DOI: https://doi.org/10.48009/3_iis_2025_2025_111

From code to concern: A demographic analysis of AI challenges and ethics in IT development

Alan Peslak, *Penn State University, arp14@psu.edu*
Lisa Kovalchick, *Pennsylvania Western University, kovalchick@pennwest.edu*

Abstract

This study explores the relationships between artificial intelligence (AI) usage in software development and the issues of confidence in code generated, ethical issues associated with AI usage, technical challenges of AI code development, and demographic differences related to these issues. AI is currently used in software development for a variety of purposes including coding, debugging, and code optimization and is generally viewed as a useful tool for programming. There are, however, significant issues associated with these usages. Our study uses the 2024 Stack Overflow Annual Developer Survey to explore AI adoption patterns, developer trust level, perceived technical and ethical challenges, and variations in areas by key demographic variables. We find important areas of concern both overall and by demographic category. These findings are relevant to both practitioners, researchers, and society. This paper contributes to the goal of optimization of AI productivity enhancement while addressing trust and ethical concerns.

Keywords: AI development tools, software developers, AI ethical issues, AI adoption patterns

Introduction

In recent years, the landscape of artificial intelligence (AI) has exploded, with AI involved in some aspects of the daily life of most people. This explosion began in late 2021 and 2022 with the release of GitHub CoPilot and ChatGPT, respectively. Since then, many other AI-powered technologies have made an appearance. AI-powered technologies quickly made their way into the software industry with the hopes of these AI tools assisting in the software development process by automatically generating code, providing AI-powered design, and automating the testing process. While AI tools used in the development process promise some amazing benefits such as increasing productivity and developer efficiency by automating mundane tasks and assisting in brainstorming and troubleshooting, these tools also have their drawbacks. These drawbacks are often substantial, ranging from insecure code to ethical concerns and trust issues.

Nonetheless, AI is here to stay as “organizations are beginning to create the structures and processes that lead to meaningful value from Generative AI” (The state of AI, 2025). According to GitHub’s 2024 The State of the Octoverse report, “in 2024 there was a 59% surge in the number of contributions to generative AI projects on GitHub and a 98% increase in the number of projects overall” (2024). Therefore, it is important that we understand the adoption pattern of AI tools in the software development process so that concerns can be addressed, and their benefits can continue to enhance our everyday lives.

The goal of this study is to understand the use of AI tools in the development process and the issues associated with the use of such tools. Specifically, this study sought to answer the following research questions.

RQ1. *What is the current usage of AI tools in the development process and what is the sentiment towards their usage by software developers?*
RQ2. *What are the major challenges regarding the use of AI tools in the development process, as perceived by software developers and how do demographic factors (age, education, experience, company size) influence these perceptions*?
RQ3. *What are the major AI ethical issues, as perceived by software developers and how do demographic factors (age, education, experience, company size) influence these perceptions?*
RQ4: *What employment groups and roles report the highest and lowest usage rate of AI tools in the development process?*

Literature Review

Although AI tools used for software development have only been in existence for a few years, several studies have been published regarding their usage and the benefits and drawbacks to their use. A review of recent literature suggests that software developers have mixed feelings when it comes to employing such tools during the software development process. Drawbacks include concerns about ethics, security, and trust, while benefits include efficiency and the freeing of developer's time to focus on more complex tasks.

Several authors have examined the reasons that software developers turn to AI tools during the development process and their perceptions of using these tools. Scoccia (2023) studied early adopters of ChatGPT, who were using it for automatic code generation, to learn their perceptions of using it and found mixed perceptions. While these early adopters recognized the positive impact it could have on the software development process, they also highlighted concerns such as trust in the code that it produced. Ge and Wu (2023) analyzed the reasons that influence professional software developers to turn to ChatGPT for assistance with fixing bugs in software and found that influence from peers, data security issues, and trust are a few of the factors considered. Das, Mondal, and Roy (2024) examined developer's use of ChatGPT to fix issues within their tracking systems and found that while it is often used for brainstorming, the code generated is often replaced with human generated code.

In the spring of 2022, Ernst and Bavota (2022) were among the first to point out possible challenges with AI-driven development environments including copyright and licensing issues, dataset quality, and sociotechnical questions; many of the challenges that they realized are still voiced as concerns among developer's today. Khan et al. (2022) examined ethics in AI by conducting a systematic literature review to find the ethical principles and challenges related to AI.

