

The Evolution of Software Engineering: How AI is Transforming the Role of Developers

Historical Perspectives and Future Predictions on the Evolution of Programming

Word Count - 2159

Candidate Number – 42439

Table of Contents

AI acknowledgement.....	2
Abstract	3
Introduction - AI transforming the role of Developers.....	4
Literature review	4
Industry Discussions	6
The Executives Perspectives.....	6
Why Augmentation over Automate – The Human in the Loop	7
Could it make developers worse? - Who is negatively impacted	7
Discussion and Conclusion.....	8
References.....	9

AI acknowledgement

I used AI tools like large language models; my personal preference is Microsoft Co-pilot for gathering and synthesize information. I often take notes while reading papers with speech to text tools and then use Co-pilot to fix the grammar and structure of these notes.

It was also used to discuss ideas and extrapolate on topics, but I do find it is quite surface level without extensive prompt engineering and time commitment which could be spent on traditional research. I also used Co-pilot to assist in making sure I referenced correctly, I had always used Harvard style for my undergraduate degree, and I am always worried about APA 7's complexities. I also used NotebookLM to turn some of my selected papers and news articles into podcasts so I could digest these information sources while traveling home. While using this I also combined the papers by section as it does an excellent job in combining and contrasting the viewpoints if you request it correctly,

I also used Grammarly AI to help with some of the grammar, spelling, and sentence construction to make sure I have not made any mistakes in those areas. It helps maintain a formal writing style and reduce work overuse. I had already had experience with these tools so did not have much learning or onboarding time for this paper, but I think this saved me around 10 hours work, but this is a rough estimate.

And my hypothetical change for the essay next year would be for students to challenge or disagree with widely accepted belief, tool or practice in AI development or tool use. Building students ability to challenge the mainstream, critically analyse and search for more obscure data would help them in their career and for topics they could use for a PHD or publish a paper.

Abstract

Artificial Intelligence (AI) is revolutionizing the field of software engineering, influencing the tasks, responsibilities, and roles of developers. This paper examines historical perspectives, current industry thought and future predictions of the evolution of programming. We focus on specific AI tools for development like LLM's like ChatGPT, GitHub Copilot, Claude, and Amazon's CodeWhisperer. This paper explores how tools effect developers regarding their enhancement of productivity and automation of tasks, shifting the role of developers to higher-level problem-solving and innovation. It discusses the implications of AI generated code, how it effects the skills of developers and explores their potential role as curators and managers of AI-generated code. We highlight key insights from industry leaders, reenforcing it with statistical trends. It also raises some ethical and practical concerns AI tool cause in development like the risks of overreliance on AI tools. The conclusion echos the need to embrace change and adapt to the future of software engineering and explores my opinions on the future of software development with AI tools. Reflecting on the positive impact of AI tools while acknowledging current fears of replacement.

"There is nothing permanent except change." - Heraclitus

Introduction

AI has emerged as a revolutionary force, reshaping jobs, and livelihoods for knowledge workers. "AI is expected to continue its transformative impact across multiple sectors, including healthcare, finance, and manufacturing" (Costa et al, 2024, p. 1531).

Software developers have found the way they learn, produce and test code change with the advent of AI tools. The pace of AI tool development from large language models (LLMs) hitting the consumer market starting with ChatGPT in 2022 has been staggering. The availability of large, centralised codebases like GitHub meant training LLM's at scale was possible, producing some incredibly powerful coding tools trained on millions of lines of code (Marar, 2024).

From that the ability of single programmers to produce and create code has been expanded. Rather than needing to learn coding skills to branch into new areas of development, that process can be replaced with a single prompt. Some skills, abilities and knowledge that mattered in the pre-AI world have been rendered obsolete, enabling a wave of new coders to emerge with vibrant and diverse skills.

This essay seeks to examine the implications of AI on software engineering roles, contrasting contemporary perspectives and outcomes with some historical predictions from academic sources. It includes a critical analysis of the future of coding, exploring perspectives from industry leaders, current statistical trends, current research, and future outcomes.

This paper will be focusing on AI coding tools like ChatGPT, GitHub Copilot, Claude, and Amazon's CodeWhisperer and full end-to-end development kits like Devin in the context of enterprise environments and freelancing and analyses the perspectives of industry leaders on these tools. In addition, it will examine the literature on what these new tools provide, what they need to do to be effective and how they can be improved in the future

Literature review

"Generative AI and Empirical Software Engineering: A Paradigm Shift," explores how artificial intelligence transforms the role of software engineers by examining the potential for AI to automate specific coding tasks. Their findings state AI can significantly enhance productivity by automating routine

tasks for developers, do bug testing and analysing large data sets. These AI coding tools reduced the importance of writing code, pushing new tasks like curating produced code (Treude and Storey, 2025).

