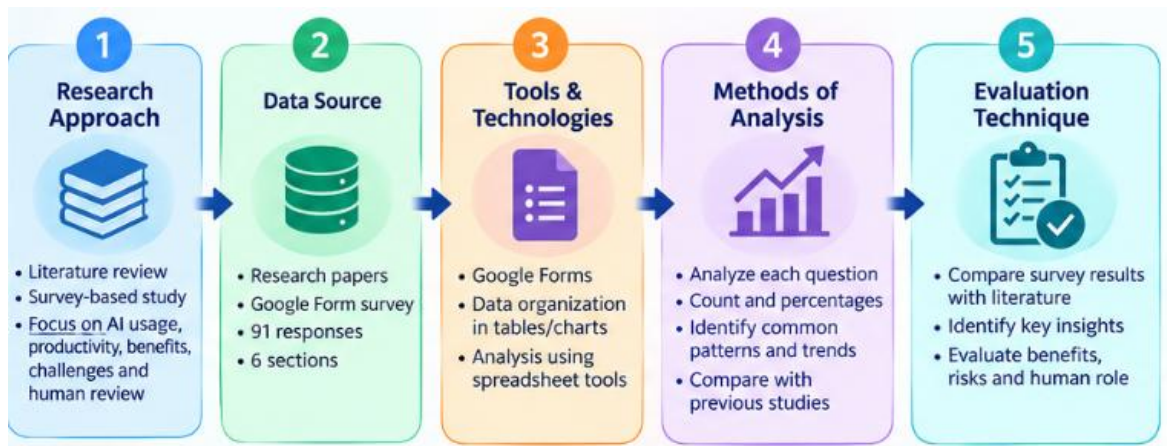


Figure 1 Research Methodology Flowchart

The current research utilizes both primary and secondary sources of data. Articles were reviewed for the literature review section, and questionnaires were distributed online to gather first-hand information from the participants. Google forms were used to create the survey, and 90 respondents completed the questionnaire. The survey consists of six questions seeking information on respondents' data, experience with AI-driven software development, the sense of increased productivity, key benefits, common limitations, and the rationale behind the necessity of human expertise in the development process.

The tools and technologies used for the research include google forms, spreadsheet, and research articles. Google forms were used to collect the data from the participants. Additionally, spreadsheets will be used to organize, count, generate percentages, and visualize the results in graphs. Lastly, research articles were reviewed to generate insights about the research topic.

The survey results will be analyzed qualitatively by identifying themes and patterns in the data. The analysis will begin with describing the data using percentages and ratios to represent the proportion of each set of responses. The analysis will focus on questions 3,4,5,6 whose results will shed light on the participants' perspectives, experience, and opinions on the use of artificial intelligence in software development. The findings will then be compared and contrasted with the results of previous studies to provide a comprehensive analysis of the research topic.

Figure 2 below represents a proposed framework of human-AI collaboration in software development. According to the model, AI can be used to generate code and suggestions, and the recommendations can be reviewed,

tested, and deployed by developers. Developers can also utilize their expertise to continuously test the applications and ensure their functionalities and security features meet the standards.

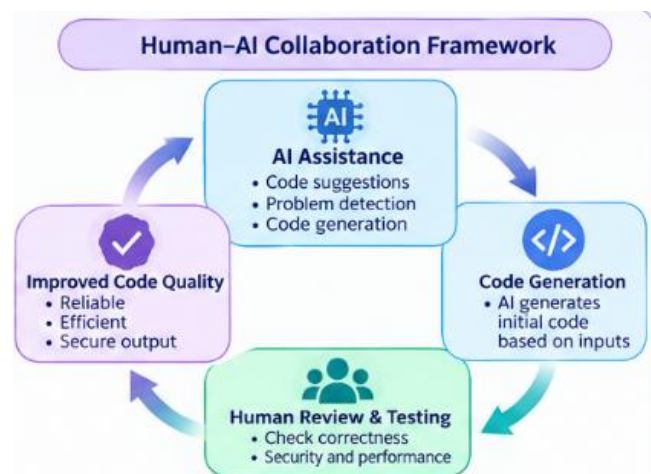


Figure 2 Human AI Collaboration Framework

RESULTS

The results of this study are based on the survey of 91 respondents with various levels of education and professional experience. This survey aimed to identify opinions, experiences, and attitudes towards the application of Artificial Intelligence (AI) tools in the field of software development. Accordingly, the data obtained was analyzed in terms of trends and commonalities in the use of AI, perception of its productivity, benefits and risks associated with the technology. The findings below represent the main outcomes of this study.