Several authors have examined both the benefits and disadvantages of using AI tools in the development process. In late 2023, CodeSignal (2024) surveyed over 1,000 developers and found that the majority of them cited a boost to productivity as a benefit to the use of AI tools, while a majority of them also cited the correctness of code as a drawback to the use of such tools. Zhang, Liang, Zhou, Ahmad, and Waseem (2023) utilized data from Stack Overflow and GitHub discussions to analyze the advantages, limitations, and disadvantages involving the use of GitHub Copilot. Their study found the primary benefit of the use of GitHub CoPilot to be the usefulness of the generated code and the biggest challenge to be that of integrating the suggested code into a current project. Odeh, Odeh, and Mohammed (2024) examined AI methods used for automatic code generation to understand their applications, advantages, and disadvantages. They found many of the same factors seen in other studies, including accuracy and correctness of the generated code. Jin (2024) examined the use of AI in agile software engineering development; providing a summary of the innovations, such as increased productivity, and problems, such as data privacy concerns, related to its use.

The correctness and security of the code generated by AI tools continues to be a concern among software developers; several authors have studied these topics. In 2023 Snyk surveyed 537 people working in technology roles to create the 2023 AI Code Security Report (2023); this report found that over half of the survey respondents reported that “AI coding tools commonly generate insecure code suggestions”. Negri-Ribalta, Geraud-Stewart, Sergeeve, and Lenzini (2024) examined the issue of secure coding as it related to code generated by AI and found that such code cannot be relied on to be free of security flaws. Jaworski and Piotrkowski conducted a survey on the use of Github Copilot for code generation and found that although the software developer’s attitudes towards it were overall positive, most were unwilling to use it stating security concerns. Dakhel, et al. (2023) studied Github Copilot’s ability to generate “correct and efficient solutions to fundamental algorithmic problems”. They found that Github Copilot was usually able to generate solutions to such problems; however, sometimes the generated solutions contained errors. They also discovered that in comparison to the solutions generated by human developers for the same set of problems, the human developer solutions were often more correct; however, the errors introduced in the Github Copilot solutions were easier to repair than their human developer counterparts.

Several studies have examined trust issues related to the use of AI tools in the development process. Wang, Cheng, Ford, and Zimmermann (2023) studied the level of trust that developers have with AI code generation tools and the challenges faced in building such trust. Pink, Quilty, Grundy, and Hoda (2024) researched the issue of trust, as it relates to the use of AI in the software industry by conducting interviews of practitioners within the industry. Chang, Wang, Zimmermann, and Ford (2024) analyzed the role online communities play in influencing a developer’s trust in AI coding tools.

Other studies have focused on the benefits of using AI tools in the software development process. In 2023, Shani and GitHub Staff (2023) surveyed 500 U.S.-based developers at enterprise companies and learned that over 4 out of 5 of them believe that AI coding tools will increase collaboration within their team and 70% of them believe that these tools will result in benefits including better code quality and faster completion time. Grewal, Lu, Nadi, and Bezemer (2024) studied the use of ChatGPT code generation in open-source GitHub projects and found that the majority of this code remains unchanged when added to a project, suggesting minimal effort required by the developer, when utilizing such code. Ulfsnes, Moe, Stray and Skarpen (2024) discovered that the use of automatically generated code generated by ChatGPT has created an adjustment in software developer’s workflow. These developers are now able to work more efficiently and accelerate their learning process by utilizing automatically generating code for repetitive coding tasks.

Methodology

Data for this study was obtained from the 2024 Stack Overflow Annual Developer Survey, which was conducted in May and June 2024. As reported by Stack Overflow, 65,437 developers responded to this survey from 185 countries (2024). SPSS 30 was used to analyze the results using a variety of statistical measurements including regression analysis and frequency tables. Chat GPT-4 was used in facilitating the literature review and assisting in sentence and narrative editing. This is accepted in science journals if it is acknowledged (Gaggioli, 2023).

Results

The Stack Overflow data used for this study was collected from all over the world (185 countries). The data was self-reported, and the survey respondents represented a large variety of ages (from under 18 to over 65 years). These respondents have obtained various levels of education from those with primary/elementary level schooling to those with a professional degree (JD, MD, Ph. D, Ed. D, etc.). Both employed and unemployed people responded to the survey.

