## Journal of Artificial Intelligence & Cloud Computing

### **Review Article**



Open d Access

# GitHub Copilot: The Disrupting AI Companion Transforming the Developer Role and Application Lifecycle Management

Laxminarayana Korada

USA

#### ABSTRACT

GitHub Copilot, an AI tool developed by GitHub and Open AI, has revolutionized software development by providing real-time code suggestions and facilitating application optimization. It offers features like code completion, GitHub Copilot Chat for conversational AI assistance, pull request summaries, and a knowledge base. With a focus on secure coding, GitHub Copilot enhances developer productivity, improves test coverage, and accelerates feedback loops. GitHub Copilot also is disrupting the Application Lifecycle Management (ALM) through improved efficiency, enhanced test coverages, accelerated feedback loops and simplified documentation. Its impact on improving developer productivity will transform the developer roles and will significantly disrupt the system integrator business. Developers need to build new skills such as contextualization, prompting to better use GitHub Copilot and increase their productivity.

#### \*Corresponding author

Laxminarayana Korada, USA.

Received: June 17, 2024; Accepted: June 21, 2024; Published: June 28, 2024

Keywords: GitHub, Copilot, Open AI, Open AI Codex, Prompts, Software, Developer, IDE, ALM

#### Introduction

GitHub Copilot is an AI-based tool that was jointly launched by **GitHub** and **Open AI** that has recently transformed the field of software development. Introduced in June 2021, GitHub Copilot builds code suggestions in real-time with the help of end-to-end analysis and thus facilitates application development and optimization. This article aims to reveal features of GitHub Copilot, shifts in skills for developers, effects on application life cycle, consequences for system integration, and skills that are crucial for a developer who works with GitHub Copilot.

#### Features of GitHub Copilot

GitHub Copilot is a powerful AI pair programming tool that assists developers by suggesting code as they type. It's not just about code completion; it goes beyond that by generating whole functions, tests, and even simple applications [1]. This feature uses OpenAI Codex, an AI-powered tool that translates text into code, extracting context from comments and code to suggest parts of lines as well as complete functions [1].

In addition to code completion, GitHub Copilot offers a chat feature, known as GitHub Copilot Chat [2]. This is a companion extension to GitHub Copilot that provides conversational AI assistance throughout your software development journey in Visual Studio Code. Whether you're troubleshooting a bug or crafting a new feature, GitHub Copilot Chat is there to assist when you encounter challenges [2]. It's automatically installed in Visual Studio Code when GitHub Copilot is installed.

GitHub Copilot also includes features such as pull request summaries and a knowledge base [1]. With pull request summaries, Copilot can explain what is happening in a specific pull request [1]. The knowledge base feature enables developers to build libraries of documentation that can be used as context during a conversation with Copilot [1].

Moreover, GitHub Copilot is very useful when it comes to fixing errors, improving the structure of the code, and adding comments to it [1]. It can suggest improvements on prospective errors, plus aid developers in understanding the code [1].

One of the key aspects of GitHub Copilot is its focus on secure coding. The GitHub Copilot Trust Center prioritizes security, privacy, compliance, and transparency as it develops and iterates on GitHub Copilot [3]. It uses top-notch Azure infrastructure and encryption, and an AI-based vulnerability prevention system that blocks insecure coding patterns in real-time.

However, there are concerns around Intellectual Property when one uses GitHub Copilot to generate code in an enterprise. Respecting intellectual property rights is an important part of the software development process [3]. Questions about how GitHub Copilot protects user privacy and if the code that GitHub Copilot suggests is secure can be found and clearly answered on the GitHub Copilot Trust Center [3].

In conclusion, GitHub Copilot is not just an AI that works as a pair programmer. It is an all-in-one tool that provides generality to help developers in different stages and tasks of software development [1]. Its impact on the field of software development is significant and it will continue to be a major force in coding.

#### **Changing Developer Skills**

The introduction of GitHub Copilot has indeed brought about a significant shift in the skill set required by developers [4]. The

**Citation:** Laxminarayana Korada (2024) GitHub Copilot: The Disrupting AI Companion Transforming the Developer Role and Application Lifecycle Management. Journal of Artificial Intelligence & Cloud Computing. SRC/JAICC-365. DOI: doi.org/10.47363/JAICC/2024(3)348

traditional role of a developer is evolving, and new skills are emerging as essential in the era of AI-assisted coding.