This also necessitates a change in the most valuable skill sets for engineers, focusing more on higher-level problem-solving and innovation. This is reinforced in contemporary papers, with Costa et al (2024, p. 1524) stating "AI-driven tools are transforming the coding phase by automating code generation and providing intelligent code assistance."

Microsoft introduced IntelliCode which uses machine learning to predict code, improving developer productivity by suggesting relevant code snippets. Microsoft designed this after research indicated the best practice with AI tools required transparent, autonomous, collaborative, and user-friendly AI-driven software development (Marar, 2024). As AI tools developed and improve, they will generate more of the total code for developers (Yuan, 2025).

Other studies researching the developer-AI tool relationship found key avenues for development. One discussed area is personalised environments enabling tailored human-tool relationships with individual developers' preferences and workflows (Damioli et al, 2025). Another development avenue is collaborative AI systems working alongside human developers to enhance productivity and innovation. These systems will facilitate better communication and coordination among development teams (Terragni et al, 2025). GitHub research found the average developers in enterprise settings had 21 people on a project (Shani and GitHub Staff, 2023)

One paper from 2021 just before the widespread AI tool implementation captured past predictions and perspectives on AI tools. They believed AI would assist and not replacing humans entirely and expected some coding tasks would be too difficult for AI to complete. Humans would be required for oversight and ethics with ethical challenges prevalent, fearing AI displaying ingrained biases and limitations learnt from humans. Finally, the physical technological limiting considering the processing capacity (Harman, 2024)

Taking from the "Proceedings of the 7th Joint Conference of the Austrian Computer Society (OCG)" in 1992 had two articles on AI future effect on coding. It was foreseen in Biró and Maros (1992) that "cooperative problem-solving, systems modelling, and cognitive psychology highlights the development of AI tools that assist in software engineering tasks.". This indicates that other academics predicted the development of the AI tools that augmented or automated software engineering tasks.

Vaidya and Asif (2023) focused on the ethical and practical considerations of AI decision-making, emphasizes the importance of human oversight to ensure ethical and unbiased outcomes. This hammers that the ethical dilemma of AI tool use in programming was identified well before these tools were created and used.

Industry Discussion

The Executives Perspectives

AI is inescapable; almost all tech firms are either investing in AI or have already implemented it in their business processes. EY AI pulse survey found that 95% of senior leaders report that their organisations are currently investing in AI (McWilliams, 2024). In 2023, Goldman Sachs predicted Investment will reach and surpass \$200 billion globally by 2025 (Goldman Sachs, 2023). Now two years later estimates have Facebook, Google, Microsoft, and Amazon expected to invest \$320 billion alone in 2025 (PYMNTS, 2025).

However, MIT's review on transforming software with generative AI found only 12% of businesses believed AI tools would fundamentally change how software is developed. 38% of respondents believe generative AI will change in 1-3 years and another 31% in 4-10 years (MIT Technology Review, 2024).

Let us examine the perspectives of industry leaders on how they believe AI tools will affect software developers and the industry. AWS CEO Matt Garman asserted that AI would replace traditional coding roles, relegating software engineers to curators. Garman stated developers would transition to curators or supervisors in just 2 years (Kim, 2024). Specific tools like Claude and ChatGPT 4.0 can already write code, design systems and manage libraries. GitHub tool GitHub Copilot is stated to be able to automate 80% of coding tasks in the average project. Even staples in evaluating the skill of a coder during interviews like Leetcode tests are getting disrupted and outmanoeuvred by a well-designed AI tool from one coder who created an assistant to effectively automate the test mid interview (Altehek, 2025). One way to see this shift in coder demand is how U.S tech companies laid off 152,499 roles from 550 tech companies in 2024 (Layoffs.fyi, 2025). In comparison generative AI postings have exploded with 75 times increase from April 2022 to April 2024 (Morris, 2024).

Bret Taylor, CEO of Sierra an AI conversation platform and ex CEO of salesforce and EX-CTO of meta told Tim O'Reilly that AI agents will be the primary tool and interface for customers. He thinks that backend developers could reskill into "Agent Engineers" to remain relevant and focus on React (O'Reilly, 2025). Google's CEO Pichai stated that already over a quarter of all new code at Google is created by AI, then reviewed by their engineers (Edwards, 2024). The head of Developer Experience for Chrome, Addy Osmani says that AI can get you 70% of the way there, but the last 30% requires experience and innovation that coding tools cannot provide. The junior coders who learnt everything with AI tools can lack the final 30%. (Osmani, 2024).

