# 01

# Improvements and Limitations of the New Version of GitHub Copilot (2025)

## 1. Introduction

This research explores the improvements and limitations of the latest version of GitHub Copilot, with a particular focus on how it compares to other agentic AI IDEs, such as Windsurf. The goal is to provide a comprehensive, up-to-date, and unbiased overview for technical decision-makers and developers considering advanced AI coding assistants.

---

## 2. Improvements in the New Version of GitHub Copilot

### a. Enhanced Context Awareness

- The new Copilot leverages larger context windows, allowing it to consider more of the surrounding code and project structure.

- Improved ability to infer developer intent from comments and code patterns.

### b. Better Integration with IDEs

- Deep integration with Visual Studio Code, JetBrains IDEs, and Neovim.

- Real-time code suggestions, inline completions, and multi-line code generation.

- Enhanced support for refactoring, code navigation, and documentation generation.

### c. AI Model Upgrades

- Uses the latest OpenAI Codex and potentially GPT-4-class models, improving code generation quality and language support.

- Improved handling of edge cases, rare APIs, and less common programming languages.

### d. Security and Compliance

- Improved filtering of insecure code suggestions.

- Enhanced support for code provenance, licensing checks, and enterprise compliance.

### e. Collaboration Features

- Early versions of Copilot Chat and Copilot Workspace for collaborative coding, debugging, and code review.

---

## 3. Limitations of the New GitHub Copilot

### a. Lack of Full Agentic Autonomy

- Copilot remains primarily a code completion and suggestion tool, not a fully agentic system.

- Cannot autonomously plan, execute, and verify complex multi-step tasks or refactorings.

### b. Limited Workflow Automation

- No built-in support for automated multi-file refactoring, dependency management, or test orchestration.

- Lacks persistent memory or project-wide reasoning.

### c. Dependency on Cloud and Privacy Concerns

- Requires continuous cloud connectivity for best performance.

- Raises concerns for sensitive codebases regarding data privacy and intellectual property.

### d. Limited Customization and Extensibility

- Few options for user-defined workflows, custom agents, or deep integration with CI/CD pipelines.

---

## 4. Comparison with Agentic AI IDEs (e.g., Windsurf)

### a. Agentic Capabilities

- \*\*Windsurf\*\* and similar agentic IDEs can autonomously plan, execute, and verify entire coding tasks, not just suggest code.

- Support for multi-step task management, debugging, and error recovery.

### b. Memory and Project Context

- Agentic IDEs maintain persistent memory of project state, past actions, and user preferences.

- Copilot has limited or no persistent memory across sessions.

### c. Automation and Orchestration

- Agentic IDEs can run scripts, tests, and builds, and can reason about their results.

- Copilot cannot initiate or manage such workflows autonomously.

### d. Extensibility and Customization

- Agentic IDEs allow for custom rules, workflows, and integration with external tools.

- Copilotâ€™s customization is limited to prompt engineering and minor settings.

---

\*\*(Continued in chunk\_02.md)\*\*

# 02

# Improvements and Limitations of the New Version of GitHub Copilot (2025)

## 5. Additional Considerations

### a. Ecosystem and Community

- GitHub Copilot benefits from a massive user base, frequent updates, and direct integration with the GitHub platform.

- Agentic IDEs like Windsurf, while powerful, may have a smaller community but often offer deeper customization for advanced users and teams.

### b. Pricing and Accessibility

- Copilot is a paid subscription service, with discounts for students and open-source contributors.

- Some agentic IDEs may be open source, enterprise-licensed, or have tiered pricing models.

### c. User Experience

- Copilot emphasizes seamless, non-intrusive suggestions, aiming to augment rather than automate.

- Agentic IDEs may require more onboarding but provide more control and automation for power users.

---

## 6. Summary Table: Copilot vs. Agentic IDEs (Windsurf)

| Feature | GitHub Copilot | Windsurf (Agentic IDE) |

|------------------------------|--------------------------------|-------------------------------|

| Code Completion | Yes (state-of-the-art) | Yes (state-of-the-art) |

| Agentic Task Automation | No | Yes |

| Persistent Project Memory | No | Yes |

| Multi-file Refactoring | Limited | Yes |

| Custom Workflows | Minimal | Extensive |

| IDE Integration | Excellent (VSCode, JetBrains) | Good (varies by IDE) |

| Collaboration Tools | Early features | Power-user workflows |

| Privacy Controls | Improving | User-controlled |

| Pricing | Paid (subscription) | Varies |

---

## 7. References and Further Reading

- [GitHub Copilot Documentation](https://docs.github.com/en/copilot)

- [Windsurf AI IDE Overview](https://windsurf.ai)

- [OpenAI Codex Papers](https://openai.com/research/publications/codex)

- [Agentic AI IDEs: Next-Gen Coding Assistants](https://arxiv.org/abs/2309.00867)

---

\*End of research summary. For detailed discussion or updates, see the respective project documentation and changelogs.\*