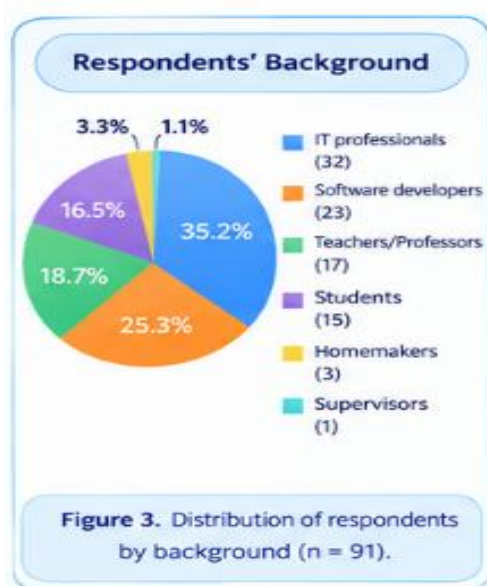


Figure 3 Distribution of respondents by background (n=91)

Table 1. Summary of Survey Results (N = 91)

Survey Category	Main Finding	Respondents	Percentage
AI tool usage	Sometimes	33	36.3%
Productivity improvement	Significant improvement	27	29.7%
Major benefit	Improved productivity	26	28.6%
Major challenge	Over-dependence on AI	22	24.2%
Human review	Yes, in most cases*	28	30.8%

Note. Benefits and challenges were selected by multiple respondents. The human-review finding includes five entries with combined or multiple answers – must be verified before submitting final version.

In terms of the frequency of the use of AI tools, the most common choice was “Sometimes,” by 33 respondents, or 36.3%. Two other common options were “Rarely,” mostly by 17 respondents (18.7%), and “Often,” selected by 15 respondents (16.5%). Moreover, 13 respondents (14.3%) stated that they use AI “Very often,” while the same number chose to say that they “Never” use these tools. According to the survey results, the majority of respondents use AI sometimes with a possibility of occasional frequent use, but not on a daily basis.

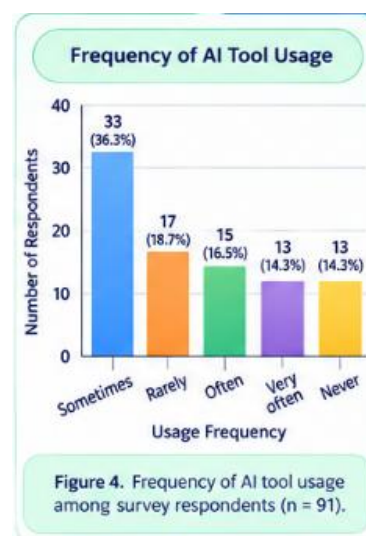


Figure 4 Frequency of AI tool usage among survey respondents (n=91)

As for the choice of options regarding the effect of the use of AI tools on productivity, 27 respondents (29.7%) opted for “Significant improvement,” while 21 respondents (23.1%) selected “Moderate improvement.” Moreover, 15 respondents (16.5%) chose “Slight improvement,” and 14 respondents (15.4%) selected “Very significant improvement.” Finally, 14 respondents (15.4%) stated that they did not notice any improvement. Thus, it is possible to conclude that the vast majority of respondents (77, or 84.6%) reported some level of improvement from the use of AI in their personal experience. It must be noted that these results represent the perception of this effect by the respondents and cannot testify for the universal productivity improvement of the use of AI in any software development task.

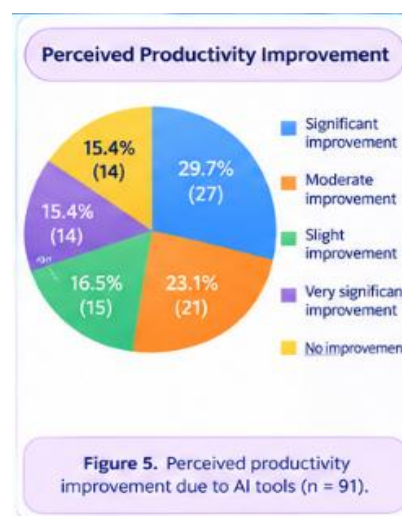


Figure 5 Perceived productivity improvement due to AI tools (n=91)

According to the survey findings, the most common choice among respondents in terms of benefits from the use of AI in software development was “Improved productivity,” selected by 26 respondents (28.6%). Furthermore, 21 respondents (23.1%) chose to learn code documentation, while 20 respondents (22.0%) stated that the major benefit was the opportunity to learn and master programming. In addition, 13 respondents (14.3%) identified debugging and detecting of errors as major benefits of AI tools. The next common choice was “Faster code generation,” selected by 8 respondents (8.8%), and “Reduced repetitive work,” selected by 7 respondents (7.7%). Since multiple choices were allowed, these results represent the selection of several benefits that were identified by the respondents. Thus, the findings suggest that respondents view AI as a beneficial tool that assists in different stages of software development and contributes to various areas of programming.