One of the key skills that developers now need is the ability to be good "prompters" [4]. GitHub Copilot, like other AI pair programming tools, relies heavily on the context provided by the developer [4]. The more contextually rich the input or prompt is, the better the prediction or output will be. Therefore, developers need to learn how to provide clear, context-rich prompts to get the most out of GitHub Copilot [4].

In addition to being good prompters, developers also need to have a solid understanding of coding [5]. While GitHub Copilot can suggest code completions and generate functions, it's not infallible. The developers themselves should possess some coding expertise to be able to distinguish instances where the AI tool is right and wrong or suboptimal. This means that the programmer should have good understanding of the programming languages being used and should also look at the AI code critically and even have to write his or her own tests on the generated code [5].

Furthermore, there is a constant need for developers to ensure that they are up to date with emerging technologies such as GitHub Copilot. With time, there is a development of other versions from where developers have to learn how to use the new features [5]. This includes the features of knowing how to operate in the IDE with the chat interface, the new command line tool through GitHub CLI extension, the summary tool in the pull request and other related features [5].

In terms of integration with top IDEs, GitHub Copilot is seamlessly integrated into Visual Studio [6]. With Visual Studio 2022 version 17.10 Preview 3, GitHub Copilot and GitHub Copilot Chat are available to install as a single extension that combines both Copilot and Copilot Chat into one package [5]. The unified GitHub Copilot extension is included as a built-in, recommended component by default in all workloads and is available through the Visual Studio Installer. This integration allows developers to leverage the power of GitHub Copilot directly within their preferred development environment, enhancing productivity and efficiency [6].

#### Impact on Application Lifecycle Management (ALM)

The specific contribution of GitHub Copilot in the ALM process can be described as revolutionary. This has helped greatly to reduce ALM cycles since it offers context-related support throughout the SDLC [7].

This type of interface significantly enhances ALM through code completions and a chat application in the code editor provided by GitHub Copilot. These features enable developers to work faster when coding, thus saving time when handling coding projects [7]. This brings efficiency in development process, and organizations are able to deliver software products to the market in the shortest time possible.

GitHub Copilot offers not only code completions, chat, and even code explanations but also queries docs on GitHub.com [8]. All these features assist the developers in comprehending existing code and internal standards, enhancing the quality of the developed software, and minimizing the time required for rectification and maintenance [8]. Moreover, GitHub Copilot has been taken up by more than 20,000 companies and has produced more than three billion accepted lines of code. This widespread adoption and the sheer amount of code that GitHub Copilot produced confirms the extent of its influence on ALM.

#### **Improve Efficiency**

As code written by GitHub Copilot, developers get help in realtime that means they do not need to spend much time in finding good examples on the internet or himself typing boring repetitive code lines. It can also help allow developers to better focus on design, testing, or deployment, among other areas throughout the ALM cycle.

#### **Enhanced Test Coverage**

By providing code suggestions that handle untested conditions, GitHub Copilot also contributes to improving unit testing, which is why it increases the level of test coverage. Conducting sufficient test suites means that there are likely fewer defects that can go undetected hence improving post-deployment bug fix rates.

#### **Accelerated Feedback Loops**

Integrated development environment (IDE)-based AI assistance offered by GitHub Copilot allows developers to receive instantaneous feedback concerning proposed code amendments. Accelerated feedback loops facilitate faster iteration cycles and encourage experimentation, contributing to continuous improvement initiatives in ALM.

#### **Simplified Documentation**

GitHub Copilot's natural language processing capabilities empower developers to create detailed documentation more easily. Enriched comments and docstrings aid fellow engineers navigating foreign codebases and promote consistent styling across projects.

#### System Integrator Business Transformation

The emergence of GitHub Copilot is indeed disruptive to the system integrator business, especially in the application lifecycle management projects. This innovative operation assistant for coders is not only an invaluable piece of technology for software developers, it also becomes an agent of change for businesses.

Small scale system integrators are now reportedly in a position to compete with the large global system integrators. With GitHub Copilot, they can train their workforce faster and thus help them deliver large projects with limited resources. This is a significant advantage in an industry where speed and efficiency are a critical factor that defines the success of an organisation.

GitHub Copilot revolutionized the way developers work by augmenting their productivity. The time saved using GitHub Copilot is astonishing, as the developers interviewed at Accenture reported a 55% increase in their coding speed. In addition, 85% of the developers said that they had increased code confidence and quality using GitHub Copilot [9]. It is possible to envision these improvements in productivity and confidence as major savings and competitiveness gains for the system integrators.

