Member-only story

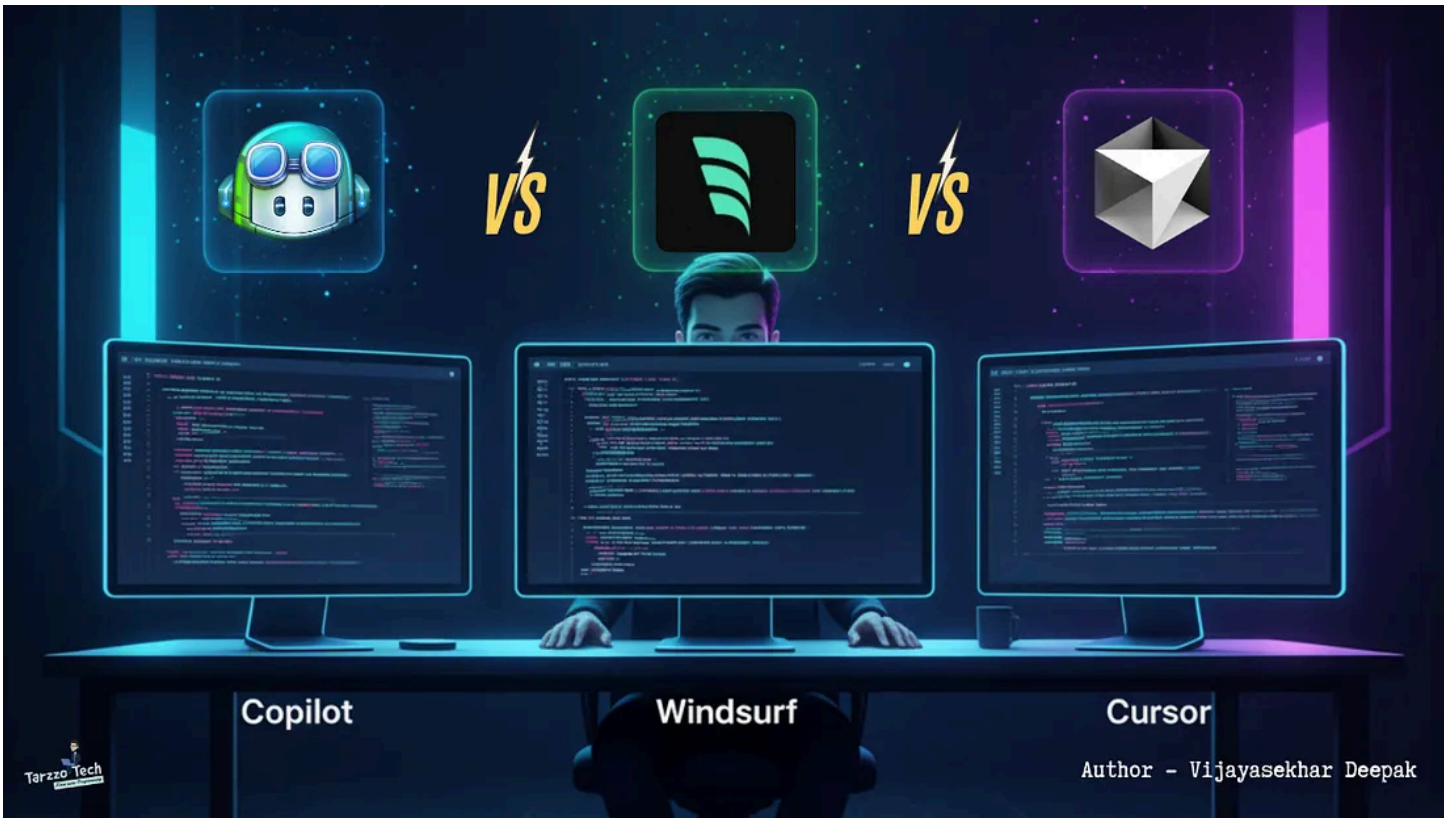
AI TOOLS

🧠 I have tried the Enterprise versions of Copilot, Windsurf, Cursor and here's What I Learned About Each

6 min read · Oct 13, 2025

Vijayasekhar Deepak Follow

Listen Share ••• More



If you're a developer in 2025, chances are you've already tried at least one of the popular AI-powered coding companions like **GitHub Copilot**, **Windsurf**, or **Cursor AI**. Each promises to supercharge your coding workflow, but which one truly *delivers* when you compare features head-to-head?

In this article, I will break down **19 feature comparisons** plus **5 additional VS Code integration capabilities** that I have experienced with working with these tools for past months and highlighting where each one shines and where they fall short. By the end, you'll know which AI coding assistant best fits your development style.

. . .

⚙️ 1. Regenerate in Chat

All three — **Copilot**, **Windsurf**, and **Cursor** — allow you to regenerate AI responses directly inside the chat. This is handy when you want a different version of the same solution without retyping your entire query.

