PriMera Scientific Engineering
Volume 5 Issue 4 October 2024
DOI: 10.56831/PSEN-05-158
ISSN: 2834-2550

PriMera Scientific Publications

Artificial Intelligence in Software Engineering

Type: Editoral Note
Received: July 28, 2024
Published: September 30, 2024

Citation:
Boris Martinovic. "Artificial Intelligence in Software Engineering". PriMera Scientific Engineering 5.4 (2024): 78-91.

Boris Martinovic*

University of Mostar, Bosnia and Herzegovina

***Corresponding Author:** Boris Martinovic, University of Mostar, Bosnia and Herzegovina.

In a relatively short period of time, artificial intelligence has become one of the most powerful technologies and an irreplaceable companion to many software engineers. Nowadays, there are multitudes of ways in which artificial intelligence has embedded itself in the software engineering process, giving engineers an edge during different stages of the process.

Due to the wide array of available tools, developers are using both general-purpose and specialized AI tools to enhance the development experience. General-purpose tools, such as ChatGPT, can be used to find information, generate simpler code solutions, or research ideas. On the other side, there are many tools which are specialized for development, which offer more in-depth knowledge and deeper integration with development tools. There are numerous such tools, among which are GitHub Copilot, Supermaven, Google IDX, and Vercel V0.

GitHub Copilot and Supermaven are tools that enhance the development experience within the engineer's preferred development environment (usually Visual Studio Code, Neovim, or JetBrains IDE) by providing automatic code completion, chat interface for queries, research, and more. The main advantage that those tools offer is the benefit of the context of the codebase, which means that engineers have a personalized experience when using such tools. Another positive is that the aforementioned tools adapt the code style to the codebase, using a similar style (or language) of variable names, the way of writing code blocks, and writing appropriate tests.

An example of deeper integration of AI in software development tools is Google IDX. Being a cloud-based integrated development environment, it does not require a separate AI-based tool to improve developer productivity. Aside from offering similar benefits as GitHub Copilot and Supermaven, IDX also provides further AI-assisted task automation, such as writing documentation or agent templates, which provide the starting point for a project or process.

AI tools go beyond just assisting engineers in parts of their workflows, but can be used to create components for their web application. One such example is V0 by Vercel. It uses AI to generate React.js components that use shadcn/ui UI components and Tailwind CSS to generate styles. It allows AI, which is "under the hood", to generate components better, as it creates them based on a well-defined template.

Despite having many advantages, different tools powered by artificial intelligence also come with some drawbacks. The main drawback of general-purpose AI tools is that an engineer has to have some experience in the topic they are researching/prompting for in order to have a completely safe experience. Besides that, the cost is also a factor which must be considered, since many professional tools

do have a subscription or token-based payment system, which can grow a lot if an engineer is using multiple tools at the same time.

Artificial intelligence today offers many ways for software engineers to make their work more productive than ever. Whether it be help inside their integrated development environments or outside of them, there are many pathways engineers can take to enhance their development experience. Despite already having many options, it is clear that this is just a beginning of a larger trend that is moving towards bigger inclusion of artificial intelligence in daily workload of software engineers.