Research Question 1: What is the current usage of AI tools in the development process and what is the sentiment towards their usage by software developers?
Before we specifically explore the usage of AI tools in the development process, we first examine software developers' sentiment towards the usage of AI tools in the development process. To collect this data, the survey asked the question "How favorable is your stance on using AI tools as part of your development workflow?" Table 1 displays the results of this question. Approximately 72% of software developers report a favorable or very favorable sentiment regarding the use of AI tools in the development process. This suggests a strong acceptance or optimism toward such tools. Approximately 18.7% are indifferent, indicating that a sizable portion of respondents neither strongly support nor oppose the use of AI tools in the development process. The 6.4% who view the use of AI tools in the development process unfavorably suggest that concerns about the use of these tools are not strong in public sentiment.

Table 1. Sentiment Related to the Use of AI Tools in the Development Process

	Frequency	Percent
	12	0.0%
Very favorable	10848	23.6%
Favorable	22167	48.3%
Indifferent	8564	18.7%
Unsure	1357	3.0%
Unfavorable	2381	5.2%
Very unfavorable	556	1.2%
Total	**45885**	**100.0%**

We now examine the usage of AI tools in the development process. To collect this data, the survey asked the question "Do you currently use AI tools in your development process?" The results of this question are displayed in Table 2. Overall, 58% of the survey respondents indicated that they use AI tools in the development process and another 13% indicated that they plan to use them soon. These results reinforce the results displayed in Table 1 and imply that a positive sentiment translates into usage.

Table 2. AI Tool Usage in the Development Process, by Survey Respondents

	Frequency	Percent
NA	4530	6.9
No, and I don't plan to	14837	22.7
No, but I plan to soon	8408	12.8
Yes	37662	57.5
Total	**65437**	**100.0**

Research Question 2: What are the major challenges regarding the use of AI tools in the development process, as perceived by software developers and how do demographic factors (age, education, experience, company size) influence these perceptions?

To answer this question, we studied only software developers who are using AI tools in the development process, since these are the people that are facing the challenges when using this technology. Since the responses are binary, we use the mean to represent the percentage of developers who report that they have encountered a specific challenge. As depicted in Table 3, 54% of survey respondents do not trust the output or answers provided by AI tools. This is surprising given the favorable sentiment and high usage found when exploring our first research question. Over half of survey respondents also report the AI tool's lack context of codebase, internal architecture, and/or company knowledge as a challenge. Developers are much less concerned about other issues, as only 26% note security as a challenge and 25% note lack of training. Other challenges have even less importance.

Table 3. AI Tool Challenges, as Reported by Survey Respondents Who Utilize AI Tools

	N	Minimum	Maximum	Mean	Std. Deviation
Don't trust the output or answers	37662	0	1	0.54	0.499
AI tools lack context of codebase, internal architecture, and/or company knowledge	37662	0	1	0.52	0.5
We don't have the right policies in place to reduce security risks	37662	0	1	0.26	0.437
Lack of proper training and education on new tools	37662	0	1	0.25	0.433
Not everyone uses them	37662	0	1	0.21	0.406
NA	37662	0	1	0.19	0.389
They create more work (more code/PRs to review, etc.)	37662	0	1	0.11	0.307
Lack of executive buy-in	37662	0	1	0.09	0.291
Other (please specify):	37662	0	1	0.05	0.216
None of the above	37662	0	0	0	0
Valid N (listwise)	37662				

Delving deeper into this research question, we explored the reported challenges presented by the use of AI tools in the development process based on demographic variables. These results are displayed below in Tables 4 through 7. The regression analysis displayed in Table 4 reports that age is a significant variable affecting trust with younger age groups less frequently noting trust as a concern. The size of the organization is only marginally significant (p < .09) with larger companies more concerned about trust than smaller companies. The two remaining independent variables provided surprising results. Lower education levels correlated significantly with higher distrust (p < .026). The more years someone is coding also resulted in higher distrust. When all independent variables are at zero, the baseline level of distrust in AI-generated output or answers is **0.622**, suggesting a moderate inherent skepticism toward AI systems.

Table 4. Regression Analysis of the 'Don't Trust the Output or Answers' Response

Model		Unstandardized Coefficients		Standardized Coefficients		
		B	Std. Error	Beta	t	Sig.
1	(Constant)	.622	.016		39.410	<.001
	Age	-.030	.005	-.061	-6.319	<.001
	Org Size	.002	.001	.010	1.684	.092
	Ed Level	-.005	.002	-.014	-2.232	.026
	Years Coding	.005	.000	.103	10.812	<.001

The regression analysis displayed in Table 5 reports that all variables significantly affect the challenge of 'AI tools lack context of codebase, internal architecture, and/or company knowledge'. Younger software developers are less likely to see this as an issue, while larger companies and less educated software developers are more likely to see this as an issue. The longer the software developer has been coding the more importance they place on this challenge.

