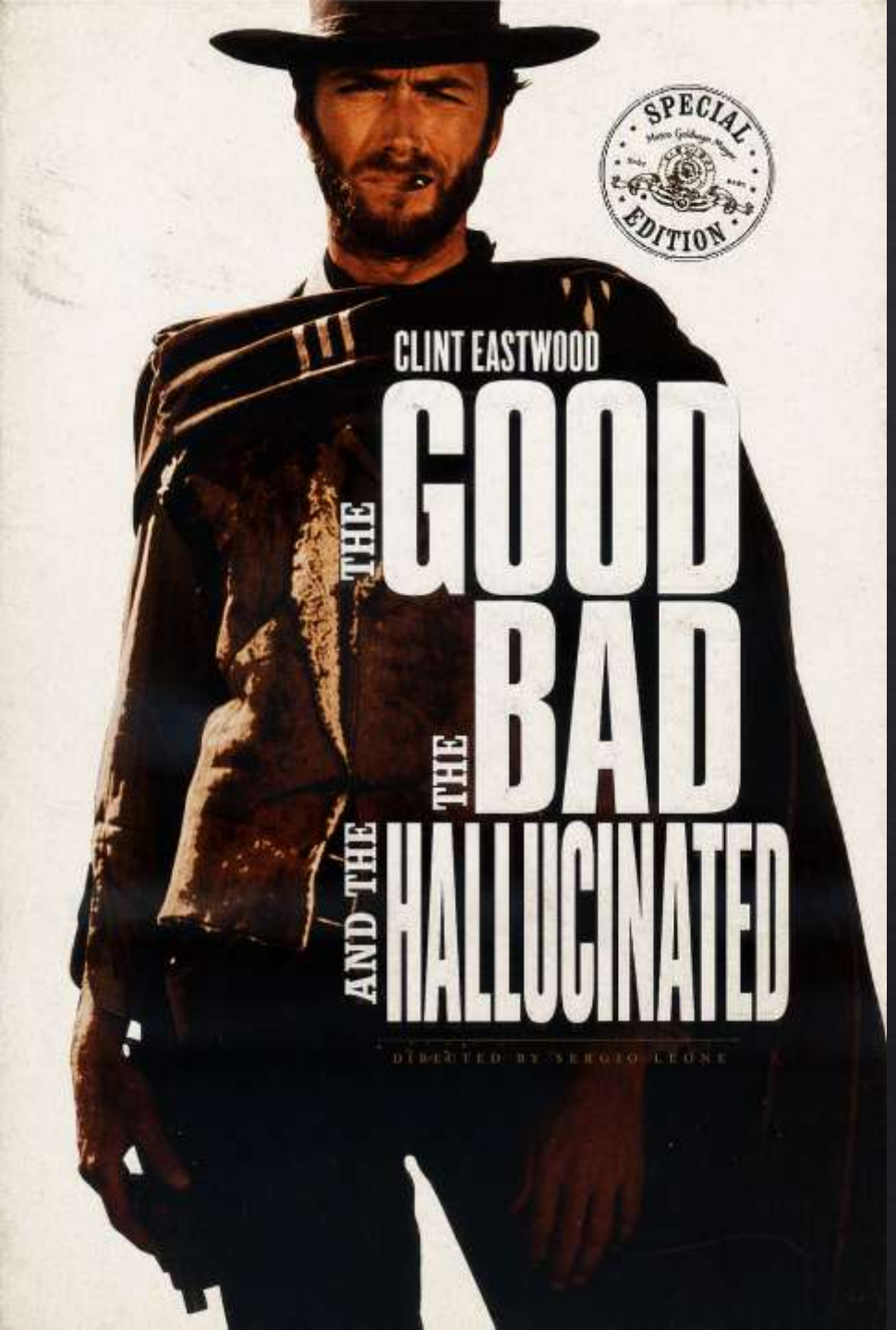


THE GOOD, THE BAD, AND THE HALLUCINATED:

AI Coding in 2026

NUNUG User Group - July 9, 2026

Key Players

- OpenAI
  - Models: GPT, O-series
  - Apps: ChatGPT, Codex, DALL-E

- Anthropic
  - Models: Claude Haiku, Sonnet, Opus
  - Apps: Claude Web, Code, Cowork

- Google
  - Models: Gemini Flash, Pro, Ultra
  - Apps: Gemini, NotebookLM

- SpaceXAI (formerly xAI)
  - Models: Grok

- Meta
  - Models: Llama

- Amazon
  - Models: Nova

- Microsoft
  - Apps: Microsoft Copilot, GitHub Copilot
  - Models: Phi
- AI Coding IDEs
  - Cursor, Windsurf, Jet Brains AI Assistant, Replit Agent
- CLIs
  - Claude Code, Gemini CLI, Codex CLI, GitHub Copilot CLI
- Autonomous Agents
  - OpenClaw (aka ClawdBot, MoltBot), LangGraph, AutoGen, CrewAI, OpenHands, Devin
- Open Source Model Repositories
  - Hugging Face, GitHub, Ollama, ModelScope

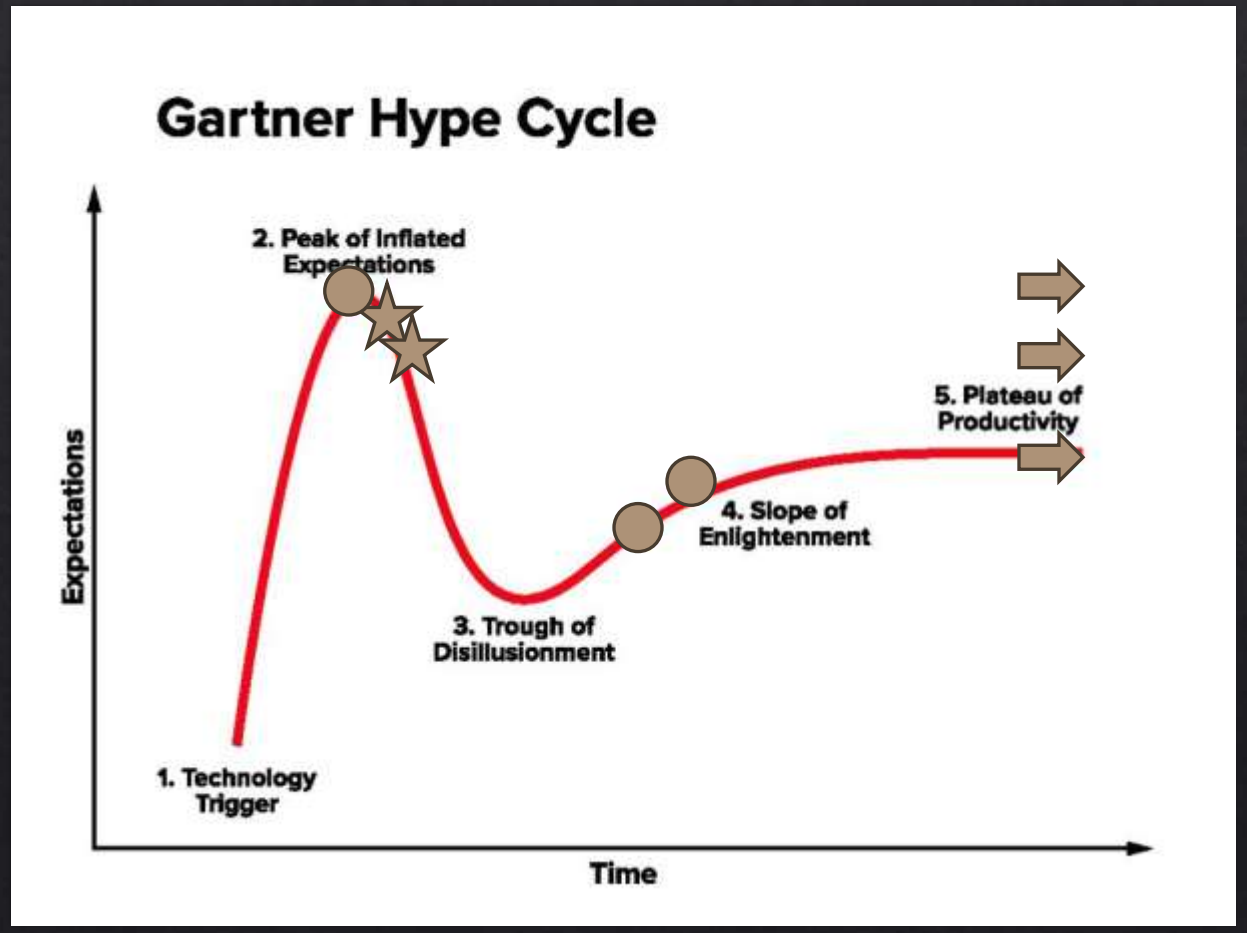
Terms & Definitions

- **Agent (or Agentic AI)**: AI system that can plan, reason, use tools, and take actions toward a goal.
- **Agent Harness:** Runtime environment that connects an agent to models, tools, memory, and permissions.
- **Agent Steering Files**: Persistent instructions that guide AI behavior within a project (e.g., AGENTS.md, CLAUDE.md, copilot-instructions.md).
- **Context Window**: Maximum amount of information a model can consider at one time.
- **Hallucination**: AI-generated content that is plausible but incorrect or unsupported.
- **MCP**: Model Context Protocol; a standard for connecting AI models to tools and data sources.
- **Model (or LLM)**: Trained AI system that processes input and generates output.
- **Permissions**: Resources and actions an AI is authorized to access or perform.
- **Prompt**: Input provided to guide an AI's response or behavior.
- **RAG**: Retrieval-Augmented Generation; using external knowledge retrieved at runtime to improve responses.
- **Request**: A single interaction submitted to an AI system.
- **Shadow AI**: Use of AI tools without organizational approval or governance.
- **Skills**: Reusable instructions or workflows for specific tasks.
- **Tokens**: Units of text used by AI models to process input and generate output.
- **Tool Use**: Ability of an AI to invoke external systems such as APIs, files, or commands.