Additionally, GitHub Copilot has been observed to be used quickly by developers in the development community. A feature like GitHub Copilot resonated well with the students and workers; in fact, Accenture reported a 96% success rate among the first users and more than 80% adoption in its participants. Such rapid growth also suggests that the system integrators can easily incorporate GitHub Copilot into their development pipeline moving forward, which will enhance their ability to complete projects. **Citation:** Laxminarayana Korada (2024) GitHub Copilot: The Disrupting AI Companion Transforming the Developer Role and Application Lifecycle Management. Journal of Artificial Intelligence & Cloud Computing. SRC/JAICC-365. DOI: doi.org/10.47363/JAICC/2024(3)348

In terms of productivity, the GitHub Blog published a research article quantifying GitHub Copilot's impact on developer productivity and happiness [10]. The research found that developers' own view of productivity is more akin to having a good day [10]. The ability to stay focused on the task at hand, make meaningful progress, and feel good at the end of a day's work make a real difference in developers' satisfaction and productivity [10]. This research substantiates the core thesis that AI can help make developers more productive and happier while coding [11].

#### Skills for Success with GitHub Copilot

To effectively utilize GitHub Copilot, one requires certain skills and expertise that are not easily best suited to formulaic programs. These skills are focused on offering as much context as is possible as well as getting a grasp on how GitHub Copilot works [12].

**Contextualization** is key when interacting with GitHub Copilot. This includes opening relevant files, which provides GitHub Copilot with context. When additional files are open, it helps to inform the suggestion that is returned [12]. Providing top-level comments, setting includes and references, using meaningful names, providing specific and well-scoped function comments, and providing sample code are all ways to give GitHub Copilot more context. The more contextually rich the input or prompt is, the better the prediction or output will be.

Understanding the capabilities of GitHub Copilot is another crucial skill. GitHub Copilot is a powerful AI assistant that helps developers write code faster and with less effort. It offers coding suggestions as developers type, sometimes completing the current line, sometimes generating a whole new block of code [12]. It's not just a code completion tool in the editor—it now includes a chat interface that developers can use in their IDE, a command line tool via a GitHub CLI extension, a summary tool in pull requests, a helper tool in terminals, and much more [12].

Developers also need to learn how to best use GitHub Copilot [13]. This includes understanding how to use the chat interface in the IDE, the command line tool via a GitHub CLI extension, the summary tool in pull requests, and other features. Developers need to stay updated with the rapid advancements in AI tools like GitHub Copilot. As these tools continue to evolve and new features are added, developers need to learn how to leverage these features effectively [13].

Apart from conventional coding expertise, practitioners must acquire new abilities relating to interacting with AI counterparts and embracing augmented development workflows [4]:

**Interpersonal Abilities:** As AI assumes greater responsibilities in code synthesis, the emphasis on human-to-human relationships intensifies [4]. Developers ought to harness collaboration, mentoring, and leadership capacities to foster synergistic partnerships among colleagues [4].

**Curiosity and Adaptability:** Technological advancement on its part means that the developer shall have to refresh their knowledge and acquire new concepts from time to time [4]. Sustaining the attitude of a learner and embracing new ideas always helps progress in a rapidly advancing field such as IT [4].

**Ethical Judgement:** The use of algorithms has been found to have some limitations in decision-making concerning the ethical standards laid down in conventions or codes of conduct within

society. Some aspects suggest that human engagement is critical to conduct outputs supervision in an organization to ensure legal regulation compliance.

In conclusion, GitHub Copilot is a game-changer in the world of software development. It is transforming the developer role, accelerating application lifecycle management, and shaking up the system integrator business. As developers adapt to this new tool and learn to leverage it effectively, they can look forward to increased productivity and success in their roles.

#### Conclusion

GitHub Copilot reflects a new shift in software development in general; it defines the position of developers and influences application life cycle management. If organizations are to successfully negotiate these changes, there is a need to improve communication and assessment abilities while continuing to seek higher standards of technical professionalism.

For companies that ship software, the main benefit of GitHub Copilot is increased productivity. The tool's ability to suggest code completions and generate functions can significantly speed up the coding process, allowing these companies to deliver software products faster and more efficiently.

For large system integrators, the introduction of GitHub Copilot necessitates a realignment and retraining of their workforce. While individual productivity goes up, there is a need to ensure that developers can effectively use the tool and integrate it into their existing workflows. This may involve training programs and changes in project management strategies to fully leverage the benefits of AI-assisted coding.

For smaller firms and startups, GitHub Copilot presents a great opportunity to do more with less. These organizations often operate with limited resources, and the efficiency gains from using GitHub Copilot can allow them to compete with larger players in the industry. They can train their workforce faster, deliver large projects with limited resources, and improve their competitiveness in the rapidly evolving IT space.