Table 5. Regression Analysis of the 'AI Tools Lack Context of Codebase, Internal Architecture, and/or Company Knowledge' Response

Model		Unstandardized Coefficients		Standardized Coefficients		
		B	Std. Error	Beta	t	Sig.
1	(Constant)	.656	.016		41.360	<.001
	Age	-.035	.005	-.072	-7.519	<.001
	Org Size	.007	.001	.033	5.420	<.001
	Ed Level	-.009	.002	-.024	-3.898	<.001
	Years Coding	.003	.000	.053	5.616	<.001

The regression analysis displayed in Table 6 reports that organizational size, education level, and coding experience play statistically significant roles in shaping perceptions of security policy adequacy in addressing the usage of AI tools in the development process. As shown in this regression analysis, the challenge of 'we don't have the right policies in place to reduce security risks' was a significant variable for organizational size, with larger organizations concerned. It was also significant by education level, where higher education resulted in higher concern. The intercept has a coefficient of **0.204** with a **highly significant p-value (<.001)**, suggesting that when all independent variables are at zero, the dependent variable—*belief that the organization does not have the right policies to reduce security risks*—is at a baseline level of **0.204**.

Table 6. Regression Analysis of the 'We Don't Have the Right Policies in Place to Reduce Security Risks' Response

Model		Unstandardized Coefficients		Standardized Coefficients		
		B	Std. Error	Beta	t	Sig.
1	(Constant)	.204	.014		14.169	<.001
	Age	-.002	.004	-.004	-.393	.694
	Org Size	.017	.001	.086	14.190	<.001
	Ed Level	.006	.002	.017	2.752	.006
	Years Coding	-.002	.000	-.040	-4.185	<.001

The regression analysis displayed in Table 7 reports that ‘lack of proper training and education on new tools’ was significant and positive for older individuals and large organizations.

Table 7. Regression Analysis of the ‘Lack of Proper Training and Education on New Tools’ Response

		Unstandardized Coefficients		Standardized Coefficients		
Model		**B**	**Std. Error**	**Beta**	**t**	**Sig.**
1	**(Constant)**	.196	.014		13.877	<.001
	Age	.015	.004	.035	3.627	<.001
	Org Size	.009	.001	.045	7.434	<.001
	Ed Level	-.003	.002	-.008	-1.360	.174
	Years Coding	-.001	.000	-.014	-1.502	.133

Research Question 3: What are the major AI ethical issues, as perceived by software developers and how do demographic factors (age, education, experience, company size) influence these perceptions? To collect this data, the survey asked the question “Which AI ethical responsibilities are most important to you?” Survey respondents were told to select all of the AI responsibilities that apply. Table 8 displays the results of this question. As reported in this table, the most common ethical issue with AI was ‘circulating of misinformation or disinformation’ with a full 71% survey respondents expressing this as an issue. This was followed by 58% noting ‘missing or incorrect attribution for sources of data’. Other ethical issues that few were concerned with include ‘biased results that do not represent diverse viewpoints’ and ‘energy demand.

Table 8: Major Ethical Issues Involving AI, as Perceived by Software Developers

	N	**Minimum**	**Maximum**	**Mean**	**Std. Deviation**
Circulating misinformation or disinformation	37662	0	1	0.71	0.454
Missing or incorrect attribution for sources of data	37662	0	1	0.58	0.494
Biased results that do not represent diverse viewpoints	37662	0	1	0.44	0.497
Imitating a person's likeness	37662	0	1	0.33	0.471
Energy demand	37662	0	1	0.32	0.468
Replacing jobs without options for new employment opportunities	37662	0	1	0.31	0.461
NA	37662	0	1	0.11	0.307
Other (please specify):	37662	0	1	0.06	0.233
Valid N (listwise)	37662				

Delving deeper into this research question, we explored the reported major ethical issues based on demographic variables. These results are displayed below in Tables 9 through 11. The regression analysis displayed in Table 9 reports that the concern of ‘circulating misinformation or disinformation’ was significantly different based on age, with younger developers expressing concern and organizational size with larger organizations expressing concern.