Current State of AI

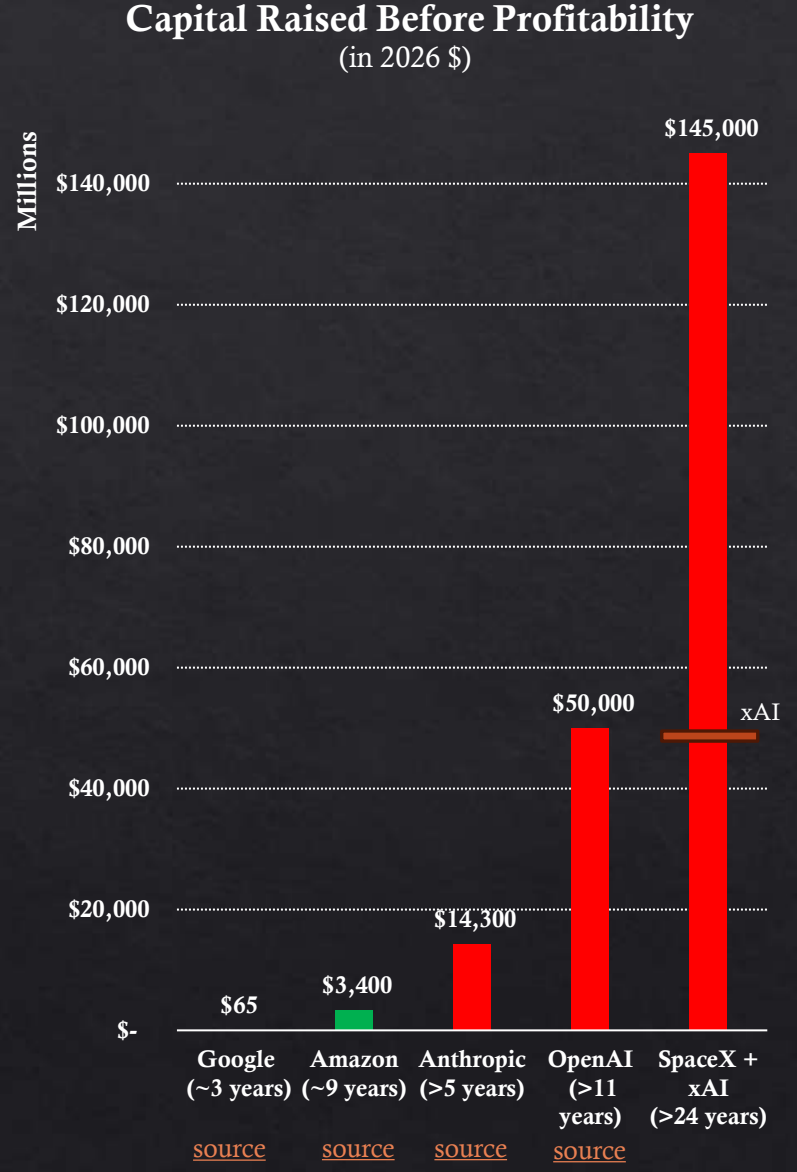
AI Hype

- Where do you think the **market** is on the Hype Cycle? ★
- Where are you **personally** on the Hype Cycle? ●
- Where do you think Productivity will Plateau? ➡

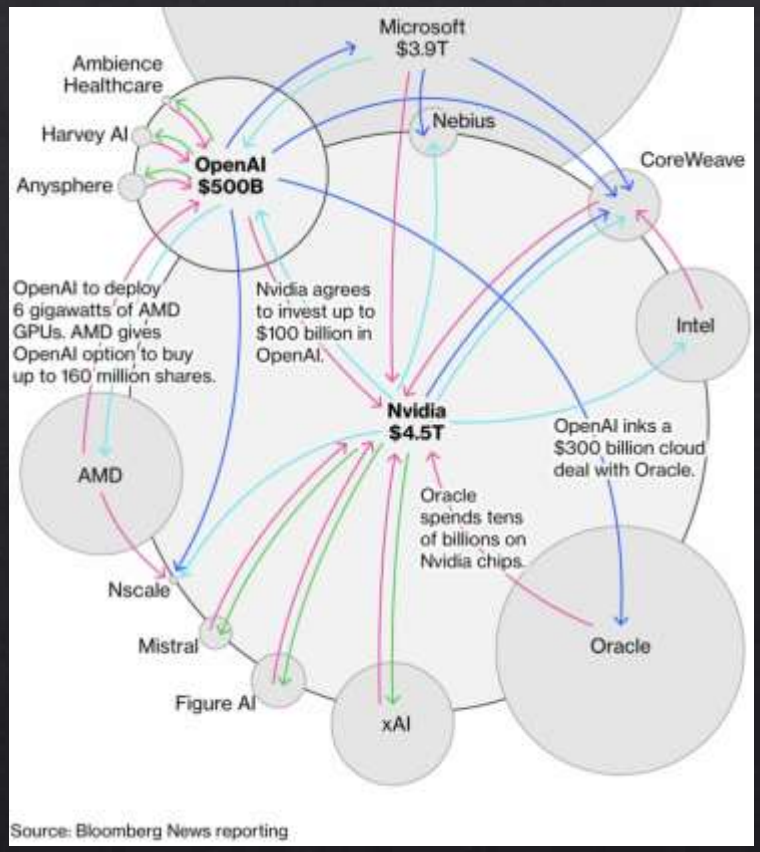


AI Companies

- Capital Investment to Profitability
- Circular Investment Structure
- Ongoing Costs



https://www.softwarethug.com/posts/ai-company-losses-vs-amazon-facebook-path-to-profitability/



https://www.bloomberg.com/graphics/2026-ai-circular-deals/

AI Perceptions vs. Reality

- AI code is “cheap”, but Software is still expensive
  - Code != Software
  - Speedup claims underestimate other activities (e.g., requirements, coordination, integration)
  - Domain knowledge and long context
- Current AI systems often give you an 80% solution
  - 20% left is often difficult due to integration, security, and architecture challenges
  - Edge cases often not included
- Open-Source Impact
- The biggest proponents of AI adoption are those preparing for an IPO:
  - Anthropic CEO states software engineers are 6 - 12 months from being obsolete (Jan 2026)
  - Sam Altman (OpenAI), lacks coding skills or machine learning understanding [The New Yorker]
  - Solution looking for a Problem
- Latest Version of Models have improved substantially from previous generation (Dec 2025 – March 2026).
  - METR 2025 Study
    - 19% slowdown for experienced devs
  - METR 2026 Study
    - ~18% improvement for experienced devs
    - ~4% improvement for inexperienced devs

AI Perceptions vs. Reality

- "Last Mile" Problem with AI
  - Transforming entire business around AI is a challenge
  - Small pockets of tasks may greatly improve productivity, but they aren't aligned with people's roles
  - Integration between processes and systems is hard
  - Tribal knowledge isn't documented or available for AI
  - Process debt and edge cases
- "CEOs are uniquely prone to AI psychosis because they're sufficiently distant from the last mile of work that still has to happen to generate most value with AI," Aaron Levie (Box founder)
- "Something Big Is Happening" (Feb 2026)
  - Matt Schumer, CEO of Applied AI
  - Impacted markets and large software companies
  - Fear → AI Sales
- Rebuttal by Paulo Carvao
  - Senior Fellow at Harvard
  - Uneven and mixed adoption more likely

"Individually, every team built a masterpiece."