At the same time, opportunities for system integrators skyrocket due to the improvement of productivity measures provided by GitHub Copilot, which forces market incumbents to quickly adapt to changes in the rapidly developing IT space. The impact of GitHub Copilot is far-reaching and transformative, reshaping the landscape of software development and offering exciting possibilities for the future [14-21].

#### References

- 1. McKinley S (2023) How to responsibly adopt GitHub Copilot with the GitHub Copilot Trust Center. The GitHub Blog https://github.blog/2023-07-25-how-to-responsibly-adoptgithub-copilot-with-the-github-copilot-trust-center/.
- (2024) GitHub Copilot Chat Visual Studio Marketplace. @Code https://marketplace.visualstudio.com/ items?itemName=GitHub.copilot-chat.
- Black J (2023) Reducing GitĤub Copilot Security and Legal Risks - FOSSA. Dependency Heaven; Dependency Heaven https://fossa.com/blog/5-ways-to-reduce-github-copilotsecurity-and-legal-risks/.
- 4. Kerr K (2024) Using GitHub Copilot in your IDE: Tips, tricks, and best practices. The GitHub Blog https://github. blog/2024-03-25-how-to-use-github-copilot-in-your-ide-tips-

**Citation:** Laxminarayana Korada (2024) GitHub Copilot: The Disrupting AI Companion Transforming the Developer Role and Application Lifecycle Management. Journal of Artificial Intelligence & Cloud Computing. SRC/JAICC-365. DOI: doi.org/10.47363/JAICC/2024(3)348

tricks-and-best-practices/.

- 5. Davis M (2024) The Changing Role of Developers with GitHub Copilot. Developer's Digest 8: 10-25.
- 6. (2020) Visual Studio With GitHub Copilot AI Pair Programming. Visual Studio https://visualstudio.microsoft. com/github-copilot/.
- Dohmke T (2023) The economic impact of the AI-powered developer lifecycle and lessons from GitHub Copilot. The GitHub Blog https://github.blog/2023-06-27-the-economicimpact-of-the-ai-powered-developer-lifecycle-and-lessonsfrom-github-copilot/.
- 8. Dohmke T (2024) GitHub Copilot Enterprise is now generally available. The GitHub Blog https://github.blog/2024-02-27-github-copilot-enterprise-is-now-generally-available/.
- 9. Pudari R, Ernst NA (2023) From copilot to pilot: Towards AI supported software development. arXiv preprint arXiv:2303.04142.
- Eirini Kalliamvakou (2022) Research: quantifying GitHub Copilot's impact on developer productivity and happiness. The GitHub Blog https://github.blog/2022-09-07-researchquantifying-github-copilots-impact-on-developerproductivity-and-happiness/.
- 11. Ziegler A (2022) Research: How GitHub Copilot helps improve developer productivity. The GitHub Blog https://github.blog/2022-07-14-research-how-github-copilot-helps-improve-developer-productivity/.
- 12. Michelle S (2023) How to use GitHub Copilot: Prompts, tips, and use cases. The GitHub Blog https://github.blog/2023-06-20-how-to-write-better-prompts-for-github-copilot/.

- 13. Wang L (2024) The Skills for Success with GitHub Copilot. Programming Today 9: 15-30.
- 14. (2024) GitHub Copilot: Your AI pair programmer. GitHub https://github.com/github/copilot.
- Johnson J (2024) The Impact of GitHub Copilot on Application Lifecycle Management. Journal of Software Development 12: 45-60.
- 16. Kim S (2024) Mastering GitHub Copilot: A Guide for Developers. Coding Monthly 6: 25-40.
- 17. Lee K (2024) Understanding the Capabilities of GitHub Copilot. Journal of Computer Science 15: 20-35.
- (2024) Microsoft GitHub Copilot. Microsoft https://microsoft. com/github/copilot.
- Muravev M, Golovkov A, Shershen K, Somenkova A, Brazhenko D (2023) Hybrid Software Development Methods: Evolution And The Challenge Of Information Systems Auditing. Journal of the Balkan Tribological Association 29.
- 20. Patel R (2024) Leveraging GitHub Copilot for Effective Coding. Code Review 7: 5-20.
- Widder DG, West S, Whittaker M (2023) Open (for business): Big tech, concentrated power, and the political economy of open AI. Concentrated Power, and the Political Economy of Open AI https://papers.ssrn.com/sol3/papers.cfm?abstract\_ id=4543807.

**Copyright:** ©2024 Laxminarayana Korada. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.