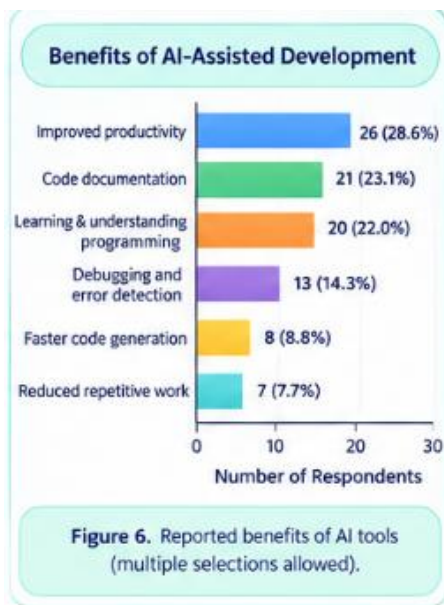


Figure 6 Reported benefits of AI tools (multiple selection allowed)

Furthermore, the survey findings on the challenges of AI-assisted software development point to the majority of respondents who selected “Over-dependence on AI” as the major concern (22 respondents, or 24.2%). Moreover, 21 respondents (23.1%) identified privacy and data-related concerns as significant, while 19 respondents (20.9%) mentioned lack of understanding of AI-generated code. Furthermore, 15 respondents (16.5%) stated that they are concerned about possible security vulnerabilities. Additionally, reduced programming skills (9 respondents, or

9.9%) and code inaccuracy (9 respondents, 9.9%) were identified as major challenges of the use of AI in software development. Thus, major challenges of using AI in software development are the danger of over-dependence on the technology, privacy and data issues, and reduced coding skills due to the reliance on AI tools.

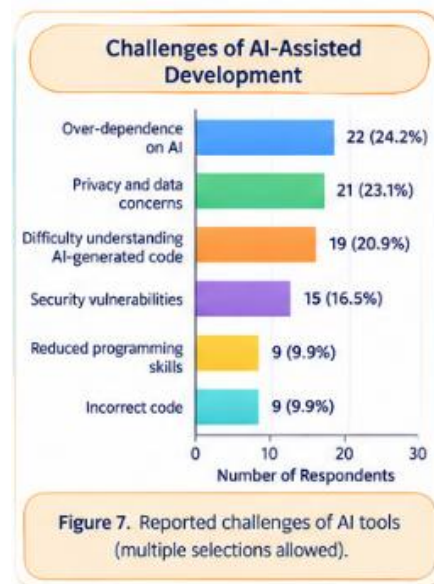


Figure 7 Reported challenges of AI tools (multiple selection allowed)

Finally, the findings on the importance of human review point to a significant concern among the respondents for the current state of affairs in AI-assisted software development. According to the survey results, 28 respondents selected “Yes, in most cases,” 26 respondents chose “Sometimes,” and 23 respondents opted for “Yes, always.” Moreover, 9 respondents selected “No,” and 5 entries included multiple answers. Since multiple selections were allowed, these results may represent the entries with combined answers – should be double-checked before submitting the final version. Nevertheless, the findings suggest that human review is viewed by respondents as a critical aspect of AI-assisted software development that must be considered carefully.

Overall, the survey findings indicate that the majority of respondents acknowledge the potential benefits of AI-assisted software development, particularly in terms of productivity improvement, code documentation, and the learning of programming basics. Nevertheless, the majority of respondents also expressed concerns over the potential risks and dangers of the use of AI, including over-

dependence on the technology, data privacy and security issues, and reduced coding skills with time. Thus, it can be concluded that the use of AI tools in software development must be viewed as a beneficial but limited experience that should complement human expertise in software development rather than replace it. These findings contribute valuable insight into the potential dangers and benefits of AI-assisted software development, although there is still much to be explored before implementing such technology on a large scale.

DISCUSSION

The results of the study provide grounds to believe that Artificial Intelligence (AI) is a beneficial assistant in the field of software development. The results of the survey highlight the software developers' perception of AI's advantages, including productivity improvement, code documentation, learning to code, and error detection. Meanwhile, the results emphasize the developers' concerns regarding the reliance on AI, privacy and security, and code understanding. The analysis of the results allows us to conclude that the impact of AI on software development is conditional and depends on developers' usage and understanding of this technology.