AI Perceptions vs. Reality

- Redefinition of Software Development
  - Shift from writing code to system design, standards
  - More formal guardrails and better unit tests
  - Additional code and architectural review and enforcement
- AI is a Development Tool
  - Not a silver bullet
  - Works well on common problems
  - Good at grunt work (rinse & repeat)
- Experienced Software Engineering is Needed Even More
  - AI makes assumptions
  - Existing code lacks proper context and design decision knowledge
  - AI can amplify bad designs
- AI Displacement of SaaS
  [John Pestana co-founder of Omniture]
  - Thin middle-tier SaaS companies may lose out to AI
    (build vs. buy decision)
  - Foundational infrastructure systems will survive

AI Shortcomings & Challenges

- Code Maintenance Challenges
  - More Technical debt
  - More Cognitive debt
  - More Code duplication
  - Big ball of mud
- Security Vulnerabilities
- Verification & Validation
  - New bottleneck

- Vibe Coding Paralysis
  - **Completionist Trap:** Feel productive without actually completing real functionality
  - **Planning Loop:** spend so much time planning and specifying that it becomes procrastination in disguise
  - **Context Collapse:** Split of focus between lots of different features and AI sessions
  - **Confidence Spiral:** loss of trust in the code and your own judgement.
- Human /AI Role Reversal
  [The Reverse Centaur's Guide to Life After AI]
  - Centaur: a human assisted by a machine.
  - Reverse Centaur: a human who is conscripted into acting as an assistant to a machine.

Work AI Institute Survey

- The survey found that for every hour a worker spends getting useful output from AI, they spend roughly another hour making it usable.
- Of the total time workers spend interacting with AI each week,
  - 37% goes to Botsitting
  - 36% to actually using the tool to produce work
  - 27% learning and building agents
- Botsitting → Botshitting
  - Botshitting (n.) The act of shipping AI-generated work that workers haven't verified, don't fully understand, or can't confidently stand behind.

[Source: Work AI Institute]

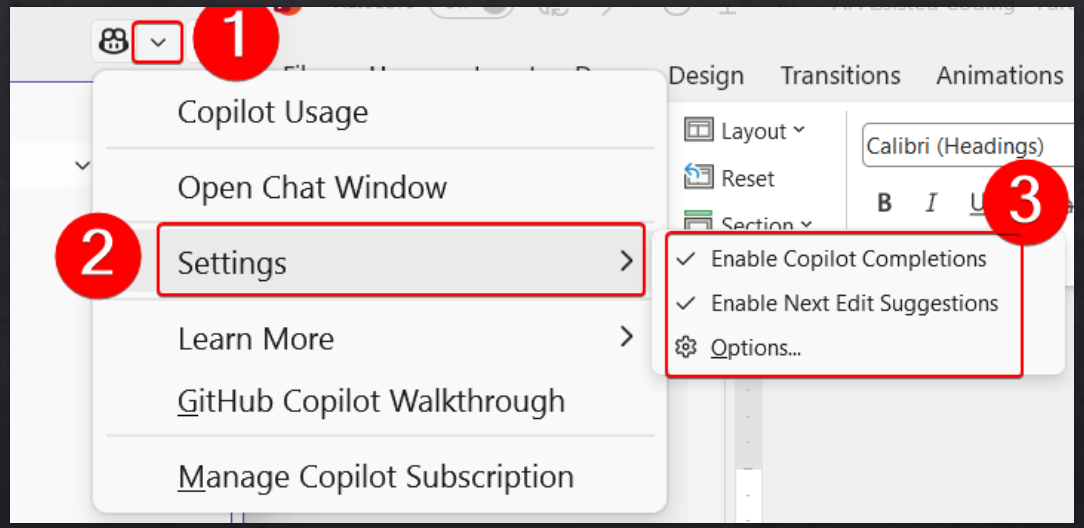
Copilot Tools

- Microsoft Copilot Chat (M365 Copilot)
  - General Chat
- GitHub Copilot in Visual Studio
  - Inline Suggestions
  - Inline Chat
  - Chat Ask, Plan, or Agent modes
- GitHub Copilot CLI
  - Terminal based coding
  - Ask, Plan, or Agent Modes
  - Multiple tabs with worktrees
- GitHub Copilot Web
  - Create PRs for Simple Issues
  - Review code or PRs
  - Integrates with GitHub Mobile app

Visual Studio

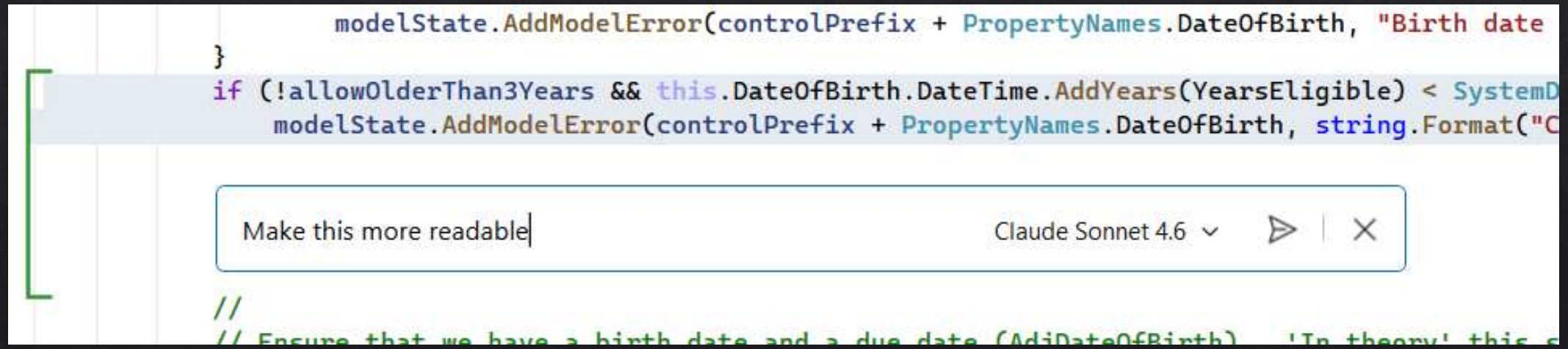
Visual Studio: Copilot Settings

- Copilot Down Arrow > Settings
- **Copilot Completions:** inline code as you type
- **Next Edit Suggestions:** sections to change next, based on what you are doing
- **Options…**
  - Planning
  - Custom Instructions (copilot-instructions.md)



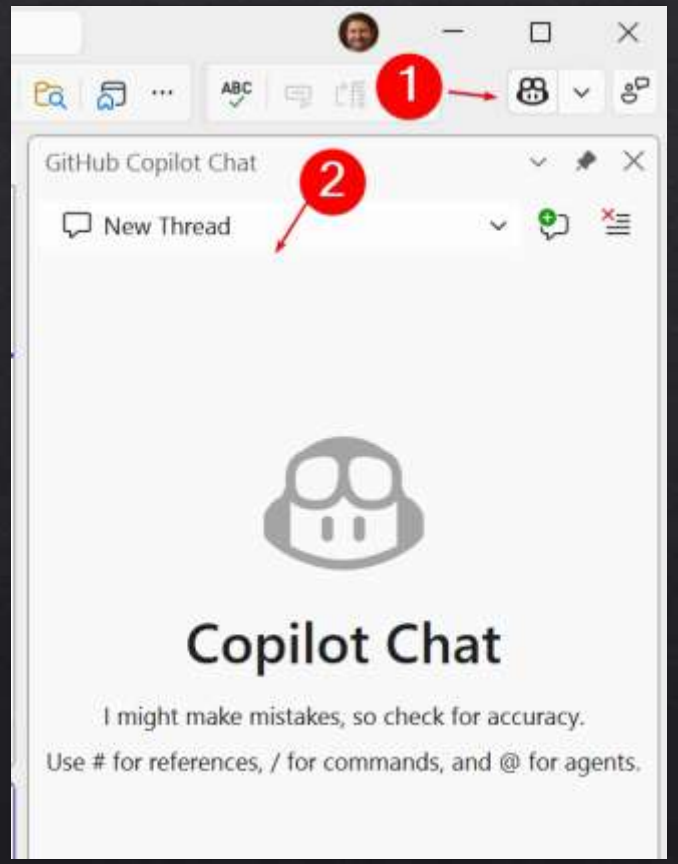
Visual Studio: Copilot Inline Chat

- Select code you want modified
- Activate Inline Chat: ALT+/
- Good for precise changes



