

Vergleichender Überblick GitHub Copilot Agenten und OpenAI Codex

1. Architektur & Ausführungskontext

GitHub Copilot Agent Mode

- Läuft lokal im VS Code-Insiders oder Stable mit “Agent”-Modus, nutzt CPU/des Entwicklergeräts oder einen Codespaces-Container.
- Arbeitet teil-autonom: Entwickler startet den Task mit Prompt, Agent erkennt Unteraufgaben, kompiliert, testet und iteriert eigenständig.
- Ausführungskontext reicht von lokalem Rechner bis hin zu GitHub Actions in Kombination mit Issue-Zuweisung – insbesondere beim Coding Agent (Pro+/Enterprise), wo das System in GitHub Actions-VMs läuft.

Project Padawan

- Voll-automatisch innerhalb GitHub: Zuweisung via Issue → Hintergrund-Ausführung per GitHub Actions. Entwickler wird später zum Review eingeladen.
- Läuft cloud-basiert, übernimmt Aufgaben komplett autonom.

OpenAI Codex Agent

- Cloud-basiert in isolierten Sandbox-VMs von OpenAI, über ChatGPT-Sidebar oder Codex-CLI ausführbar.
- Arbeitet multi-parallel: kann mehrere Tasks gleichzeitig ausführen, indem es Tests nutzt, Code schreibt, committet und Pull Requests vorbereitet.

2. Modell-Backend & Steuerung

System	Modell-Backend	Modell-Konfigurierbar?	Arbeitsweise
Copilot Agent Mode	GPT-4o, GPT-4.1, GPT-o1 (Preview), GPT-o3-mini, Claude 3.5 Sonnet, Claude 3.7, Claude 3.7 Thinking, Gemini 2.0 Flash, Gemini 2.5 Pro (Preview)	Ja	Plant intern, iteriert autonom
Project Padawan	Vermutlich identische Backend-Modelle wie Agent Mode	Nein	Plant & handelt autonom
OpenAI Codex Agent	codex-1 (o3-Variante)	Nein	Arbeitet direkt, erstellt Tasks, commitet, testet

- **Copilot Agent Mode** bietet Modellpicker – wechselbar nach Geschmack.
- **Planungsstil:** Copilot Agent Mode und Padawan erstellen intern Sub-Agenten/Pläne vor Ausführung. Führen Code mit Tests iterativ aus.
- **Codex Agent** erhält Task, plant intern, führt Code mit Tests iterativ aus.

3. Funktionen & Grenzen

GitHub Copilot Agent Mode / Coding Agent / Padawan

- *Feature-Building, Refactoring, Bugfixen, Test-Generierung, Dokumentation, CI/CD-Kommandos, Integration neuer Bibliotheken.*
- **Coding Agent:** Zuweisung von Issues → VM bootet → Codeänderung via Draft PR + Test/Lint-Überprüfung.
- **Volle MCP-Unterstützung** in VS Code. Ca. 1500 Tools auf dem GitHub MCP-Server
- **Padawan:** Autonom, eigenständige Issue-Abwicklung inklusive Tests & Pull Requests.
- Grenzen: geeignet für low-bis-medium complexity, bei Hochkomplexität menschlicher Input weiterhin nötig; GitHub-Policies, Rate-Limits und Action-Minuten gelten.

OpenAI Codex Agent

- *Feature-Erstellung, Bugfixes, Fragen zum Code, Test-Runs, Cross-Pull-Requests.*
- Sandbox-Limitationen: Internetzugang beschränkt, Task-Dauer z. B. Setup max 10 min, Diffs limitiert, Rate-Limits geplant.

4. Zugriffsmodell & Preisgestaltung

GitHub Copilot

- *Agent Mode:* in allen IDE's verfügbar; Coding Agent: alle Kostenpläne, Free Plan verfügbar; Copilot Pro+ & Enterprise: Administrator-Freischaltung nötig.
- *Padawan:* auf der GitHub Website (arbeitet in der Cloud), Aufträge können von VS Code übergeben werden; Nur in Copilot Pro+ & Enterprise: Administrator-Freischaltung nötig.
- Nutzen erfolgt durch Verbrauch von GitHub Actions-Minuten und Premium-Copilot-Requests.
- Preise richten sich nach bestehenden Copilot-Plänen (Free, Pro: 10 USD/Mon, 100 USD/Jahr; Pro+: 39 USD/Mon, 390 USD/Jahr).

OpenAI Codex Agent

- Verfügbar für ChatGPT Pro, Team, Enterprise, Plus bald.
- Anfangs "generous access", später credit-basiert, bestimmte Rate-Limits.

5. Sicherheit & Datenschutz

Copilot

- Läuft in isolierter GitHub Actions-VM/Container. Policies (Branch-Protection, Review-Checks) greifen analog zu menschlicher Mitarbeit.
- Internetzugriffe streng limitiert, Vorgaben über MCP-Server.
- Logs, Commits, Tests nur nach Reviewer-Freigabe ins Haupt-Repo integriert.

Codex Agent

- Sandbox-VMs mit eingeschränktem Internet, nur genehmigte Abhängigkeiten.
 - Auditable Logs, Test-Ergebnisse, Terminal-Output nach Taskende transparent bereitgestellt.
-

6. Alleinstellungsmerkmale & Unterschiede

Copilot Agent Mode vs Project Padawan

- *Agent Mode*: Innerhalb Editor, interaktives Iterieren, Entwickler als Manager.
- *Padawan/Coding Agent*: Vollwertiges Teammitglied in GitHub Repos. Vollständige Issue-Abwicklung in der Cloud, minimaler menschlicher Input bis zum Review. Asynchrone Tasks in GitHub Actions Sandbox.

OpenAI Codex Agent

- Multi-parallel, cloud-basiert, unabhängig von GitHub, über ChatGPT/CLI steuerbar.
- Fokussiert auf asynchrone Tasks und Code-Sandbox-Ausführung.
- Stärke: gleichzeitiges Abarbeiten mehrerer Tasks und strukturierte Sandbox mit Logs & Tests.

Zielgruppen

- *Copilot Agent Mode*: Entwickler, die im IDE bleiben, interaktive Hilfe oder allein durch Anweisungen in natürlicher Sprache programmieren möchten (Vibe Coding).
 - *Padawan/Coding Agent*: Teams mit klaren Issues, automatisierte PR-Erstellung und Review-Flow.
 - *Codex Agent*: ChatGPT-Nutzer, die Software-Engineering-Aufgaben außerhalb von GitHub-Projekten automatisieren und parallelisieren möchten.
-

Fazit

- **Architektur**: Copilot bietet lokale und cloud-basierte Varianten, Codex läuft ausschließlich in der Cloud.
- **Modelle**: Copilot nutzt eine große bis sehr große Auswahl (GPT-4o, o3, Claude, Gemini, usw.) Codex setzt auf codex-1 (o3).
- **Funktionen**: Alle drei decken Feature-Entwicklung, Tests und Bugfixing ab – Codex punktet mit paralleler Taskausführung.
- **Security**: Sehr ähnliche Sandboxing und Policy-Einhaltung in isolierten Umgebungen.
- **Zugriff**: Copilot Agent stark limitiert auch in der Free Version nutzbar, Codex für ChatGPT-Pro und höher.

Downloadlink:

https://joerg-walkowiak.de/downloads/Vergleichender_Ueberblick_GitHub_Copilot_Agenten_und_OpenAI_Codex.pdf