



## Will Ai Replace Software Engineers?

*Mohit Ramesh Sasane<sup>1</sup>, Neha Anandkant Pandey<sup>2</sup>*

<sup>1</sup> (MC2223095) [Mohitsasane74@gmail.com](mailto:Mohitsasane74@gmail.com)

<sup>2</sup> (MC2223072) [neha.a.pandey2000@gmail.com](mailto:neha.a.pandey2000@gmail.com)

### ABSTRACT.

The development of artificial intelligence (AI) has considerably reshaped various sectors, including software engineering. This article investigates the potential of AI to supplant software engineers, examining present capabilities, constraints, and future potential. Although AI and machine learning technologies have progressed to automate routine tasks, enhance code, and even generate code snippets, the role of software engineers remains crucial due to the need for imaginative problem-solving, intricate system design, and human-centered considerations. AI tools are becoming invaluable assistants, enhancing productivity and empowering engineers to concentrate on more complex tasks. However, complete replacement is unlikely in the near future. Instead, a collaborative relationship is anticipated, where AI fortifies the capabilities of software engineers, driving innovation and efficiency in software development.

**Keywords** Chatbot, AI Models, Devin AI, Machine Learning,

### 1.Introduction :

The advancements in artificial intelligence (AI) and machine learning (ML) technologies have revolutionized many industries and redefined job roles. In software engineering, AI has introduced tools and techniques that improve coding efficiency, automate repetitive tasks, and optimize development processes. This raises the question of whether AI will eventually replace software engineers.

AI's potential to generate code, debug, and design software autonomously has sparked a debate about the future of the profession. Proponents argue that AI's capabilities are advancing rapidly and that it may soon be able to perform tasks traditionally done by human engineers. However, critics emphasize the irreplaceable value of human creativity, problem-solving, and contextual understanding in software development.

This paper examines the current state of AI in software engineering, exploring both its successes and limitations. It offers a balanced perspective on how AI is changing the field, the areas where human engineers still hold an advantage, and the emerging collaborative dynamics between AI and human intelligence. Through this exploration, we aim to determine whether AI will complement or replace the role of software engineers and what this means for the future of the profession.

- 1.1. What are the potential impacts of AI on the role of software developers?
- 1.2. How effective is AI in automating software development tasks compared to human developers?
- 1.3. What limitations does AI currently face that prevent it from fully replacing software developers?
- 1.4. How can software developers leverage AI tools to enhance their productivity without being replaced?

## 2.AI Tools and Chatbots: Enhancing Productivity and Interaction in Software Development

### A. GitHub Copilot

GitHub Copilot is a cutting-edge AI-powered code completion tool that was created by GitHub in collaboration with OpenAI. It is designed to help developers by suggesting lines or entire blocks of code as they write. To enhance the software development process, it offers a range of features. Firstly, it provides real-time code suggestions and completions directly in the code editor based on the context of the code being written.

Secondly, it supports multiple programming languages, including Python, JavaScript, TypeScript, Ruby, Go, and more. Thirdly, it uses advanced machine learning models to understand the context of the code and provide intelligent suggestions that align with the existing codebase and coding style. This includes suggesting entire functions and code blocks, which reduces the time spent on writing boilerplate code and repetitive tasks.

Additionally, it can generate documentation strings and comments for functions and code blocks, making it easier to understand unfamiliar code. Furthermore, it can provide explanations for complex code snippets, improve code quality by suggesting more efficient coding practices, and detect potential bugs or issues in the code, offering suggestions for fixes.

## ***B. DeepCode***

DeepCode, which is currently a part of Snyk, is an AI-powered code review tool created to enhance code quality by detecting bugs, security vulnerabilities, and areas for improvement. By utilizing machine learning and large-scale code analysis, it provides real-time feedback to developers. In the following sections, we will explore its key features and how it can improve the software development process.

**AI-Powered Code Analysis:** DeepCode analyzes code as it is being written, providing immediate feedback on potential issues and areas for improvement. It uses sophisticated machine learning models trained on billions of lines of code to identify coding patterns and detect anomalies.

**Comprehensive Bug Detection:** DeepCode identifies a wide range of bugs, from simple syntax errors to complex logic flaws, ensuring that code is both functional and secure by scanning for known security vulnerabilities. Additionally, the tool suggests coding best practices and improvements, helping developers write cleaner, more maintainable code.

