



To discuss this course and customizations:
Call: 434-509-5680 or Email: sales@cloudcontraptions.com

Choosing Your AI IDE: Cursor, Windsurf, Zed, and VS Code Copilot Compared

Class Duration

14 hours of live training delivered over 2-3 days to accommodate your scheduling needs.

Student Prerequisites

- Professional software development experience
- Basic familiarity with at least one of the covered editors

Target Audience

Software engineers, engineering managers, and DevEx/platform teams evaluating which AI-native editor or assistant to standardize on. Equally useful for developers torn between multiple tools and for organizations trying to avoid fragmented tooling across teams.

Description

Rather than teaching one editor in depth, this course provides a structured evaluation framework and live comparison of the four leading AI IDEs and assistants: Cursor, Windsurf, Zed, and VS Code with GitHub Copilot. We assess each tool across a consistent set of dimensions: inline generation quality, multi-file and agent capabilities, context management, model flexibility, team/enterprise features, privacy controls, pricing, and ecosystem maturity. Each editor receives roughly two hours of dedicated coverage with hands-on time, followed by extended head-to-head labs in which participants complete identical representative engineering tasks across all four tools. Participants leave with a documented decision framework, a populated scorecard for their own organization, and a rollout plan for the chosen tool.

Learning Outcomes

- Apply a consistent evaluation framework to AI IDE and assistant comparisons.
- Compare Cursor, Windsurf, Zed, and VS Code Copilot across capability, cost, and organizational fit dimensions, with hands-on experience in each.



To discuss this course and customizations:
Call: 434-509-5680 or Email: sales@cloudcontraptions.com

- Configure rules files, context controls, and agent settings to materially change the quality of each tool's output.
- Assess enterprise and team features: SSO, policy controls, telemetry, audit logging, and data residency.
- Identify which tool best fits a given team composition, codebase size, and workflow.
- Articulate migration costs and risks when switching from one tool to another, including project file portability and habit transfer.
- Create a repeatable scorecard for evaluating new AI tooling releases.
- Produce a phased rollout plan including pilot group, success metrics, and decision review checkpoints.

Training Materials

Comprehensive courseware is distributed online at the start of class. All students receive a downloadable MP4 recording of the training.

Software Requirements

Students should have all four covered tools installed for hands-on comparison labs (Cursor, Windsurf, Zed, and VS Code with GitHub Copilot). Free tiers are sufficient. The OpenAI Codex IDE extension is also used in two of the editors during the cross-editor segment.

Training Topics

Evaluation Framework

- Dimensions: generation quality, agent capability, context management, model flexibility, team controls, privacy, ecosystem maturity, pricing
- How to run a fair side-by-side comparison: controlling for prompt, context, and model
- Task-based evaluation design and scorecard construction
- Common evaluation pitfalls: cherry-picked demos, unfair context advantages, model swapping

VS Code + GitHub Copilot

- Feature overview: suggestions, chat, edits, agent mode, Spaces
- Custom instructions, `.github/copilot-instructions.md`, and prompt files



To discuss this course and customizations:
Call: 434-509-5680 or Email: sales@cloudcontraptions.com

- MCP server integration and extension ecosystem
- Enterprise controls: Copilot Business/Enterprise, content exclusion, audit log, data retention
- Hands-on: configure instructions and run a multi-file edit task
- Strengths, limitations, and best-fit profile

Cursor

- Composer, Agent mode, and Background Agents
- Rules files (.cursor/rules), .cursorrules legacy, and project context
- Context controls: @ references, codebase indexing, and ignore patterns
- Model agnosticism and backend switching (Anthropic, OpenAI, Google, custom)
- Privacy mode and team/enterprise plans
- Hands-on: configure rules and exercise Composer on a multi-file refactor
- Strengths, limitations, and best-fit profile

Windsurf (Cognition)

- Cascade agent and flow-state design philosophy
- Cognition's July 2025 acquisition: Windsurf as part of the combined Devin + Windsurf platform (Windsurf 2.0 with the Agent Command Center shipped April 2026, including Arena Mode for side-by-side model comparison)
- SWE-1.5: Cognition's frontier coding model now powering Cascade
- Agent Command Center for unified Devin + Cascade orchestration
- Windsurf Rules, workspace configuration, and Memories
- Context engine and codebase awareness
- Hands-on: drive Cascade through a multi-step task with rules in place
- Strengths, limitations, and procurement implications of the consolidation

Zed

- Collaborative editing model and performance profile
- Zed AI: assistant panel, inline assist, and slash commands
- Edit Predictions and the Zeta model
- Context servers (MCP) and extension ecosystem maturity
- Hands-on: complete a feature task using assistant panel and inline assist



To discuss this course and customizations:
Call: 434-509-5680 or Email: sales@cloudcontraptions.com

- Strengths, limitations, and best-fit profile

Cross-Editor: OpenAI Codex IDE Extension

- Codex as an extension that runs inside VS Code, JetBrains, Cursor, Windsurf, Xcode, and Eclipse
- When to run Codex alongside the editor's native AI vs. as an alternative
- Hands-on: install Codex inside two of the covered editors and compare outputs
- Implications for the editor selection decision (covered in depth in the dedicated Codex in Practice course)

Extended Head-to-Head Labs

- Lab 1: Implement a small feature from a written spec across all four tools
- Lab 2: Debug a multi-file regression with limited context
- Lab 3: Generate a test suite for an existing module
- Lab 4: Refactor across packages with rules and instructions in place
- Lab 5: Perform a code review with each tool's review/agent capabilities
- Scoring against the framework and group discussion of tradeoffs

Organizational Decision Factors

- Team size, workflow standardization, and one-tool vs. multi-tool policies
- Privacy, compliance, and data residency requirements
- IP, licensing, and code-suggestion attribution risk
- Migration effort and switching costs (rules portability, habit transfer)
- Procurement, licensing tiers, and seat economics
- Telemetry, audit, and incident response readiness

Decision Workshop and Rollout Plan

- Scorecard completion against the participant's organization
- Pilot group selection and success-metric definition
- Phased rollout, decision review checkpoints, and exit criteria
- Group presentations and instructor feedback
- Q&A session