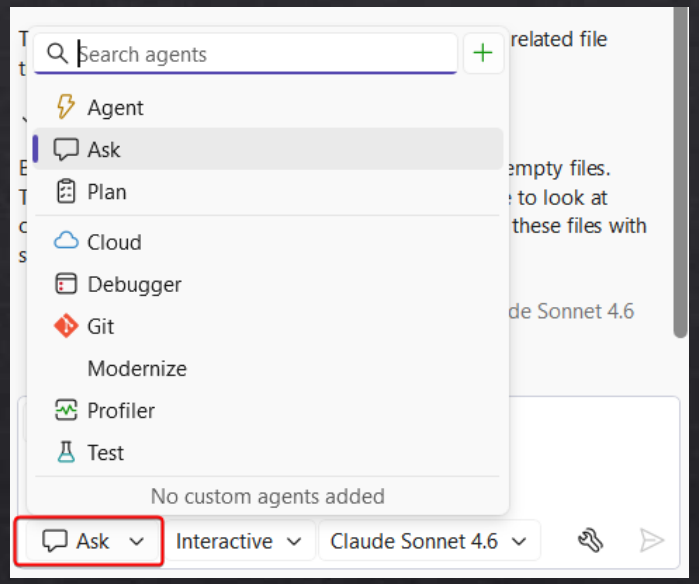
Visual Studio: Copilot Chat Window

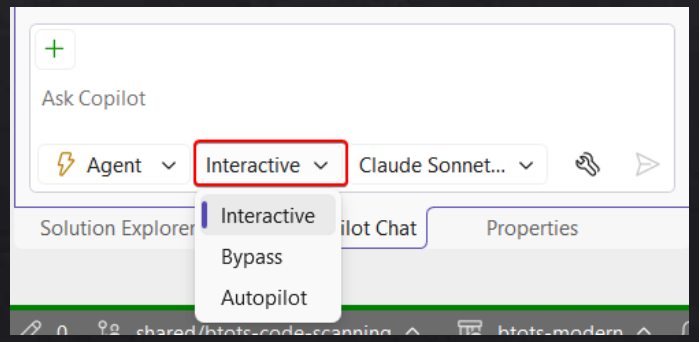
- Top Right Copilot Toolbar Button
- View > GitHub Copilot Chat
- Hotkey: CTRL+\,C