According to the results of the survey, 77 respondents out of 91 (84.6%) stated that they observed an increase in their productivity while using the AI assistants. This result confirms the study's initial hypothesis that AI implementation has a positive impact on the software development in terms of productivity. AI tools allow developers to automatize recurrent tasks and rely on AI assistance while writing code, documenting, learning, and detecting errors. Nevertheless, the mentioned results are indicators of respondents' beliefs in terms of AI's impact on productivity. It is difficult to measure how much AI contributes to productivity and code quality improvement since it depends on a specific task and an individual's skills.

The results of the survey indicate that the majority of the respondents admit that their role in the development process is indispensable despite the growing number of AI implementations in the field of software development. This conclusion can be drawn from the fact that only 20.4% of respondents believe that developers will become obsolete one day. Besides, the article by Schedlberger and Böhm (2024) provides evidence of the insurmountable gap between human performance and that of artificial intelligence in terms of professional development and workforce transformation.

The results of the survey demonstrate that the majority of respondents believe that the main limitation of using AI assistants in software development is its potential dysfunctionality. Other limitations include privacy concerns and a lack of explainability of AI's functioning principles. Privacy concerns are associated with the possibility of personal data being used by companies for marketing purposes, whereas the absence of explainability is related to the black-box nature of most AI algorithms. The dysfunctionality of AI can be attributed to the lack of protective measures or insufficient training data. However, Peres et al. (2020) mention that the main barriers to AI implementation in software development are technical, including AI programming, model, and validation. Nonetheless, it is essential to distinguish between industrial AI and AI-driven software development.

The emergence of large language models and AI-powered coding assistants alters the role of developers in terms of responsibilities and skills. The paper by Vijayakumar et al. (2025/2026) draws attention to the paradigm shift in the field of software development in terms of AI-native development. According to the article, developers are no longer coding at the lowest level but designing while relying on AI agents to execute commands. Therefore, with the growing number of AI implementations in software development, the role of a developer is in constant evolution. As such, it is essential to reconsider the significance of coding in software development.

The results of the study provide grounds to believe that students and professionals must consider AI as a beneficial assistant in software development. Nevertheless, students should rely on AI while practicing coding to improve their writing skills while learning the fundamental principles of software development. In addition, students need to hone their coding abilities through practice and code debugging to gain professional competencies in the sphere. From a professional perspective, software developers should utilize AI as an assistant while writing code to improve their productivity. At the same time, professional developers should implement peer reviews to ensure the quality of their code. The article by Madupati (2024) draws attention to AI's role in coding, testing, debugging, and deployment in the context of a software development lifecycle. These educational and professional implications are essential for achieving a sufficient balance between automation and human expertise in terms of responsibility and performance.

CONCLUSION

The results of the survey and published articles provide evidence of the fact that AI is a beneficial but limited technology in terms of software development. The results of the survey represent the respondents' beliefs about the role of AI in software development. The opinions expressed by the respondents are aligned with the information presented in the chosen articles. The analysis of the articles highlights the necessity of using AI as an assistant while developing software in order to improve code quality and productivity. The current study was based on survey results and secondary data. Therefore, it is essential to conduct further studies to explore AI's benefits and limitations in greater detail. In particular, the future research on the topic of AI-driven software development can provide more insight into the benefits of using specific AI assistants in terms of developer productivity and code quality.

ACKNOWLEDGEMENT

I would like to acknowledge my profound gratitude to my project guide and faculty members for their insightful guidance, support, and motivation that led me to the completion of this research work on "The Impact of Artificial Intelligence in Software Development: Opportunities, Challenges, and Future Directions."

I am obliged to my college for giving me an opportunity to pursue this research work and increase my knowledge about Artificial Intelligence and its application in the field of software development.

I would also like to extend my thanks to all the respondents who participated in my survey and provided me with their respective feedbacks and opinions.

My special thanks also goes out to the authors and researchers whose published works helped me with collating necessary literature review for this research paper.

Lastly, I would like to show my appreciation to all those people who directly or indirectly contributed to my research work.