✅ *All three support this equally.*

. . .

⚙️ 2. Auto Model Select Option

Only **Cursor** provides an **Auto Model Selection** feature that intelligently chooses between models like GPT-4o, Claude, or its enterprise LLM for the best results.
Copilot and Windsurf rely on their default or manually set models.

🏆 *Winner: Cursor*

. . .

⚙️ 3. Project or Task Planning

The **Project or Task Planning** option is available only in **Cursor** as of now, which provides features to plan your task if you are going to create a small feature or you can give a outline for a new project and develop a full project plan with Cursor.

🏆 *Winner: Cursor*

. . .

⚙️ 4. Auto Fixing Code by Running

All three tools — **Copilot**, **Windsurf**, and **Cursor** — can automatically fix issues by running the code snippet, analyzing the error, and regenerating a fixed version.

✅ *All are equally capable here.*

. . .

⚙️ 5. Deep Wiki

Windsurf takes the lead here with a **Deep Wiki** feature that allows you to create contextual documentation, link files, and share team insights.
Copilot and Cursor currently lack this depth of built-in documentation handling.

🏆 *Winner: Windsurf*

. . .

⚙️ 6. Rules Creation

All three editors allow developers to define rules or constraints for the code genenration for you project, that is useful for maintaining coding standards or project guidelines. This feature is intrudeced first in Cursor as i know, then every one added to their editors as well.

✅ *Supported by all.*

. . .

⚙️ 7. Best Testcase Writer for React Components

While **Copilot** and **Windsurf** can partially generate test cases, **Cursor** produces more accurate and structured React test files, leveraging a deeper understanding of component logic.

This point is completely based on my experience, it may vary for you.

🏆 *Winner: Cursor*

. . .

⚙️ 8. VS Code Extension Support

Copilot and **Windsurf** fully support all VS Code extensions.
Cursor, while integrated tightly with VS Code, occasionally has compatibility issues with some extensions.

⚖️ *Slight edge to Copilot & Windsurf.*

. . .

⚙️ 9. “Keep/Keep All” Button During Generation

This UX improvement is **exclusive to Cursor**, letting you selectively keep generated suggestions or apply all changes at once when the code generation is in progress. Copilot and Windsurf lack this interactive feature, they allow only to accept the code once the generation is completed.

🏆 *Winner: Cursor*

. . .

⚙️ 10. Terminal in Chat Window

Windsurf and **Cursor** include a **built-in terminal** within the chat window for smoother command execution.
Copilot depends on the regular VS Code terminal.

🏆 *Winner: Windsurf & Cursor*

. . .

⚙️ 11. Updating UI Components from UX Images

Currently, **none** of the three editors can fully update an existing UI component just from an uploaded UX image. It's an area where AI editors still have room to grow.

⚙️ *No winner here — yet.*

. . .

⚙️ 12. Multi-Tab Chat Window

If multitasking is your thing, **Cursor** is your best friend. It allows **multiple chat tabs** so you can handle different projects or contexts simultaneously.
Copilot and Windsurf stick to single-threaded chats.

🏆 *Winner: Cursor*

. . .

⚙️ 13. MCP (Model Context Protocol) Support

All three support **MCP**, enabling integration with other AI tools and external data sources seamlessly.

✅ *Equally supported.*

. . .

⚙️ 14. File or Folder Search for Context

Windsurf stands out with **auto-suggestion**-based context fetching, while Copilot and Cursor rely on manual or separate search commands.

🏆 *Winner: Windsurf*

. . .

⚙️ 15. Limited Context for a Single Chat

Cursor handles large contexts is much better, maintaining a wider context window without truncation, ideal for large projects or enterprise-level apps.

🏆 *Winner: Cursor*

. . .

⚙️ 16. Figma to Code (via MCP)

Copilot offers a surprisingly efficient **Figma-to-code** integration through its MCP support, generating clean component structures.
Windsurf and Cursor don't yet match this ability.

🏆 *Winner: Copilot*

. . .

⚙️ 17. Continuing Context Across Chats

Only **Cursor** allows you to pick up the other chat context where you left off to the new chat context and keeping your conversation history and project state intact.

🏆 *Winner: Cursor*

. . .

⚙️ 18. Restoring to Specific Points

Copilot offers a clear **restore checkpoint button**, while **Windsurf** and **Cursor** allow restoring by simply editing your previous prompts at any level.

🏆 *Slight advantage: Copilot* **(for the dedicated button)**

. . .

⚙️ 19. Revert Icon for Recently Generated Code

Copilot again wins for convenience with a **direct revert button.** Windsurf and Cursor let you manually revert by prompt editing.

🏆 *Winner: Copilot*

. . .

🧩 Additional VS Code Integrations

These features make these AI editors even more powerful inside VS Code:

1.Creating Prompt Files: It lets you store reusable prompt templates for repetitive tasks. Useful in Cursor and Windsurf.