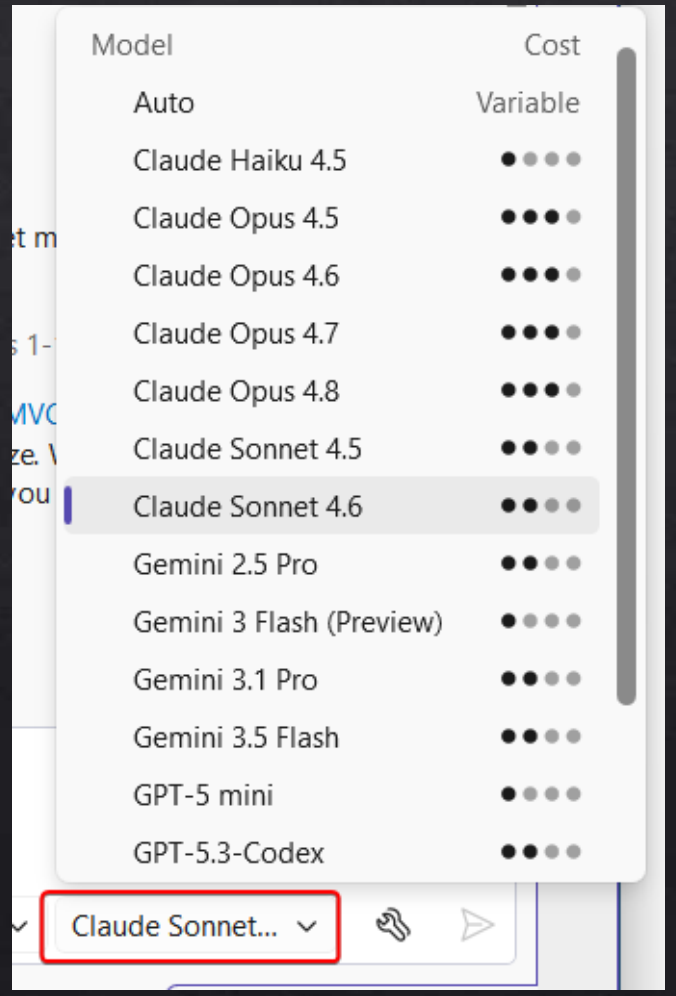


Visual Studio: Copilot Agent Mode & Models

- Agents
- Mode
- Models

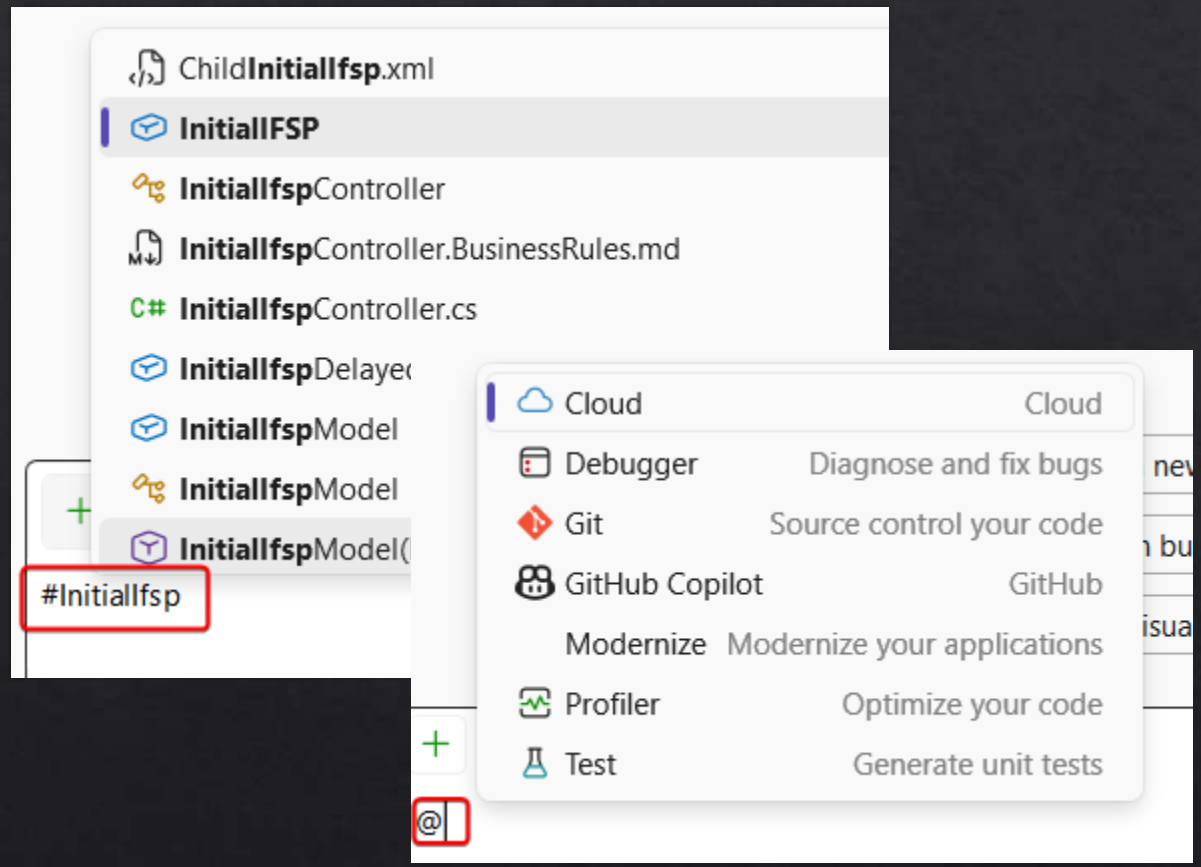






Visual Studio: Copilot Commands

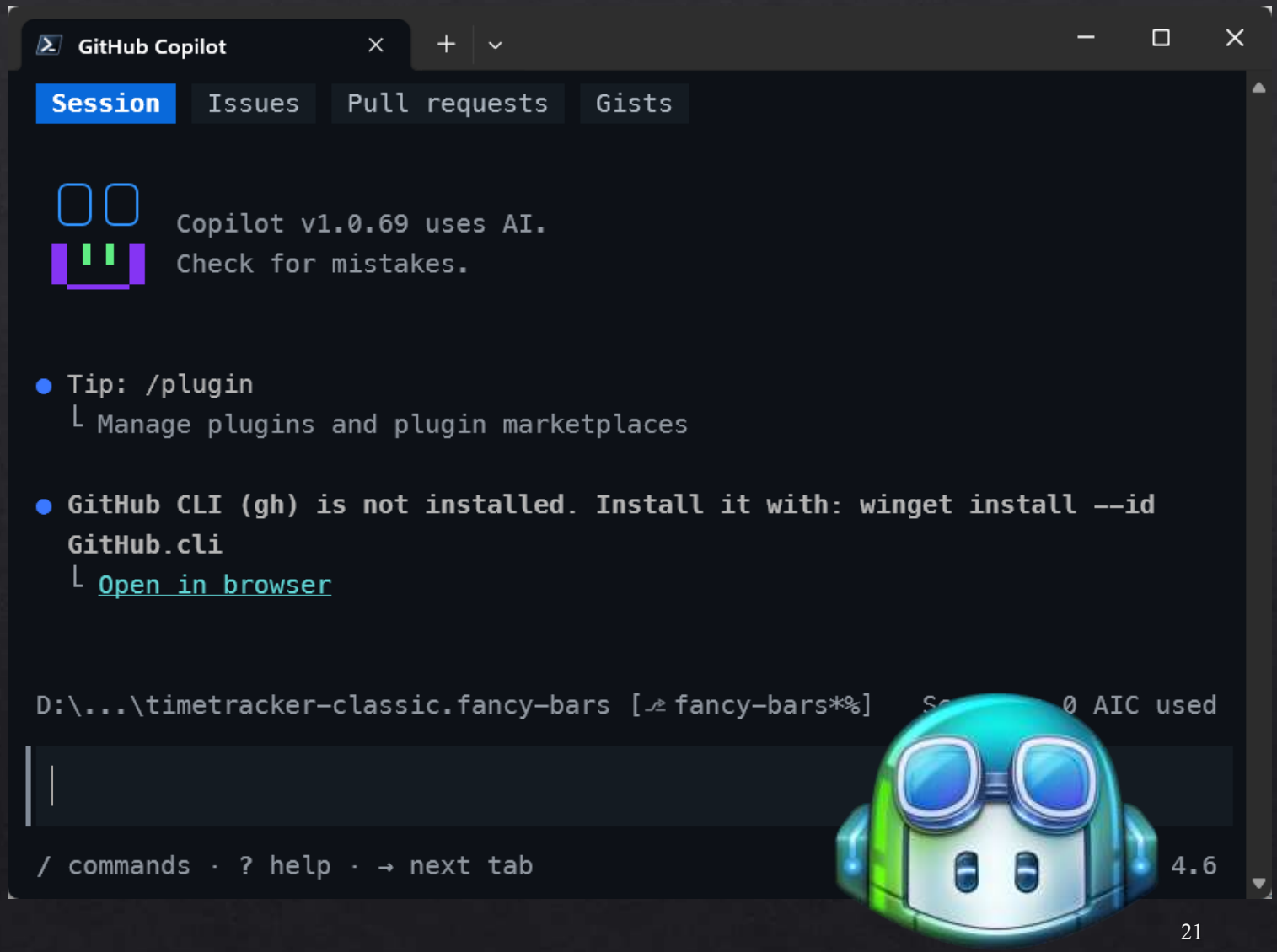
- Context Commands (@)
  - @Test, @Debug, @Optimize
- References (#)
  - #file:filename, #class:classname, #method:methodname
  - #solution
  - #Output (Tests)
- Slash Commands (/)
  - /savePrompt, /compact, /generateInstructions



Visual Studio: Copilot Additional Helps

- Quotes (“”)
  - Quotes around specific code or values.
- Explicit Constraints
  - Constraints:
    - No LINQ
    - No new packages
- Request to Ask Questions
  - Explicitly tell Copilot to ask questions when needed for clarification rather than making assumptions

Copilot CLI


GitHub Copilot
Session
Issues
Pull requests
Gists
Copilot v1.0.69 uses AI.
Check for mistakes.
Tip: /plugin
└ Manage plugins and plugin marketplaces
GitHub CLI (gh) is not installed. Install it with: winget install --id GitHub.cli
└ Open in browser
D:\...\timetracker-classic.fancy-bars [⎇ fancy-bars*%]
0 AIC used
/ commands · ? help · → next tab
4.6

Copilot CLI: Install & Setup

Install

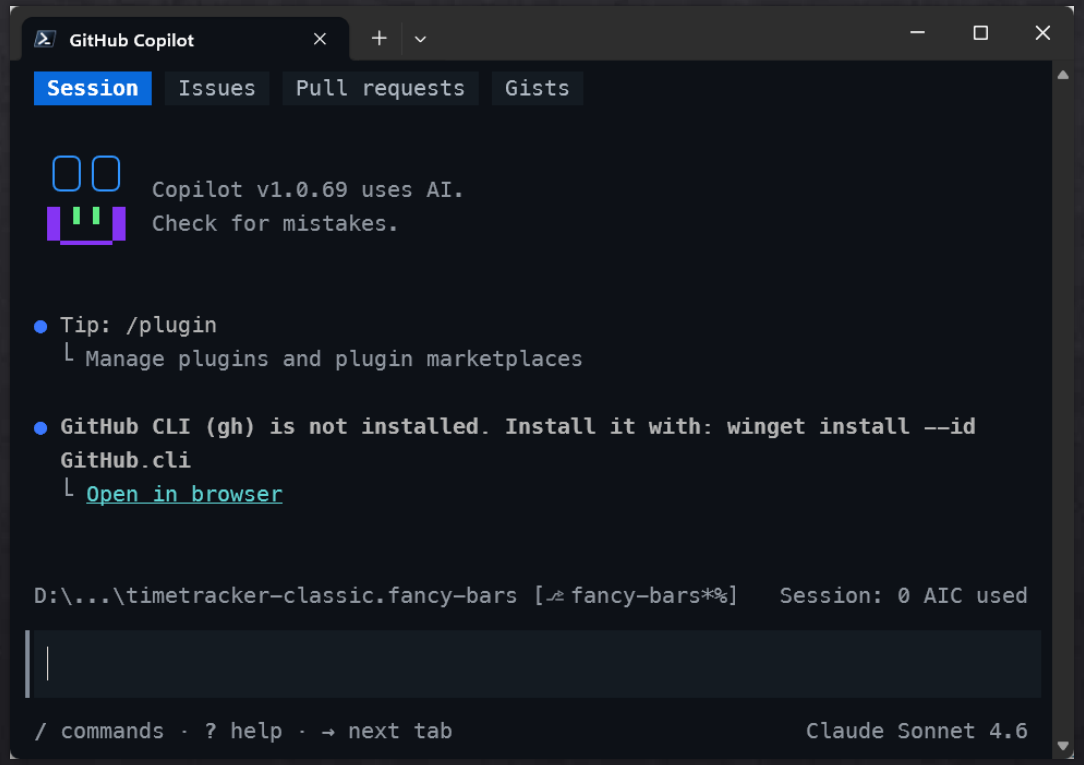
```
> winget install GitHub.Copilot
```

Run

```
> copilot
```

Connect to GitHub Account (first time)

```
> /login
```



Copilot CLI: Commands & Modes

Commands (/)

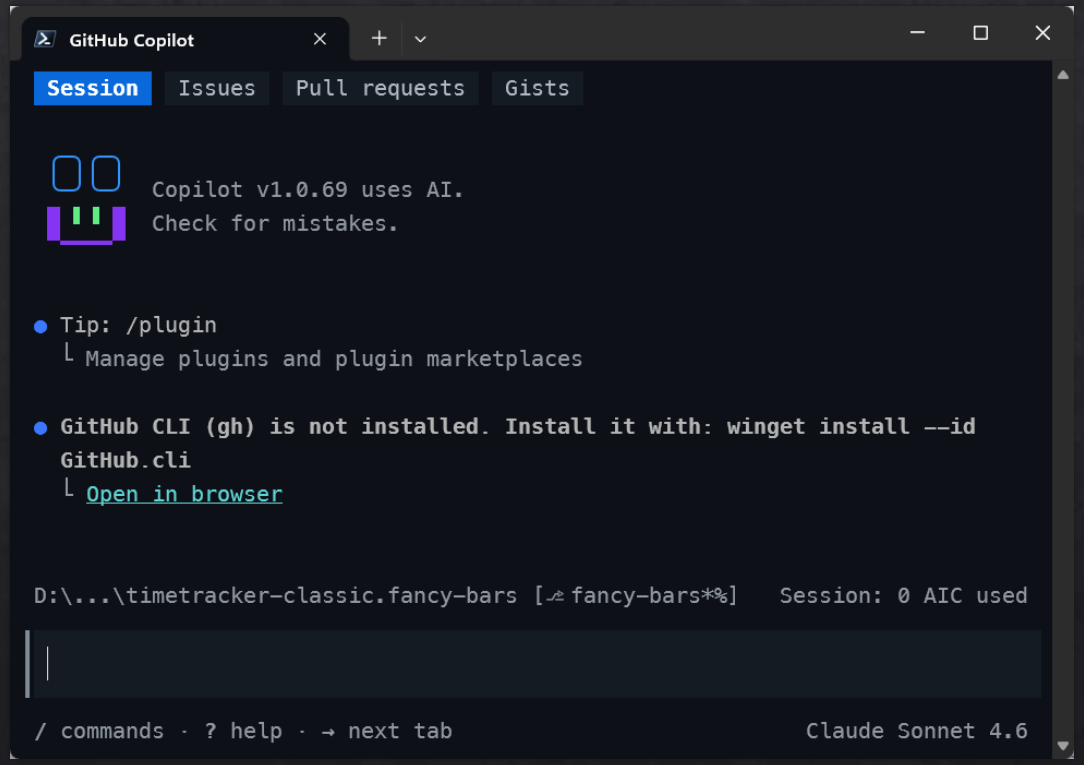
```
> /new
> /cwd [directory]
```

File Mentions (@)

```
> @Child.cs
```

Mode Changes (SHIFT+TAB)