It is also important not to take these statements as fact. IBM CEO Arvind Krishna said publicly tools will really generate 20-30% not 90% predicted by CEO of Anthropic Dario Amodei (Wiggers, 2025). But

IBM wants AI coding tools to be accepted by engineers without fear while they stopped back end hiring of coders in 2023, predicting AI replacing their junior intake (Bomey, 2023).

Why Augmentation over Automate – The Human in the Loop

We have seen how good these AI tools are expected to become, but what limitations are going to keep humans in the loop. Why are we predicting augmentation instead of full automation? These tools just regurgitate what already worked, it remembers previous success and replicates it. There is a risk of overreliance which could lead to the erosion of critical skills. Research found ChatGPT, GitHub Copilot, and Amazon CodeWhisperer spat out correct code just 65.2%, 46.3%, and 31.1% of the time (Yetiştiren et al, 2023). Thomas Wolf, CEO of Hugging Echos these thoughts on that AI treats the information it is trained on as intangible absolutes and will reiterate and agree with what it has been given before, not seeking to explore or grow and end up as a perennial yes man to its human users (Wolf, 2025).

AI Tool dependent companies experience more downtime and issues with their codebase. These businesses are more likely to be unable to fix AI tool produced code to address these errors. A culture of poor pre-implementation reviews and blind trust are common culprits for these problems. We see the emergence of the phenomenon of low skill developers who wrote significantly less secure code but were more likely to believe they wrote secure code when using AI tools (Perry et al, 2023).

Could it make developers worse? - Who is negatively impacted

Future dependence on AI tools could lead to a decline in critical thinking and problem-solving skills of not only junior developers but also senior developers reliant on the tools for an extended period, creating blind spots (Vaidya and Asif, 2023). A study on AI assistance saw the most quality increases in low performers code output and had less impact on high performers. This suggests that less experienced coders rely on these tools heavily and have implicit bias towards the AI code believing it is high quality. This code quality perception can have performance and security considerations when that generated code is used (Peng et al, 2023).

For independent coders, the top freelancers see disproportionate effects on their earnings. Coders with high-quality service history on freelancing platforms, measured by past performance and employment history, do not moderate the adverse effects of generative AI users entering the platform. Top freelancers are disproportionately affected by AI, suggesting that generative AI reduces overall demand for knowledge workers of all types (Hui et al, 2023). The current batch of code producing tools still have measurable limits with tools like \$500 a month builder like Devin struggling to tackle tasks with only a 15% success rate on its challenges (Husain et al, 2025)

Discussion and Conclusion

I have explored the impact of AI on software engineering throughout this paper. It charts the shifting role of software engineers from coding to curating and managing AI-generated code. The transformation of roles and responsibilities required in this landscape are radically different and change how they will be approached in the future. Curation, creation, and the human element will be so important, AI may not prioritize efficiency or structure.

In my opinion, experienced developers with a good understanding of the implicit knowledge of codebases, networks, and servers are still set up to survive. The seniors of the future could have multiple AI tools with unique thinking patterns, training and innovation pathways, tasked with parsing through their outputs to jigsaw technical solutions. Already many software jobs are just minding and curating the bots and algorithms that maintain the digital world (O'Reilly, 2016). This is just the extension of an already ongoing process.

“We must become editors as software engineers. We must become system-level thinkers, as opposed to the individual,” Anand Kulkarni, CEO of Crowdbotics (Jones, 2024). The top-end engineers will get better, and the bottom developers will struggle. As per-developer output increases, the need for lower skill developers will decrease as better developers absorb more of the available work using these tools. Coding has evolved from a static creation and delivery process to a constant cycle of improvement, change, and innovation. The drive for openness and accessibility has allowed anyone with interest to become a coder, developer, tester, or builder. The barriers to entry gone, democratizing the power to build digital constructs.

I believe the separation between human language and coding language is closing. These coding tools have enabled human language, not coding language, to be the medium for humans to design technology. This evolution allows machines and algorithms to understand human communication, humanizing our interactions in technology construction. In the future will we code and create with just human language, leaving coding interactions to the AI tools.

Software engineering will continue to be shaped by the evolution of these AI tools and technologies. In conclusion, Software Developers roles will be shaped by AI's advancements, but also play a role in how those specific tools affect their work. Their approach and ability in using these tools will define that relationship. Embracing change, seeking to evolve, and accepting adaption will let developers thrive. These tools allow developers to do so much so fast; Pandora's box has been opened and there is no going back. I genuinely believe that these AI tools are a positive for software developers and the fears of true replacement are overblown currently. Do not forget, it might be writing the code, but we are doing the programming.