Table 9. Regression Analysis of the ‘Circulating Misinformation or Disinformation’ Concern

Model		Unstandardized Coefficients		Standardized Coefficients		
		B	Std. Error	Beta	t	Sig.
1	(Constant)	.716	.014		51.655	<.001
	Age	-.020	.004	-.046	-4.842	<.001
	Org Size	.009	.001	.046	7.545	<.001
	Ed Level	-.001	.002	-.002	-.331	.741
	Years Coding	.004	.000	.095	9.961	<.001

The regression analysis displayed in Table 10 reports that the concern of ‘missing or incorrect attribution for sources of data’ was a concern for younger developers, those with less years of coding, and larger organizations.

Table 10. Regression Analysis of the ‘Missing or Incorrect Attribution for Sources of Data’ Concern

Model		Unstandardized Coefficients		Standardized Coefficients		
		B	Std. Error	Beta	t	Sig.
1	(Constant)	.639	.016		40.843	<.001
	Age	-.028	.005	-.058	-6.026	<.001
	Org Size	.005	.001	.025	4.103	<.001
	Ed Level	-.003	.002	-.008	-1.321	.186
	Years Coding	.004	.000	.068	7.164	<.001

Table 11 displays the regression analysis for the concern of ‘biased results that do not represent diverse viewpoints’. As shown in this table, this issue was only a major concern with larger organizations.

Table 11. Regression Analysis of the ‘Biased Results that Do Not Represent Diverse Viewpoints’ Concern

Model		Unstandardized Coefficients		Standardized Coefficients		
		B	Std. Error	Beta	t	Sig.
1	(Constant)	.448	.016		28.054	<.001
	Age	.008	.005	.015	1.612	.107
	Org Size	.003	.001	.013	2.137	.033
	Ed Level	-.003	.002	-.009	-1.407	.159
	Years Coding	-.001	.000	-.019	-1.945	.052

Research Question 4: What employment groups and roles report the highest and lowest usage rate of AI tools in the development process?

As shown in Table 12, most members of all the groups of software developers report using AI tools in the development process. Those who classify themselves as employed, full-time independent contractors, freelancers, or self-employed individuals report the highest usage of AI tools in the development process with 66.43% of the members of this group indicating use. However, all other groups report a usage of at least 55.78% of the members of the group. This indicates that the use of AI tools in the development process is becoming a norm.

Table 12. Usage of AI Tools Among Employment Groups

Employment Status and AI Usage	Total in Category	Use AI	% Use AI
Employed, full-time	22036	17005	0.564432
Independent contractor, freelancer, or self-employed	2801	2046	0.577883
Student, full-time	2752	1957	0.584413
Employed, full-time; Independent contractor, freelancer, or self-employed	2363	1194	0.664324
Not employed, but looking for work	1306	1035	0.557881

Tables 13 and 14 present a breakdown of the usage of AI tools across various developer and professional roles. Table 13 presents the five roles with the highest AI tool adoption rates. The Developer, AI group has the highest adoption rate of AI tools at 80.11%, which is expected as AI development naturally involves AI tools. The Developer Advocate and Data Scientist/Machine Learning Specialist groups also show high rates of adoption of AI tools, likely due to their roles in AI and machine learning model building and automation. The Blockchain Developers group shows a strong adoption of AI tools, likely due to AI's role in security, automation, and smart contract development. The Executives group rounds out the top five, indicating high adoption rates of AI tools, likely for business intelligence, decision support, and automation.

Table 13. The Five Roles with the Highest AI Tool Adoption Rates

Role	% Use AI
Developer, AI	80.11%
Developer Advocate	74.29%
Data Scientist / Machine Learning Specialist	70.70%
Blockchain Developer	70.64%
Senior Executive (C-Suite, VP, etc.)	69.30%

Table 14 presents the four roles with the lowest AI tools adoption rates. Two of the groups reporting low usage are the Hardware Engineers and Embedded Developers, respectively; this is not surprising as these groups are likely to utilize AI tools less due to the hands-on nature of hardware and firmware work. The Database Administrators group also reports low usage; this group may be less involved as AI-related automation is less likely to pertain to this field. The final group reporting low usage is the Scientists group; this group may not use AI as frequently due to the nature of their work, except in instances of computational or data-heavy disciplines.