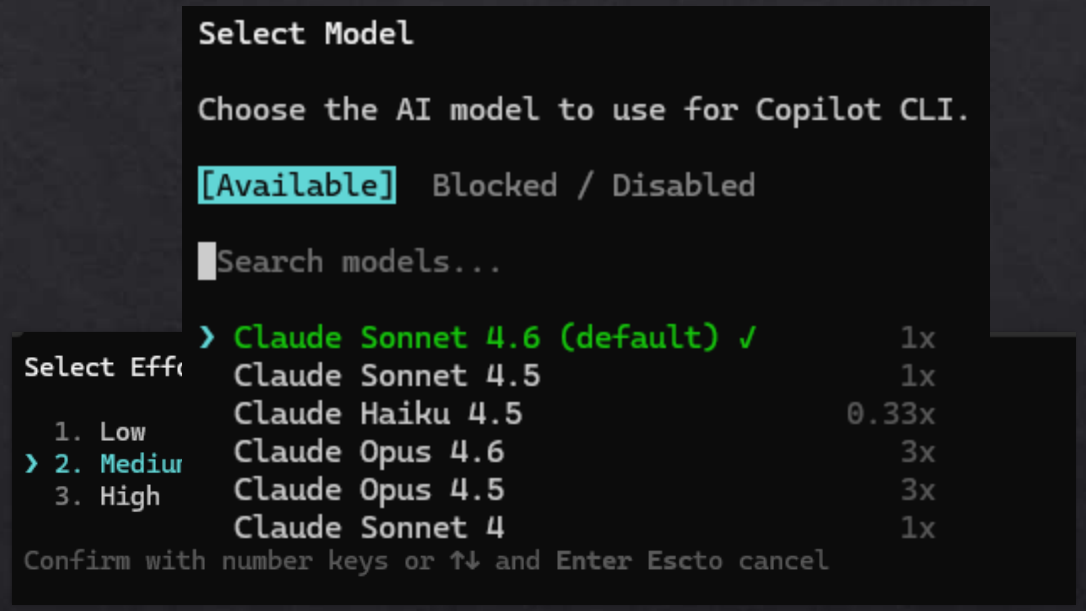
```
Normal Mode → Plan Mode → Autopilot Mode
```



Copilot CLI: Model and Effort Level

Model & Effort

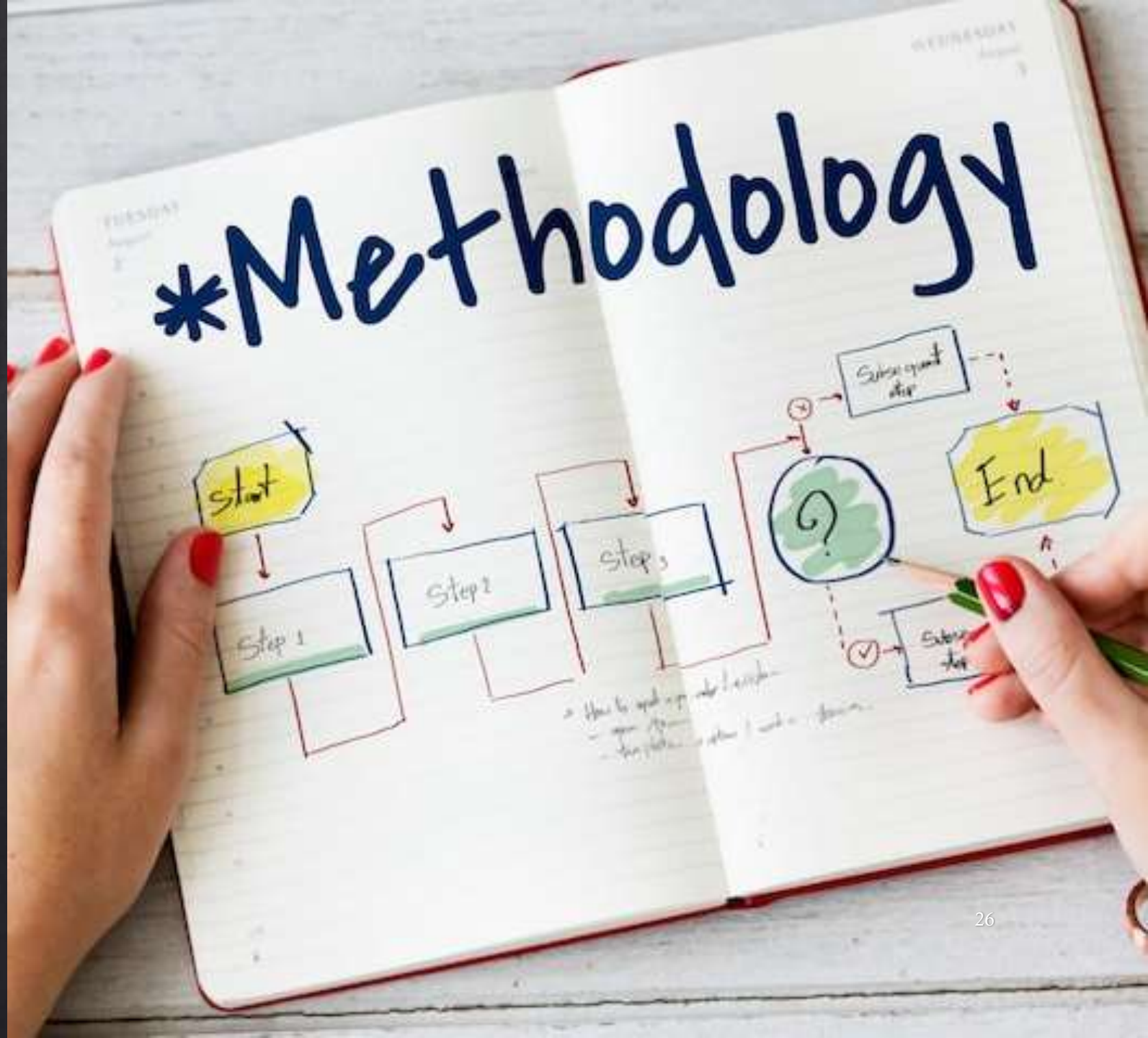
```
> /model
```



Copilot CLI: Multiple Tabs

- Multiple Tabs
  - Use Git Worktrees for Isolation
  - Difficult to keep context strait
  - Feels busy / productive
  - Some developers claim they only use CLI (no IDE)
  - Squirrel Mode

Usage Methodologies



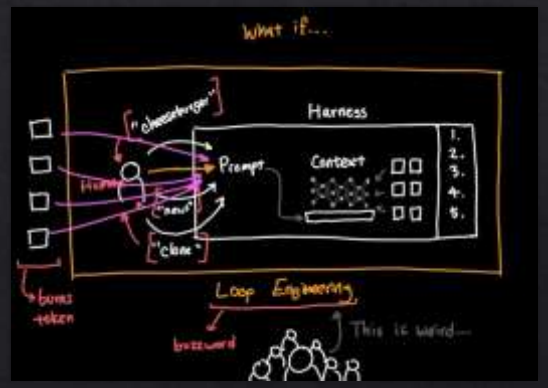
Levels of AI Usage

- AI as Knowledge Replacement
  - Google/Stack Overflow Replacement
- Basic AI-Assisted Coding
  - Inline Suggestions
  - Inline Chat
  - Simple Agent Chat
- Specification-Driven AI Coding
  - Explicit specifications with coding and architecture specifics
  - May use AI to assist in specification draft and improvements
  - 2 Step: Plan → Agent
- Intent-Driven Coding (Vibe Coding)
  - Describe high-level intent and goals
- Ralph Loops (Ralph Wiggum Loops)
  - An agentic coding pattern where an AI repeatedly selects a task, implements it, validates the results, commits the changes, and starts a fresh session for the next task.
  - Lots of tokens!
- Skynet

