AI Helping vs. Replacing Software Development

Abdur-Rahman Bilal

1. INTRODUCTION

Many debate about whether AI replaces software developers or helps them work better. This paper will focus on that argument and explain how AI affects tasks such as coding, debugging, and software workflows. However, the caveat is that they may still raise concerns about job changes for beginners.

After looking at different perspectives, my argument is the following: Although AI is changing software development, the main issue is whether it is replacing software engineers or helping them work more effectively and efficiently. Based on recent research, AI tools are mostly aiding developers in their productivity and helping them with complex tasks.

This paper will compare three main ideas: **replacement**, **stagnation**, and **augmentation**.

2. REPLACEMENT: WILL AI TAKE DEVELOPER JOBS?

In his article, "Tech Companies Should Stop Pretending AI Won't Destroy Jobs," Kai-Fu Lee argues that the rapid advancement of artificial intelligence will lead to many people losing jobs in both hands-on and professional roles.[1] He explains that AI can perform large numbers of tasks faster and at a lower cost than humans, which makes them function as a replacement for many tasks.[2]

3. STAGNATION: AI TOOLS FAILING TO DELIVER

The editors of CIO take a more mixed position. In their article, "Devs Gaining Little (If Anything) from AI Coding Assistants," they explain that AI tools do not always improve productivity.[3] They point to a study showing that developers gained little to no performance in the usage of AI tools and even experienced an increase in the number of bugs in the software.[4]

Their perspective shows that AI may not always be of use but rather be an obstacle in the way of real development work.

4. AUGMENTATION: AI AS SUPPORT RATHER THAN REPLACEMENT

Others argue that AI improves and supports software developers in their work. In his article, "AI is Transforming How Software Engineers Do Their Jobs. Just Don't Call It

---

[1] Lee
[2] Lee
[3] *CIO*
[4] *CIO*

'Vibe-Coding,'" Matt O'Brien explains that AI helps developers in handling repetitive tasks so they may focus on more important and advanced tasks.[5]

Moreover, Sudheekar Poithireddy argues in "AI-Powered Copilots Are Revolutionizing Low-Code Development in the Power Platform" that AI copilots increase developers' productivity when it comes to low-code development, and it makes the creation of websites and software easier for non-technical users.[6]

Vladimir Sonkin and Cătălin Tudose go even further in "Beyond Snippet Assistance: A Workflow-Centric Framework for End-to-End AI-Driven Code Generation," saying that AI can automate entire software workflows, improving the accuracy of the code and reducing the manual work of the developers.[7]

5. THE EARLY DATA: MIXED RESULTS AND GROWING SKILLS

Tăbuscă and his colleagues explain that AI tools improve specifically java programming, and its overall productivity.[8]

Zheyuan Cui and coauthors present workplace experiment data showing that developers complete tasks faster using AI, especially when junior engineers use it.[9]

Cihon and Demirer balance the argument by saying that we only have early results of AI, signaling that it needs time to saturate to have definitive results.[10] However, results show that AI mostly improved developer abilities and frees their time to more creative tasks.

6. UNDERSTANDING THE THREE PERSPECTIVES CLEARLY

There is a major difference when it comes to AI completely replacing jobs, and AI just not having a strong productive effect on its users in their jobs.

Some authors argue that AI will replace software developers because they can simply perform tasks faster and cheaper than humans. We can refer to this as **replacement**.

Others believe that AI will not replace jobs but will not be able to provide a boost of productivity one would expect after investing in this technology. This can be referred to as **stagnation**.

A third group believes AI is keeping jobs and helping developers work more efficiently and productively. This is called **augmentation**.

These are three separate perspectives when it comes to this argument, and they should not be treated the same.

---

[5]O'Brien

[6]Pothireddy

[7]Sonkin and Tudose

[8]Tăbuscă et al.

[9]Cui et al.

[10]Cihon and Demirer

7. Replacement and Stagnation Up Close

Kai-Fu Lee argues that AI will replace many workers, including software developers.[11] He claims AI can perform job tasks faster and less costly than humans. "It will soon be obvious that half of our job tasks can be done better at almost no cost by AI and robots."[12] His position is the most effective version of the replacement argument.