**Multi-Language Support:** DeepCode supports various programming languages, including JavaScript, Python, Java, TypeScript, and more, making it suitable for diverse development environments.

**Seamless Integration:** DeepCode integrates with popular IDEs such as Visual Studio Code, IntelliJ IDEA, and PyCharm, allowing developers to receive feedback directly within their coding environment. Additionally, it can be integrated with version control systems like GitHub, GitLab, and Bitbucket, providing continuous code analysis throughout the development lifecycle.

## ***C. OpenAI API***

The OpenAI API grants developers access to cutting-edge AI models, such as GPT-3, which is a state-of-the-art natural language processing (NLP) model capable of generating human-like text. This API offers a variety of applications, including text generation, language translation, text summarization, and language understanding. It allows for seamless integration into web and mobile applications through RESTful APIs and the option to train custom models. The API also provides high-performance infrastructure and optimized algorithms for scalable and reliable performance. To ensure ethical AI practices, the API includes safety and compliance measures, as well as transparency in AI-generated content and decisions.

## ***D. ChatGPT***

ChatGPT is an advanced natural language processing model created by OpenAI, specifically the GPT-3.5 version. It is designed to understand and generate human-like text. Its notable features and uses include: **Contextual Understanding:** The ability to comprehend and respond to a wide range of conversational prompts while maintaining context throughout multiple turns of dialogue. **Semantic Interpretation:** Interpreting the meaning behind user queries for accurate and context-appropriate responses. **Human-like Responses:** Generating coherent and contextually relevant responses that mimic human conversational patterns and styles. **Flexible Output:** Capable of producing diverse types of text, such as informative answers, creative stories, and explanations. **Virtual Assistants:** Enhancing virtual assistants and chatbots across various platforms by providing personalized interactions and customer support. **Education and Learning:** Supporting educational applications by answering questions, explaining concepts, and offering learning resources. **Content Creation:** Assisting in generating content for articles, blogs, social media posts, and marketing materials based on provided prompts. **Productivity Tools:** Integrating with productivity tools to automate tasks like scheduling, reminders, and information retrieval through natural language commands.

**Entertainment:** Engaging users through interactive storytelling, jokes, quizzes, and other entertainment-oriented interactions. **API Access:** Available through OpenAI's API, allowing integration into websites, mobile apps, and other digital platforms for real-time interactions.

**Custom Responses:** Enabling developers to customize ChatGPT's responses to align with specific business or application requirements, enhancing user experience and functionality.

**Safety and Privacy:** Upholding ethical standards by prioritizing user safety, privacy, and responsible AI practices in interactions and data handling. **Bias Mitigation:** Implementing measures to reduce biases in responses and interactions, ensuring fair and inclusive user experiences.

## ***E. Character.ai***

Character.ai is a cutting-edge platform that harnesses the power of artificial intelligence to create and animate virtual characters. This platform offers a wide range of features and applications, including:

**Virtual Characters:** Character.ai generates realistic virtual characters with natural facial expressions, movements, and behaviors.

**Customization:** Users have the ability to personalize various aspects of the characters, such as their appearance, personality traits, and voice characteristics.

**Realistic Animation:** The platform utilizes AI algorithms to animate characters dynamically, resulting in natural movements and gestures.

**Behavioral Simulation:** The characters exhibit lifelike behaviors and interactions, making them suitable for various applications, including storytelling, virtual assistants, and entertainment.

**API Integration:** Character.ai provides APIs that enable developers to integrate virtual characters into a variety of applications, websites, and virtual environments.

**Cross-Platform Compatibility:** The platform is compatible with deployment across different platforms and devices, enhancing accessibility and usability.

**Interactive Experiences:** Users can engage in real-time interaction with virtual characters, fostering immersive and engaging experiences.

Response Generation: AI algorithms enable characters to generate contextually appropriate responses and adapt to user inputs dynamically.

Entertainment: Character.ai enhances storytelling, gaming, and digital entertainment experiences by introducing interactive and dynamic characters.

Education: The platform facilitates interactive learning experiences through virtual tutors, educational games, and simulations.

Customer Engagement: Character.ai improves customer interaction and support services through virtual assistants and interactive customer service agents.