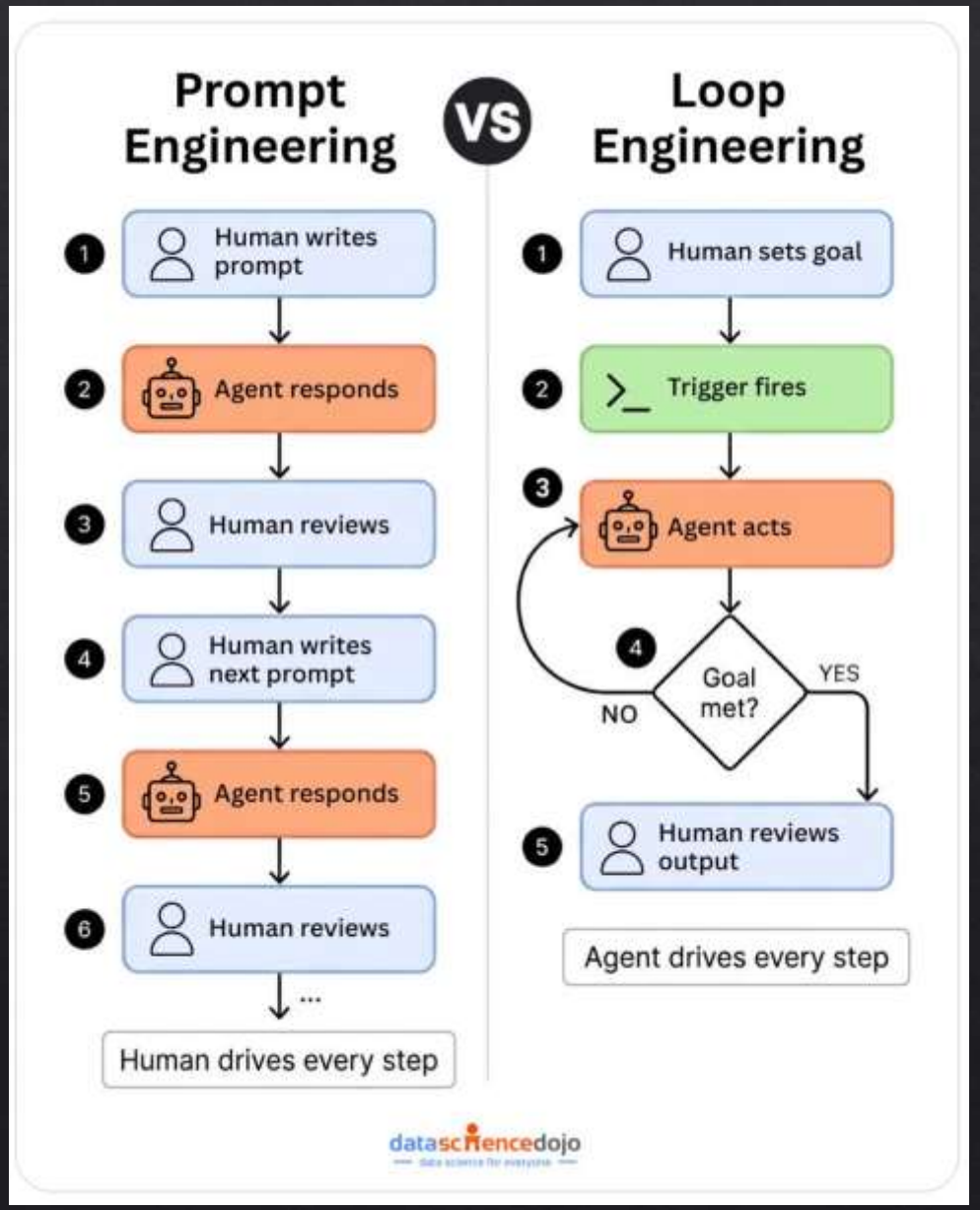

Vibe
Coding
Vibe
Coding
+ Vibe
Reviewing

AI Engineering Evolution

- Prompt Engineering
- Context Engineering
- Harness Engineering
- Loop Engineering



Source: Loop Engineering explained in 8min..



https://x.com/DataScienceDojo/status/2064739436395434269?shem=rimspwouoe,

AI Techniques

Case Study: dotnet/runtime

- 10 Months Using Copilot Coding Agent (CCA) with dotnet/runtime
  - Strengths
    - Removal and Clean-up
    - Writing Unit Tests
    - Refactoring
    - Bug Fixes
  - Weakness
    - Native Code (C++)
    - Performance Improvements
- Tips
  - Push the agent to do more: “are there additional tests that need to be created?”, “what other vulnerabilities might exist?”
  - Preparation matters more than the model
  - Update your instructions (e.g., copilot-instructions.md) when you see bad behavior

Prompt Tips

- Start General, then get specific
  - Think of it as each sentence is filtering the possible results down to what you need done.
- Give Examples
  - Example input or output helps
- Break Complex Tasks into Simpler Tasks or make a plan beforehand
- Avoid ambiguity
  - Avoid “this”, “that”, “it” and instead identify the thing you are speaking about (e.g., “the findUser function”)
- Indicate Relevant Code
  - Add a references to example files and/or functions
- Experiment and Iterate your Prompt
  - Sometimes undo changes and try again
  - Possibly start a new chat (i.e., new context)
- Keep History Relevant
  - New conversation for each new task

[Source: Prompt engineering for Copilot Chat]

Specification-Driven AI Coding

- **New Chat:** New conversation per task
- **Ask:** Investigate requirements
- **Plan:** Create a plan
- **Refine:** Adjust the Plan
- **Execute:** Agent execution
- **Validate:** Review results
- **Commit:** Commit often
- **Restart/Continue:** Continue process or start new chat

Agent Repository Isolation w/ Git

Multiple Repositories (10.8GB)

- project (2.7GB)
  - /.git/
- project-v2 (2.7GB)
  - /.git/
- project-v3 (2.7GB)
  - /.git/
- project-v4 (2.7GB)
  - /.git/

Worktrees (3.3GB)

- project (2.7GB)
  - /.git/
- project-v2 (198MB)
  - *worktree*
- project-v3 (198MB)
  - *worktree*
- project-v4 (198MB)
  - *worktree*

Agent Steering Files

Steering File	VS (Copilot)	VS Code (Copilot)	Copilot CLI	Cursor	Claude
.github/copilot-instructions.md	✅ Yes	✅ Yes	✅ Yes	❌ No	❌ No
.github/instructions/**/*.instructions.md	✅ Yes	✅ Yes	✅ Yes	❌ No	❌ No
.github/agents/*.agent.md	✅ Yes	✅ Yes	✅ Yes	❌ No	❌ No
.github/prompt/*.prompt.md	Preview	Preview	Preview	❌ No	❌ No
AGENTS.md	❌ No	✅ Yes	✅ Yes	✅ Yes	❌ No
.cursor/rules/*.mdc	❌ No	❌ No	❌ No	✅ Yes	❌ No
CLAUDE.md	❌ No	❌ No	❌ No	❌ No	✅ Yes
SKILL.md	❌ No	❌ No	❌ No	❌ No	✅ Yes

https://codersera.com/blog/agents-md-vs-claude-md-vs-cursor-rules-comparison-2026/

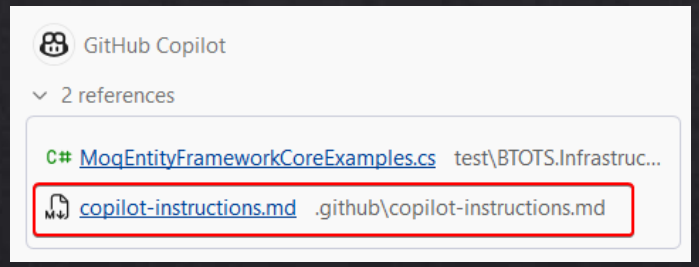
https://docs.github.com/en/copilot/reference/custom-instructions-support

https://docs.github.com/en/copilot/how-tos/copilot-on-github/customize-copilot/add-custom-instructions/add-repository-instructions