References

- Altchek, A. (2025). *Columbia suspends student who created AI tool that helps people cheat in coding interviews*. Business Insider. Retrieved 16 April 2025, from <https://www.businessinsider.com/columbia-suspends-student-ai-interview-coder-cheat-tool-chungin-lee-2025-3>
- Biró, M., & Maros, I. (1992). The Use of Deep Knowledge from the Perspectives of Cooperative Problem Solving, Systems Modeling, and Cognitive Psychology. In R. Mittermeir (Ed.), *Shifting Paradigms in Software Engineering* (pp. 56–67). Springer. https://doi.org/10.1007/978-3-7091-9258-0_7
- Bomey, N. (2023, May 2). *IBM's CEO just said it out loud: He'll eliminate jobs with AI*. Axios. <https://www.axios.com/2023/05/02/ibm-stock-artificial-intelligence-hiring>
- Costa, P. M., & Almeida, C. (2024). Innovations in AI and Their Impact on Software Engineering and Beyond. 04(07). https://www.researchgate.net/publication/387271524_Innovations_in_AI_and_Their_Impact_on_Software_Engineering_and_Beyond
- Damioli, G., Van Roy, V., Vertesy, D., & Vivarelli, M. (2025). Is artificial intelligence leading to a new technological paradigm? *Structural Change and Economic Dynamics*, 72, 347–359. <https://doi.org/10.1016/j.strueco.2024.12.006>
- Lizzie McWilliams. (2024). *New EY research finds AI investment is surging, with senior leaders seeing more positive ROI as hype continues to become reality*. Retrieved 15 April 2025, from https://www.ey.com/en_us/newsroom/2024/07/new-ey-research-finds-ai-investment-is-surging-with-senior-leaders-seeing-more-positive-roi-as-hype-continues-to-become-reality
- Edwards, B. (2024, October 30). *Google CEO says over 25% of new Google code is generated by AI*. Ars Technica. <https://arstechnica.com/ai/2024/10/google-ceo-says-over-25-of-new-google-code-is-generated-by-ai/>
- Goldman Sachs. (2023). *AI investment forecast to approach \$200 billion globally by 2025*. Retrieved 15 April 2025, from <https://www.goldmansachs.com/insights/articles/ai-investment-forecast-to-approach-200-billion-globally-by-2025>
- Harman, M. (2012). The role of Artificial Intelligence in Software Engineering. *2012 First International Workshop on Realizing AI Synergies in Software Engineering (RAISE)*, 1–6. <https://doi.org/10.1109/RAISE.2012.6227961>
- Hui, X., Reshef, O., & Zhou, L. (2023). *The Short-Term Effects of Generative Artificial Intelligence on Employment: Evidence from an Online Labor Market* (SSRN Scholarly Paper No. 4527336). Social Science Research Network. <https://doi.org/10.2139/ssrn.4527336>

- Husain, H., Flath, I., & Whitaker, J. (2025, January 8). *Thoughts On A Month With Devin*. Answer.AI. <https://www.answer.ai/posts/2025-01-08-devin.html#what-is-devin>
- Jones, H. (n.d.). *The Automation Takeover: Are Software Engineers Becoming Obsolete?* Forbes. Retrieved 15 April 2025, from <https://www.forbes.com/sites/hessiejones/2024/09/21/the-automation-takeover-are-software-engineers-becoming-obsolete/>
- Kim, E. (2024). *AWS CEO Says Most Developers Could Stop Coding Soon as AI Takes Over—Business Insider*. Retrieved 15 April 2025, from <https://www.businessinsider.com/aws-ceo-developers-stop-coding-ai-takes-over-2024-8>
- Kim, E. (2024). *In a leaked recording, Amazon cloud chief tells employees that most developers could stop coding soon as AI takes over*. Business Insider. Retrieved 15 April 2025, from <https://www.businessinsider.com/aws-ceo-developers-stop-coding-ai-takes-over-2024-8>
- Layoffs.fyi. (2025). *Layoffs.fyi—Tech Layoff Tracker and DOGE Layoff Tracker*. Layoffs.Fyi. Retrieved 15 April 2025, from <https://layoffs.fyi/>
- Marar, H. W. (2024). Advancements in software engineering using AI. *Computer Software and Media Applications*, 6(1), 3906. <https://doi.org/10.24294/csma.v6i1.3906>
- MIT Technology Review. (2024, October 17). *Transforming software with generative AI*. MIT Technology Review. <https://www.technologyreview.com/2024/10/17/1105295/transforming-software-with-generative-ai/>
- Morris, C. (2024, June 6). *Generative AI job postings increase tenfold in the past year*. Fast Company. <https://www.fastcompany.com/91136553/generative-ai-job-postings-increase-indeed>
- Nascimento, N., Alencar, P., & Cowan, D. (2023). *Artificial Intelligence versus Software Engineers: An Evidence-Based Assessment Focusing on Non-Functional Requirements*. In Review. <https://doi.org/10.21203/rs.3.rs-3126005/v1>
- O'Reilly, T. (2016, May 31). *Managing the Bots That Are Managing the Business*. MIT Sloan Management Review. <https://sloanreview.mit.edu/article/managing-the-bots-that-are-managing-the-business/>
- O'Reilly, T. (2025, February 4). *The End of Programming as We Know It*. O'Reilly Media. <https://www.oreilly.com/radar/the-end-of-programming-as-we-know-it/>
- Osmani, A. (2024, December 4). The 70% problem: Hard truths about AI-assisted coding [Substack newsletter]. *Elevate*. <https://addyo.substack.com/p/the-70-problem-hard-truths-about>
- Ozkaya, I. (2022). A Paradigm Shift in Automating Software Engineering Tasks: Bots. *IEEE Software*, 39(5), 4–8. <https://doi.org/10.1109/MS.2022.3167801>