Marketing and Advertising: The platform utilizes virtual characters for promotional campaigns, virtual spokespersons, and brand ambassadors.

---

### 3. Adaptability in Software Engineering: Embracing AI as a Tool, Not a Replacement

The most important quality for a software engineer is adaptability. Given the rapid pace of technological advancement in this industry, it is essential to continuously learn and utilize new tools. Engineers who fail to adapt may not be well-suited for this field. However, it is important to note that AI will not replace software engineers. This notion stems from a misunderstanding of how technology operates. AI is simply another tool to add to our toolkit, and embracing it can streamline our workflow. By utilizing AI tools such as GitHub Copilot, we can leverage their capabilities to our advantage.

Marketing around AI often exaggerates its advancements, creating temporary hype. While researchers make significant breakthroughs and present impressive demos, this excitement invariably subsides until the next major development. Despite predictions of programmers being replaced by new technologies for decades, this has not come to pass. For instance, the introduction of website builders did not render web developers obsolete. Similarly, Web3.0 has been around since 2014, but it has not replaced Web2.0 as expected.

Can we truly trust AI to handle tasks for us when it is only 90% accurate? AI-generated code may contain numerous bugs because it is trained on flawed code from the internet. The internet is replete with buggy code since no one writes perfect code. Consequently, AI models like GPT are trained on a vast amount of defective data, leading to the perpetuation of these issues. Until we can create flawless code for AI to learn from, it will continue to replicate existing problems. While we must adapt to using AI as a tool, it is unlikely that AI will fully replace software engineers.

People have been discussing the possibility of AI replacing software developers since the 1960s or even earlier. However, this notion stems from a lack of understanding of what software development truly entails. Software development is not just about generating code; rather, it involves solving problems. While generating code can be effortless with the help of tools like GPT, it is the problem-solving aspect that sets software developers apart from machines. While GPT can produce code efficiently, it cannot solve problems like a human can. Just as a nailgun cannot build a house, GPT cannot replace a software developer entirely. To replace software developers, we need an advanced form of AI that can perform tasks across various domains, not just text generation. Unfortunately, we seem no closer to achieving this than we were decades ago.

Consider the example of Stack Overflow, where a wealth of code is readily available for copying and pasting. However, this does not mean that a copy-paste monkey can replace a human developer. Instead, ChatGPT can be thought of as an efficient search engine for Stack Overflow, capable of automating code generation in some cases. Nevertheless, intelligent human input is still necessary to determine what needs to be automated.

Moreover, there are certain situations where automatic code generation is not advisable, such as when sensitive, privacy-bound data is involved. For instance, Samsung engineers once exposed their non-open source code to OpenAI's API, inadvertently giving Microsoft access to their proprietary information. Although Microsoft had a privacy policy that prevented them from looking at the code, the fact that the Samsung engineers voluntarily surrendered the code to a different API under different contracts created a legal loophole that allowed Microsoft to view the code. In these types of context-specific situations, human developers are still indispensable.

---

### 4. Devin.ai

The capacity of Devin AI to carry out software engineering jobs independently has drawn attention. The software uses machine learning approaches to develop, debug, plan, and solve problems more effectively than the GitHub Copilot tool. When a user prompts Devin AI with a task in natural language, the program responds by displaying its strategy and running the code. When learning how to do a task, it seeks web resources. During the implementation phase, the software also listens to user cues and modifies its plans accordingly, for example, when a user reports a defect or issue.

Devin AI can be used, for example, to create websites. According to a test done by Bloomberg, the technology could replicate a Pong website in about the same amount of time as it could develop a website in ten minutes. The tool also generated a strategy, source code, and benchmark tests for a website based on the Llama 2 language model, according to a Cognition Labs demo. Other instances include assembling a computer vision model from an Upwork project and creating a project to display photographs from a blog article. Devin was shown to resolve 13.86 percent of problems with no human aid in a benchmark test for evaluating the effectiveness of big language models on real-world tasks, compared to an average of 1.96 percent and 4.8 percent.

Devin has been evaluated on SWE-bench, a benchmarking platform used primarily by software engineers that assigns agents to fix real-world problems on open-source projects. Devin was assessed using a randomly selected 25% subset of the dataset, claims Cognition. Devin was left on his own, but all models received assistance—that is, they were informed of the precise files that required editing. Devin successfully handled 13.86% of the problems from start to finish, a significant improvement over Claude 2's 4.8% and ChatGPT-4's 1.74%. Cognition promised to release a more thorough technical report shortly!