REFERENCES

- [1]. Allamanis, M., Barr, E. T., Devanbu, P., & Sutton, C. (2018). A survey of machine learning for big code and naturalness. *ACM Computing Surveys*, 51(4), Article 81. <https://doi.org/10.1145/3212695>
- [2]. Austin, J., Odena, A., Nye, M., Bosma, M., Michalewski, H., Dohan, D., Jiang, E., Cai, C., Terry, M., Le, Q. V., & Sutton, C. (2021). *Program synthesis with large language models*. arXiv. <https://arxiv.org/abs/2108.07732>
- [3]. Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H. P. de O., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., et al. (2021). *Evaluating large language models trained on code*. arXiv. <https://arxiv.org/abs/2107.03374>
- [4]. Fan, A., Gokkaya, B., Harman, M., Lyubarskiy, M., Sengupta, S., Yoo, S., & Zhang, J. M. (2023). Large language models for software engineering: Survey and open problems. arXiv. <https://arxiv.org/abs/2310.03533>
- [5]. Hou, X., Zhao, Y., Liu, Y., Yang, Z., Wang, K., Li, L., Luo, X., Lo, D., Grundy, J., & Wang, H. (2024). Large language models for software engineering: A systematic literature review. *ACM Transactions on Software Engineering and Methodology*. <https://arxiv.org/abs/2308.10620>
- [6]. Jimenez, C. E., Yang, J., Wettig, A., Yao, S., Pei, K., Press, O., & Narasimhan, K. (2024). SWE-bench: Can language models resolve real-world GitHub issues? *International Conference on Learning Representations*. <https://arxiv.org/abs/2310.06770>
- [7]. Li, R., Allal, L. B., Zi, Y., Muennighoff, N., Kocetkov, D., Mou, C., Marone, M., Akiki, C., Li, J., Chim, J., et al. (2023). *StarCoder: May the source be with you!* arXiv. <https://arxiv.org/abs/2305.06161>
- [8]. Nijkamp, E., Pang, B., Hayashi, H., Tu, L., Wang, H., Zhou, Y., Savarese, S., & Xiong, C. (2023). *CodeGen: An open large language model for code with multi-turn program synthesis*. International Conference on Learning Representations. <https://arxiv.org/abs/2203.13474>
- [9]. Pearce, H., Ahmad, B., Tan, B., Dolan-Gavitt, B., & Karri, R. (2022). Asleep at the keyboard? Assessing the security of GitHub Copilot's code contributions. *2022 IEEE Symposium on Security and Privacy*. <https://doi.org/10.1109/SP46214.2022.9833571>
- [10]. Peng, S., Kalliamvakou, E., Cihon, P., & Demirer, M. (2023). *The impact of AI on developer*

- productivity: Evidence from GitHub Copilot.* arXiv. <https://arxiv.org/abs/2302.06590>
- [11]. Peres, R. S., Jia, X., Lee, J., Sun, K., Colombo, A. W., & Barata, J. (2020). Industrial artificial intelligence in Industry 4.0: Systematic review, challenges and outlook. *IEEE Access*, 8, 220121–220139. <https://doi.org/10.1109/ACCESS.2020.3042874>
- [12]. Rozière, B., Gehring, J., Gloeckle, F., Sootla, S., Gat, I., Tan, X. E., Adi, Y., Liu, J., Remez, T., Rapin, J., et al. (2023). *Code Llama: Open foundation models for code.* arXiv. <https://arxiv.org/abs/2308.12950>
- [13]. Sandoval, G., Pearce, H., Nys, T., Karri, R., Garg, S., & Dolan-Gavitt, B. (2023). Lost at C: A user study on the security implications of large language model code assistants. *32nd USENIX Security Symposium*, 2205–2222. <https://www.usenix.org/conference/usenixsecurity23/presentation/sandoval>
- [14]. Svyatkovskiy, A., Deng, S. K., Fu, S., & Sundaresan, N. (2020). IntelliCode Compose: Code generation using transformer. *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering.* <https://doi.org/10.1145/3368089.3417058>
- [15]. Vaithilingam, P., Zhang, T., & 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>
- [16]. Zhang, Q., Fang, C., Xie, Y., Zhang, Y., Yang, Y., Sun, W., Yu, S., & Chen, Z. (2024). *A survey on large language models for software engineering.* arXiv. <https://arxiv.org/abs/2312.15223>
- [17]. Zhang, Z., Chen, C., Liu, B., Liao, C., Gong, Z., Yu, H., Li, J., & Wang, R. (2023). *Unifying the perspectives of NLP and software engineering: A survey on language models for code.* arXiv. <https://arxiv.org/abs/2311.07989>
- [18]. Schedlberger, L.-M., & Böhm, K. (2024). The essential human contribution and its impact on future workforce development in the field of software engineering. <https://doi.org/10.1145/3729434.3729444>