On the other hand, some authors disagree with this idea and focus on stagnation instead. The CIO editors argue that AI tools do not always help developers work better.[13] According to Ivan Gekht, CEO of Gehtsoft, "It becomes increasingly more challenging to understand and debug the AI-generated code, and the time spent on troubleshooting the AI code is so resource-intensive that it is easier to rewrite the code from scratch than fix it."[14]

I believe that both positions are important. Stagnation argues that even when AI replaces workers, it still fails to achieve productivity rates, or it may even slow it down. However, these problems do not mean that AI will fully replace developers, rather it may just replace the type of work beginners do in this field.

8. Productivity: Misunderstood or Overstated?

Cihon and Demirer examined the research available on AI and found that the results are mixed.[15] They suggest that developers will need time to adjust to AI before seeing any benefits.

However, Cui and his coauthors present evidence that developers using AI completed tasks 26% faster than those who were not using it.[16] The biggest improvement was seen in the junior developers who used AI tools more frequently.

Another factor that affects these results is how well developers can **prompt** AI tools. Prompt engineering is an essential skill—those who know how to communicate tasks accurately will see far better results. This is likely one of the reasons some studies show stagnation while others show augmentation.

9. Augmentation in Depth: AI Empowering Developers

Sonkin and Tudose argue that AI can automate entire workflow models, improving accuracy and reducing repetitive work.[17] This allows developers to focus on designing systems and solving bugs, showing AI can expand developer capabilities rather than replace them.

---

[11]Lee

[12]Lee

[13]*CIO*

[14]*CIO*

[15]Cihon and Demirer

[16]Cui et al.

[17]Sonkin and Tudose

Tăbuscă and his colleagues found that AI tools improved programming in Java specifically and sped up the rate of code generation.[18] They also affirm that knowledge of AI tools and how to use them, specifically through prompt engineering, is a very essential skill for today's developers.

Matt O'Brien explains that it is impossible for software developers to disappear because software demand is exploding faster than humans can code it.[19] "You can imagine a world where we're creating 10 times as much software. That's going to require more software engineers, not less."

Sudheekar Pothireddy shows that copilots democratize app creation and broaden who can build software, including non-technical users.[20]

10. Conclusion: Developers Won't Disappear — Their Skills Will Evolve

AI is changing software development without a doubt, and this debate is focused on 3 key areas: **replacement**, **stagnation**, and **augmentation**.

While all perspectives are supported by research, the strongest and most compelling factor is **augmentation**. AI is most powerful when developers understand how to use it properly. It improves productivity, expands capability, and accelerates development speed.

The job of the developer will most likely **shift rather than disappear**. Skills like prompt engineering, system design, and critical thinking will grow in importance. Developers who adapt will see increased performance in their work.

The future of software development will depend more on how we work with AI than whether AI replaces us.

11. Works Cited

Cihon, Peter, and Mert Demirer. "How AI-Powered Software Development May Affect Labor Markets." *Brookings Institution*, 1 Aug. 2023.

Cui, Zheyuan (Kevin), et al. "The Effects of Generative AI on High-Skilled Work: Evidence from Three Field Experiments with Software Developers." *SSRN*, 20 Aug. 2025.

"Devs Gaining Little (If Anything) from AI Coding Assistants." *CIO*, 26 Sept. 2024.

Lee, Kai-Fu. "Tech Companies Should Stop Pretending AI Won't Destroy Jobs." *Technology Review*, Mar./Apr. 2018.

---

[18]Tăbuscă et al.
[19]O'Brien
[20]Pothireddy

O'Brien, Matt. "AI Is Transforming How Software Engineers Do Their Jobs. Just Don't Call It 'Vibe-Coding.'" *AP Online*, 29 Sept. 2025.

Pothireddy, Sudheekar Reddy. "AI-Powered Copilots Are Revolutionizing Low-Code Development in the Power Platform." *International Journal of Communication Networks and Information Security*, vol. 17, no. 2, 2025, pp. 86-115.

Sonkin, Vladimir, and Cătălin Tudose. "Beyond Snippet Assistance: A Workflow-Centric Framework for End-to-End AI-Driven Code Generation." *Computers*, vol. 14, no. 3, 2025.

Tăbuscă, Alexandru, et al. "Generating Java Code with AI Tools: Usage and Implications." *Journal of Information Systems & Operations Management*, vol. 19, no. 1, Summer 2025, pp. 306-328.