Peng, S., Kalliamvakou, E., Cihon, P., & Demirer, M. (2023). *The Impact of AI on Developer Productivity: Evidence from GitHub Copilot* (No. arXiv:2302.06590). arXiv. <https://doi.org/10.48550/arXiv.2302.06590>

Perry, N., Srivastava, M., Kumar, D., & Boneh, D. (2023). Do Users Write More Insecure Code with AI Assistants? *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, 2785–2799. <https://doi.org/10.1145/3576915.3623157>

PYMNTS. (2025, February 9). Tech Giants Plan Massive \$320 Billion AI Spend for 2025. *PYMNTS.Com*. <https://www.pymnts.com/artificial-intelligence-2/2025/tech-giants-plan-massive-320-billion-ai-spend-for-2025/>

Shani, I., & GitHub Staff. (2023, June 13). Survey reveals AI’s impact on the developer experience. *The GitHub Blog*. <https://github.blog/news-insights/research/survey-reveals-ais-impact-on-the-developer-experience/>

Terragni, V., Vella, A., Roop, P., & Blincoe, K. (2025). The Future of AI-Driven Software Engineering. *ACM Transactions on Software Engineering and Methodology*, 3715003. <https://doi.org/10.1145/3715003>

Thomas Wolf [@Thom_Wolf]. (2025, March 6). *I shared a controversial take the other day at an event and I decided to write it down in a longer format: I’m afraid AI won’t give us a ‘compressed 21st century’. The ‘compressed 21st century’ comes from Dario’s ‘Machine of Loving Grace’ and if you haven’t read it, you probably* [Tweet]. Twitter. https://x.com/Thom_Wolf/status/1897630495527104932

Treude, C., & Storey, M.-A. (2025). *Generative AI and Empirical Software Engineering: A Paradigm Shift* (No. arXiv:2502.08108). arXiv. <https://doi.org/10.48550/arXiv.2502.08108>

Vaidya, J., & Asif, H. (2023). *A Critical Look at AI-Generated Software—IEEE Spectrum*. Retrieved 15 April 2025, from <https://spectrum.ieee.org/ai-software>

Váncza, J., & Márkus, A. (1992). Lessons of a First-Year Use of the Automated Reasoning Tool. In R. Mittermeir (Ed.), *Shifting Paradigms in Software Engineering* (pp. 68–79). Springer. https://doi.org/10.1007/978-3-7091-9258-0_8

Wiggers, K. (2025, March 11). IBM’s CEO doesn’t think AI will replace programmers anytime soon. *TechCrunch*. <https://techcrunch.com/2025/03/11/ibms-ceo-doesnt-think-ai-will-replace-programmers-anytime-soon/>

Yuan, E. (2025, April 10). Debug-gym: Can AI agents lighten developers’ debugging load? *Microsoft Research*. <https://www.microsoft.com/en-us/research/blog/debug-gym-an-environment-for-ai-coding-tools-to-learn-how-to-debug-code-like-programmers/>

Yetiştiren, B., Özsoy, I., Ayerdem, M., & Tüzün, E. (2023). *Evaluating the Code Quality of AI-Assisted Code Generation Tools: An Empirical Study on GitHub Copilot, Amazon CodeWhisperer, and ChatGPT* (No. arXiv:2304.10778). arXiv. <https://doi.org/10.48550/arXiv.2304.10778>