Table 14. The Four Roles with the Lowest AI Tool Adoption Rates

Role	% Use AI
Hardware Engineer	39.00%
Database Administrator	41.52%
Embedded Applications Developer	42.58%
Scientist	43.37%

Table 15 reports AI tool usage across major developer roles. The Front-End Developers role reports the highest adoption rate of usage of AI tools at 66.98%, this is likely because members of this group are using AI-powered design tools, automated testing, and performance optimizations. The Full-Stack Developers and Mobile Developers roles report the next highest rates of adoption of AI tools at 63.82% and 63.48%, respectively, reflecting AI's role in code generation and debugging. The Cloud Infrastructure Engineers role utilizes AI at a slightly lower rate than the aforementioned roles, possibly because this role has more of a focus on automation and monitoring.

Table 15. AI Tool Usage in Major Developer Roles

Role	% Use AI	Total in Category
Full-Stack Developer	63.82%	18,260
Back-End Developer	60.42%	9,928
Front-End Developer	66.98%	3,349
Mobile Developer	63.48%	2,021
Cloud Infrastructure Engineer	59.46%	634

Table 16 reports on AI tool usage by those with ‘Student’ and ‘Educator’ roles. Not surprisingly, students have a relatively high AI tool usage rate at 60.64%, which suggests AI is becoming a key learning tool. While educators exhibit a lower usage rate at 53.52%, possibly reflecting slower adoption in traditional teaching methodologies.

Table 16. AI Tool Usage by Student and Educator Roles

Role	% Use AI	Total in Category
Students	60.64%	5,102
Educators	53.52%	355

As shown in Table 17, those in product and project manager roles are using AI tools moderately, likely in areas such as project planning, documentation, and analytics.

Table 17. AI Tool Usage by Project and Product Management Roles

Role	% Use AI	Total in Category
Product Managers	55.86%	290
Project Managers	50.48%	418

Conclusion

In this study we examined software developers’ use of AI tools in the development process. It explored the relationships between AI usage in software development and the issues of confidence in code generated, ethical issues associated with AI usage, technical challenges of AI code development, and demographic differences related to these issues. We also reported on the sentiment towards these tools and presented adoption rates of AI tools in the development process by employment groups and roles.

We found that the use of AI tools in the software development process is becoming more mainstream and the use of them is seen favorably in the eyes of most developers. However, these tools still have perceived drawbacks including trust in the correctness of the answers provided by AI and security concerns.

The data produced through this survey was self-reported by people who consider themselves developers. This can be viewed as a drawback, since the questions could be interpreted differently by different respondents, leading to inaccurate information. Also, survey respondents may unintentionally provide inaccurate answers due to bias or the desire to provide socially desirable answers. As the use of AI continues to grow, the findings of this study may be used to assist in addressing software developer's concerns leading to more widespread adoption and use of AI tools during the development process. The implications of this research may also assist future researchers in understanding the sentiment of AI in the early years of its use and the initial technical challenges and ethical concerns raised by its use.

Acknowledgments

Artificial intelligence tools, including large language models and natural language processing software, were used during the creation of this paper to support literature review, assist in data analysis, and aid in writing and composition review. All sources referenced were independently verified, and all content generated or suggested by AI was critically reviewed, edited, and approved by the author to ensure accuracy, integrity, and alignment with academic standards.

References

Cheng, R., Wang, R., Zimmermann, T., & Ford, D. (2024). "It would work for me too": How Online Communities Shape Software Developers' Trust in AI-Powered Code Generation Tools. ACM Transactions on Interactive Intelligent Systems. 14(2), 1–39. https://doi.org/10.1145/3651990 .

CodeSignal (2024). Trends Report Developers & AI Coding Assistant Trends. CodeSignal. Retrieved May 11, 2025, from https://codesignal.com/report-developers-and-ai-coding-assistant-trends/

Das, J.K., Mondal, S., & Roy, C.K. (2024). Investigating the Utility of ChatGPT in the Issue Tracking System: An Exploratory Study. ArXiv, abs/2402.03735.

Daigle, K. & GitHub Staff (2024). The State of the Octoverse. GitHub. Retrieved on May 11, 2025, from https://GitHub.blog/newsinsights/research/the-state-of-open-sourceand-ai/

Dakhel, A., Majdinasab, V., Nikanjam, A., Khomh, F., Desmarais, M., & Jiang, Z. (2023). GitHub Copilot AI pair programmer: Asset or Liability?, Journal of Systems and Software, 203.

Ernst, N., & Bavota, G. (2022). AI-Driven Development Is Here: Should You Worry? IEEE Software, 39(2) 106-110.