Regarding the future of software employment and associated ones, the astounding numbers, as seen on the benchmarking, have confused people's thoughts, especially those of software developers and engineers.

According to Cognition, an applied AI lab that specializes in reasoning, it is creating AI partners with powers beyond those of current AI tools. Devin is a dedicated, capable teammate who is equally prepared to work independently to finish projects for you to approve or to build alongside you, according to Cognition. With Devin, engineering teams may aim for more ambitious objectives and engineers can concentrate on more fascinating issues.

Oddly enough, Cognition, the company that created Devin, is actively employing "human" software engineers, despite the widespread belief that Devin represents the end for many software professionals! There are differing views, and we are unable to draw any judgments until Devin has undergone extensive testing.

Automating software engineering will resemble automating driving, according to Andrej Karpathy, the former director of AI at Tesla. He continues by saying that there will be significant changes in software engineering. It would include pitching in high-level directives, concepts, or progression techniques in English, along with a great deal more supervised automation.

---

## 5. Devin.ai is a Lie

Even with these great credentials, there remains suspicion in the development community due to a number of articles and a viral video that suggest Devin's abilities may have been exaggerated. While Devin shows promise, some contend that its operational capabilities in routine coding jobs may not be fully reflected in the presentations and statistics provided by Cognition AI. These criticisms have acquired popularity on technical audiences' discussion and analysis forums, such as Medium and YouTube.

The aforementioned video, which has received a great deal of attention, implies that the chores Devin is claimed to accomplish on his own might have been organized in before or made simpler in order to improve the performance results. The writers of a number of other articles and videos share this suspicion, casting doubt on the openness of the standards that were used to assess Devin's performance.

According to the author of a video uploaded by user Internet of Bugs, Devin was unable to do a task on the Upwork site as the developers had indicated.

The developer community has responded in a variety of ways. Some developers are excited about how AI might eliminate repetitive coding duties, freeing up human workers to work on more difficult and innovative projects. Others, on the other hand, are wary, drawing attention to the differences that exist between stated capabilities and real performance in uncontrolled conditions. The ethical implications of AI in software development, including employment displacement and the dependability of code generated by AI, are also being discussed in this debate.

Is Devin phony then? Although it's too soon to say, the development community must closely monitor technologies like Devin as AI begins to permeate more and more industries. Although it's a big step forward for AI-driven software development, there's still disagreement about the full scope of its potential. It has to be seen if Devin can actually deliver on the promises offered by Cognition AI in more extensive real-world applications. The developer community is still keeping a close eye on Devin and is interested to see how it develops and how it might eventually change the software engineering landscape.

The significance of openness and thorough, independent testing of AI technology prior to their widespread adoption and application is highlighted by this ongoing discussion.

---

## Summary :

Software engineering is one of the many industries that have seen a transformation thanks to advances in AI and ML, which have improved coding productivity, automated activities, and optimized development processes. This begs the question of whether software engineers will be replaced by AI. But software engineers need to be flexible in order to succeed, and artificial intelligence is seen as a tool to support rather than replace them in their work. Examples of how AI may improve workflows are shown by programs like GitHub Copilot. Devin AI is a powerful tool that uses machine learning (ML) techniques to build, debug, plan, and solve problems. It can execute software engineering activities independently, responding to user inputs and adjusting based on feedback.

---

## REFERENCES :

1. [https://www.reddit.com/r/computerscience/comments/12kee2p/will\\_ai\\_replace\\_software\\_engineers/](https://www.reddit.com/r/computerscience/comments/12kee2p/will_ai_replace_software_engineers/)
2. <https://www.comptia.org/blog/will-ai-replace-software-developers>
3. <https://opencv.org/blog/devin-ai-software-engineer/>
4. <https://www.codemotion.com/magazine/ai-ml/is-devin-fake/#:~:text=Rising%20skepticism%3A%20Devin%20Fake%20Claims&text=Critics%20argue%20that%20while%20Devin,capabilities%20in%20everyday%20coding%20tasks.>
5. <https://yourstory.com/2024/04/debunking-devin-ai-software-engineering-claims-exposed>