2. Instructions: Add custom instruction sets or “developer profiles” — available in Cursor and Windsurf.

3. Tool Sets: Predefined actions (like lint, format, test) that the AI can trigger automatically. Strongest in Cursor.

4. Modes: Context modes like “Explain,” “Refactor,” or “Debug.” Supported across all three but more refined in Windsurf.

5. Generating Agent Instructions: Create full AI agents with specific purposes or roles. Cursor Enterprise takes the lead here.

✅ **Overall Winner: Cursor,** It’s not just an AI assistant, it’s a full AI coding environment built around collaboration, context, and control.

That said, **Copilot** remains the easiest for quick integrations, and **Windsurf** excels in structured workflows with team-oriented features like Deep Wiki.

. . .

💬 Final Thoughts

Each AI editor brings something unique to the table:

- **Copilot** is your reliable co-coder fast, consistent, and deeply embedded into VS Code.
- **Windsurf** is for teams who value documentation, structure, and collaborative clarity.
- **Cursor** is for developers seeking the next level of context-aware, multi-tab AI development.

. . .

🗣️ Over to You

So, which editor you are using in your office or in your development journey?
Which feture do you love in these editors?

💭 Drop your thoughts in the comments — and if I missed a feature you love, let me know!
I'd love to update this comparison with your insights.

Thank You for Reading!

I hope you found it helpful and informative. If you have any questions or feedback, feel free to leave a comment below. Your support and engagement mean a lot to me.

I appreciate your support. See you in the next blog!

Happy Coding!

A message from our Founder

Hey, Sunil here. I wanted to take a moment to thank you for reading until the end and for being a part of this community.

Did you know that our team run these publications as a volunteer effort to over 3.5m monthly readers? **We don't receive any funding, we do this to support the community.** ❤️

If you want to show some love, please take a moment to **follow me on LinkedIn, TikTok, Instagram**. You can also subscribe to our **weekly newsletter**.

And before you go, don't forget to **clap** and **follow** the writer!

Github Copilot | Windsurf | Cursor | AI | Coding

Follow

Published in Artificial Intelligence in Plain English

32K followers · Last published just now

New AI, ML and Data Science articles every day. Follow to join our 3.5M+ monthly readers.

Follow

Written by Vijayasekhar Deepak

804 followers · 317 following

Hi 🙋 , I write stuffs around Technology 🌐 , Development 👨‍💻 , Software Industry Trends 💻 , and Innovations
#Quote: Change is Only CONSTANT 🕸️

No responses yet

Bgerby

What are your thoughts?

More from Vijayasekhar Deepak and Artificial Intelligence in Plain English

In Python in Plain English by Vijayasekhar Deepak

🧪 UV Cheatsheet for Python Developers

A Beginner-Friendly Guide

Jul 5 576 3

In Artificial Intelligence in Plain English by Simranjeet Singh

RAG is Hard Until I Know these 12 Techniques → RAG Pipeline to 99% Accuracy

RAG is Hard Until I Know these 12 Techniques → RAG Pipeline to 99% Accuracy. Best Blog to Scale or increase RAG Pipelines Accuracy.

Sep 27 498 8

In Artificial Intelligence in Plain English by Simranjeet Singh

OpenAI ML Engineer Interview Questions 2025

A mock interview with an OpenAI ML engineer covering LLM deployment, low-latency inference, quantization, mixed precision, and strategies.

Sep 24 140 7

In Artificial Intelligence in Plain English by Vijayasekhar Deepak

AI vs GenAI vs AGI vs ASI vs OI — The Next Wave of Intelligence

Not a Member? Read for FREE here.

Sep 8 196 1

See all from Vijayasekhar Deepak

See all from Artificial Intelligence in Plain English

Recommended from Medium

In Beyond the Brackets by Stefano Alvares

Mastering GitHub Copilot Custom Instructions and Prompts in VS Code

Turn your chaotic AI intern into a disciplined coding teammate

Aug 25 5

In ITNEXT by Mario Bittencourt

Up your AI Development Game with Spec-Driven Development

Spec Driven Development is new promising way to adopt AI and keep the developer in the loop. I will cover all aspects of SpecKit here.

Oct 20 365 2

In Realworld AI Use Cases by Chris Dunlop

Using AI to make a mobile game in 2 weeks

Tips, Tricks and things learnt using Claude Code in the process.

3d ago 179 1

In Stackademic by Somendradev

10 Niche Developer Tools You Didn’t Know Existed

Let’s be real—the developer world moves fast. Every week, a dozen new tools launch promising to “boost productivity” or “make your life...

Oct 15 185 4

In Level Up Coding by Elye - A Dev By Grace

Handy Tips and Tricks for GitHub Copilot in VS Code

Master GitHub Copilot Beyond Just Vibe Coding Prompting

Oct 21 19

Ondřej Popelka

Generating user documentation from nothing

AI vision and reasoning to transform screen recordings into structured user guides

Oct 19 390 6

See more recommendations