Gaggioli A. (2023). Ethics: Disclose Use of AI in Scientific Manuscripts. Nature. 614 (7948): 413

Ge, H., & Wu, Y. (2023). An Empirical Study of Adoption of ChatGPT for Bug Fixing among Professional Developers. Innovation & Technology Advances, 1(1), 21–29. https://doi.org/10.61187/ita.v1i1.19

GitHub (2024). GitHub Developer Survey. GitHub. Retrieved May 11, 2025, from https://GitHub.blog/newsinsights/research/survey-reveals-ais-impacton-the-developer-experience/#developerswant-more-opportunities-to-upskill-anddrive-impact

Grewal, B., Lu, W., Nadi, S., & Bezemer, C. (2024). Analyzing Developer Use of ChatGPT Generated Code in Open Source GitHub Projects. 21st International Conference on Mining Software Repositories.

Jaworski, M., & Piotrkowski, D. (2023). Study of software developers' experience using the Github Copilot Tool in the software development process. ArXiv, abs/2301.04991.

Jin, Ziyao (2024). Integrating AI into Agile Workflows: Opportunities and Challenges. Applied and Computational Engineering. 100. 49-54. 10.54254/2755-2721/100/20251754.

Khan, Arif & Badshah, Sher & Liang, Peng & Waseem, Muhammad & Khan, Bilal & Ahmad, Aakash & Fahmideh, Mahdi & Niazi, Mahmood & Azeem Akbar, Muhammad. (2022). Ethics of AI: A Systematic Literature Review of Principles and Challenges. 10.1145/3530019.3531329.

McKinsey (2025). The state of AI: How organizations are rewiring to capture value. McKinsey. Retrieved May 11, 2025, from https://www.mckinsey.com/capabilities/quantumblack/our-insights/the-state-of-ai

Negri-Ribalta, C., Geraud-Stewart, R., Sergeeva, A., & Lenzini, G. (2024). A systematic literature review on the impact of AI models on the security of code generation. Frontiers in big data, 7, 1386720. https://doi.org/10.3389/fdata.2024.1386720 .

Odeh, Ayman & Odeh, Nada & Mohammed, Abdul. (2024). A Comparative Review of AI Techniques for Automated Code Generation in Software Development: Advancements, Challenges, and Future Directions. TEM Journal. 726-739. 10.18421/TEM131-76.

OpenAI. (2022). Introducing ChatGPT. https://openai.com/index/chatgpt/

Pink, Sarah & Quilty, Emma & Grundy, John & Hoda, Rashina. (2024). Trust, artificial intelligence and software practitioners: an interdisciplinary agenda. AI & SOCIETY. 40. 639-652. 10.1007/s00146-024-01882-7.

Scoccia, G. (2023). Exploring Early Adopters' Perceptions of ChatGPT as a Code Generation Tool. 38th IEEE/ACM International Conference on Automated Software Engineering Workshops (ASEW), 88-93.

Shani, I., & GitHub Staff (2023). Survey reveals AI’s impact on the developer experience. GitHub. Retrieved May 11, 2025, from https://GitHub.blog/2023-06-13-surveyreveals-ais-impact-on-the-developerexperience/

Stack Overflow. (2024). 2024 Developer Survey. https://survey.stackoverflow.co/2024

Snyk (2023). 2023 Snyk AI-Generated Code Security Report. Snyk. Retrieved on May 11, 2025, from https://snyk.io/reports/ai-codesecurity/

Ulfsnes, Rasmus, Moe, Nils, Stray, Viktoria, & Skarpen, Marianne. (2024). Transforming Software Development with Generative AI: Empirical Insights on Collaboration and Workflow.

Wang, R., Cheng, R., Ford, D., & Zimmerman, T. (2024). Investigating and Designing for Trust in AI-powered Code Generation Tools. Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency, 1475-1493. https://doi.org/10.1145/3630106.3658984

Zhang, Beiqi & Liang, Peng, Zhou, Xiyu, Ahmad, Aakash, & Waseem, Muhammad. (2023). Practices and Challenges of Using GitHub Copilot: An Empirical Study.

Zhang, B., Liang, P., Zhou, X., Ahmad, A., & Waseem, M. (2023). Demystifying Practices, Challenges and Expected Features of Using GitHub Copilot. International Journal of Software Engineering and Knowledge Engineering, 33(1) 1653-1672.