Github Copilot and the concept of vibe coding
English for Engineers, HS25

Matteo Frongillo

IDEs are software designed for developers, enabling writing and coding in a highly efficient and optimized environment. With the advent of AI technologies, IDEs have begun to include predictive coding features so as to facilitate work and increase productivity. Vibe coding refers to the practice of coding in a relaxed and creative manner, relying on AI's automatic predictive code completion suggestions, consequently eliminating the need for a structured approach before starting code. This report explores the concept of vibe coding on Visual Studio Code, a popular IDE developed by Microsoft that has, by default, GitHub Copilot as its built-in coding assistant, based on various AI models (Microsoft, 2025b).

Studying university subjects efficiently requires optimal time management and and well-structured distribution of workload. This can be achieved by paraphrasing and summarizing study material. As an advanced user of Visual Studio Code, taking notes in LaTeX, which is a typesetting system commonly used for technical and scientific documents (LaTeX Project, 2025), saves me both time and effort compared to traditional study with pen and paper. GitHub Copilot comes into play here, facilitating the note-taking process by analyzing and interpreting the content of the code, thereby effectively predicting the next words that are most likely to be written.

Vibe coding has the potential to completely revolutionize the world of IT and coding in general. By exploiting AI suggestions, developers can focus more on the creative aspects of coding, resulting in a smoother and more intuitive workflow. GitHub Copilot, in particular, has proven to be an effective and intuitive tool for programmers, as the accuracy of its suggestions and ability to understand context improve the quality of work (Microsoft, 2025a). In order to demonstrate Copilot's quality, I will let it suggest the final lines of this paragraph for me:
These capabilities make Copilot suitable for a wide range of applications, from educational tools that provide interactive coding hints and feedback, to rapid prototyping and scaffolding of boilerplate code, automated documentation and test generation, and enhanced accessibility for developers with disabilities. When integrated into team workflows, AI suggestions can speed up code reviews, refactoring, and debugging.

Despite the numerous advantages of vibe coding with GitHub Copilot, one of the main problems associated with the vibe coding is over-reliance on AI generated suggestions. By handing over all the work to the code assistant, developers lose the ability to resolve potential issues due to a partial understanding of the code the person is working on. Nevertheless AI is extremely useful as a writing completion assistant, when it comes to studying or working, it is essential to keep a critical eye on the suggestions provided by the AI and actively work on the code rather than passively accepting AI suggestions.

GitHub Copilot offers the option to choose which AI model to rely on for predictive coding. Among the wide selection, OpenAI's ChatGPT models are the most commonly used for text prediction, as they are known for their excellent natural language capabilities and task understanding, whilst Antropic's Claude models are known for their outstanding coding performance (GitHub Docs, 2025a).

Advances in AI technology and therefore in vibe coding suggest a future where programmers and AI will work closely together, as it will become increasingly common to find AI assistants integrated into IDEs or browsers. Specifically for GitHub Copilot, new AI models will be added to the selection and the existing features will be optimized in order to maximize efficiency and collaboration between programmers and AI (GitHub, 2025).

Both coherence and critical thinking should always be prioritized while vibe coding. Although GitHub Copilot improves productivity and workflow, it is essential to pay attention to potential discrepancies in the suggestions provided by AI (GitHub Docs, 2025b). Programmers are therefore advisable to always evaluate the accuracy of suggestions, as over-reliance could potentially lead to an actual decline in performance and the possibility of what is known as "AI Slop", or "Workslop", i.e., papers or content generated by AI that, while appearing professional, are in fact frivolous and useless (Lee et al., 2025).

In conclusion, vibe coding is a powerful approach to programming that can increase productivity and creativity by benefiting from AI predictions. By selecting the best AI model for the task at hand, the efficiency of a tool such as GitHub Copilot increases dramatically, only if code suggestions are carefully analyzed so as to limit potential future issues, such as misunderstanding of the code or an "AI Slop" result.

References

GitHub. (2025). *Copilot news and announcements.* https://github.com/orgs/community/discussions/categories/copilot-news-and-announcements.

GitHub Docs. (2025a). *AI model comparison.* https://docs.github.com/en/copilot/reference/ai-models/model-comparison.

GitHub Docs. (2025b). *Best practices for using GitHub Copilot.* https://docs.github.com/en/copilot/get-started/best-practices.

LaTeX Project. (2025). *LaTeX – A document preparation system.* https://www.latex-project.org/.

Lee, A., Kellerman, G. R., Hancock, J. T., Niederhoffer, K., Rapuano, K., & Liebscher, A. (2025). Ai-generated "workslop" is destroying productivity. *Harvard Business Review.* https://hbr.org/2025/09/ai-generated-workslop-is-destroying-productivity.

Microsoft. (2025a). *Code completions with GitHub Copilot in VS Code* ["Context" section]. https://code.visualstudio.com/docs/copilot/ai-powered-suggestions#_context.

Microsoft. (2025b). *GitHub Copilot in VS Code.* https://code.visualstudio.com/docs/copilot/overview.

Declarations about AI tools

- "GitHub Copilot" was used to generate words suggestions. https://github.com/features/copilot
- "DeepL" was used as a translator. https://www.deepl.com