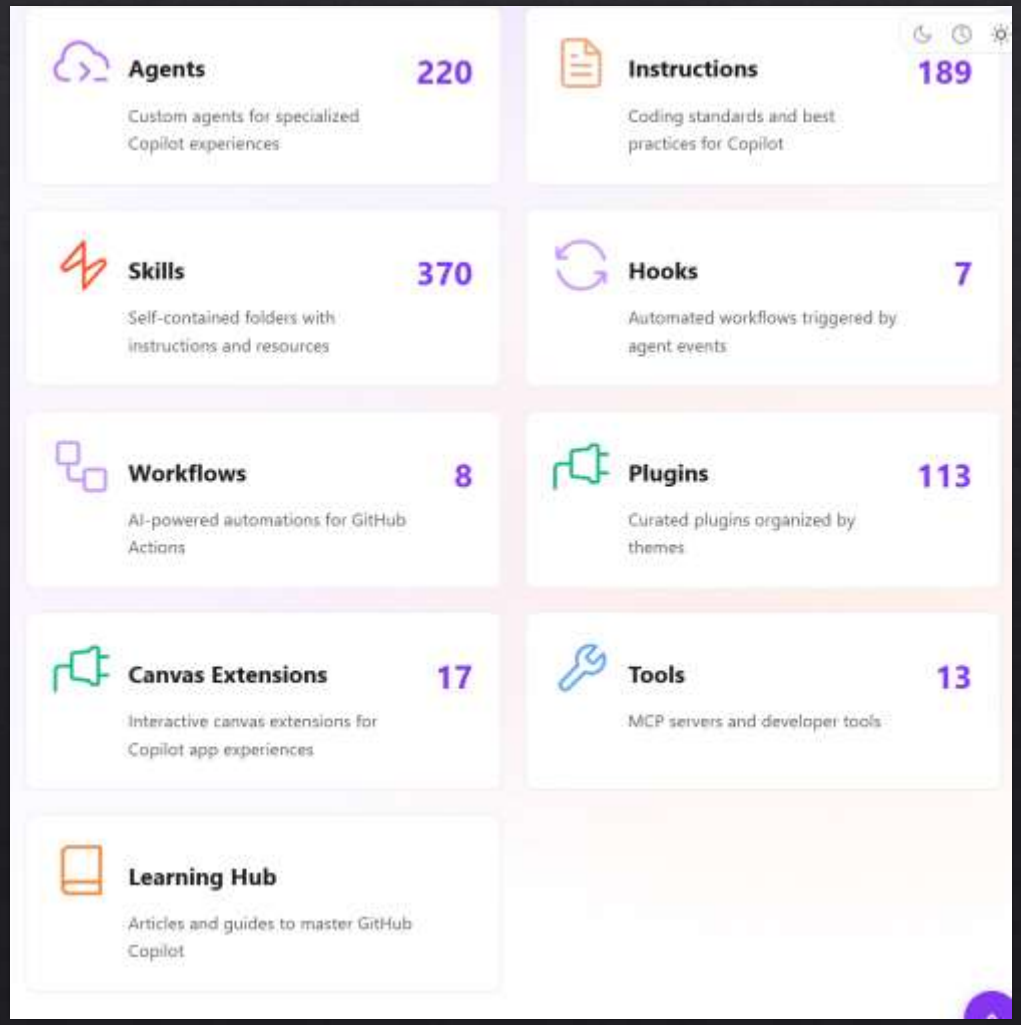
https://devblogs.microsoft.com/visualstudio/custom-agents-in-visual-studio-built-in-and-build-your-own-agents/

Helpful Tools

Agent Steering File Repository

- Awesome GitHub Copilot
  - Community-contributed agents, instructions, and skills
- Check "references" in chat to verify use
  - 

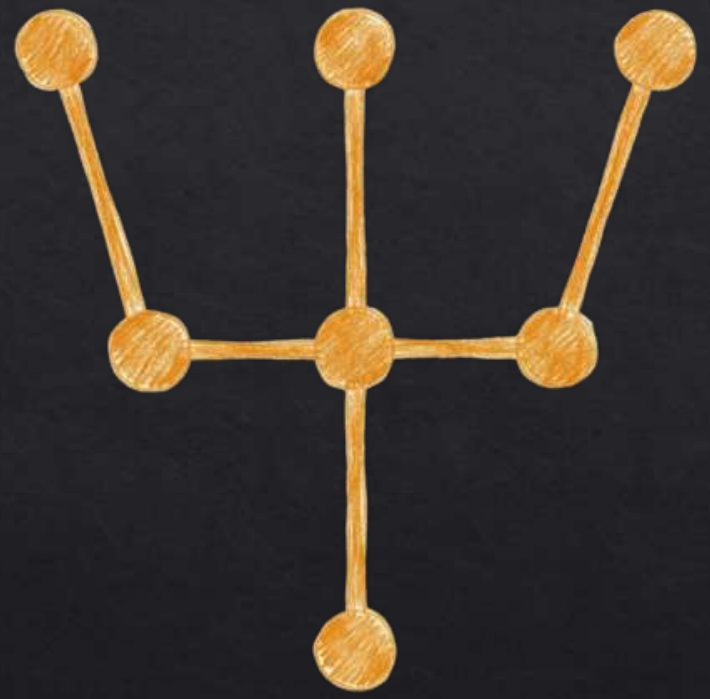
- Use Copilot to draft Agent Steering Files
  - Example Instructions



Worktrunk

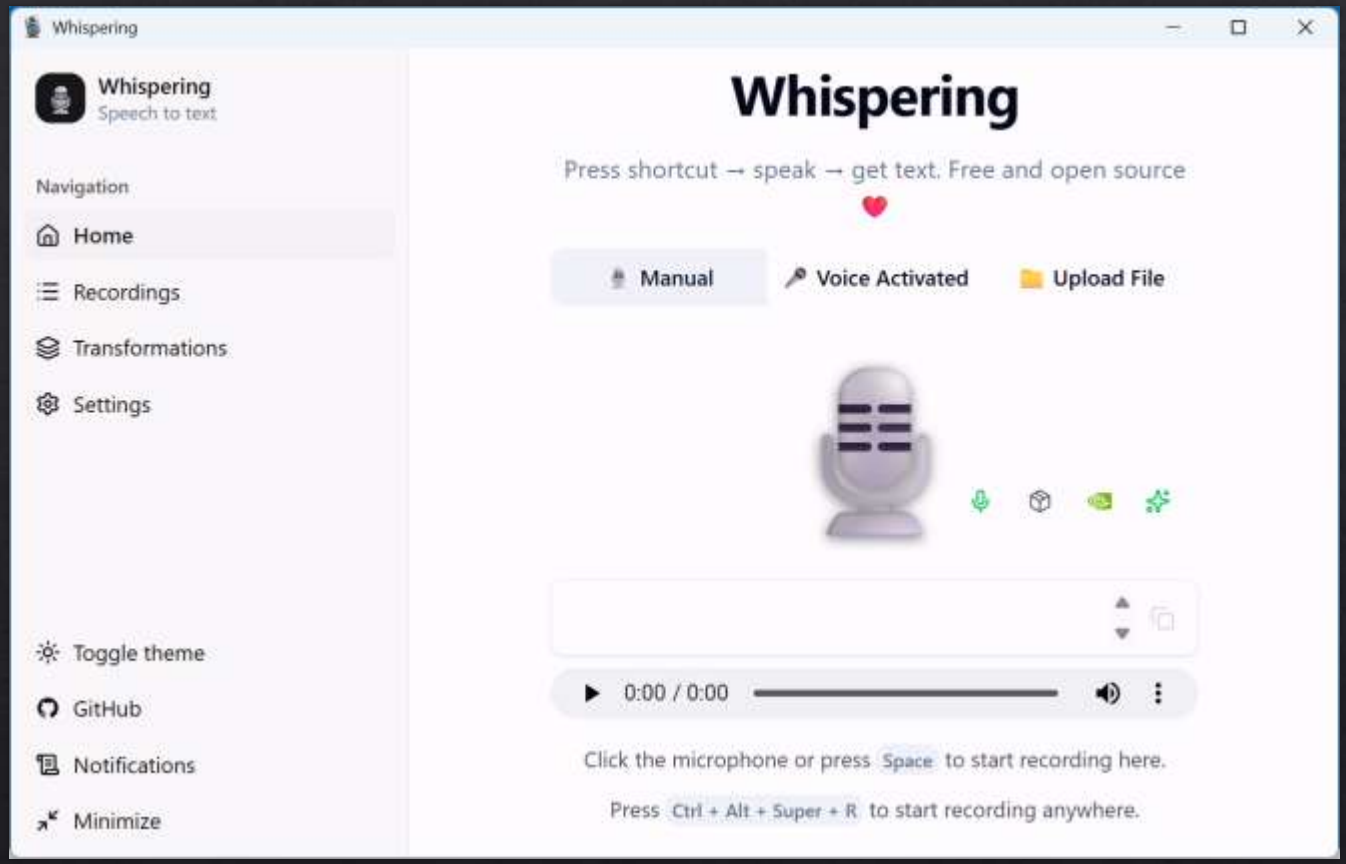
- CLI tool to assist with Worktrees
- Hooks (create, pre-merge, post-merge)
- Subcommands
  - Copy-ignored – clone .gitignore cached and config files
  - Custom Subcommands

```
> wt switch my-new-feature
> wt step copy-ignored
> wt merge
```



Epicenter Whispering

- Speak/Read Advantage
  - Speaking ~3x faster than typing
  - Reading ~2x faster than listening
- Open-source Speech to Text Transcription
  - Local AI Models
    - NVIDIA Parakeet Model
  - Cloud (API) Models
    - Use your own API key
  - Self-Hosted Transcription Server
  - Post Transformations (Regex or LLM)



LocalAI

- Free, OpenAI and Anthropic alternative.
- A small, composable AI stack.
- Drop-in replacement for OpenAI API

AI Coding Wins

AI Coding Wins

- Prototyping
- Exploring Legacy Systems
- Terraform Refactoring
- Debugging Error Messages
- Script Authoring
- Documenting Business Rules
- Converting code between languages
- Draft of Unit Tests

AI Coding Failures

AI Coding Failures

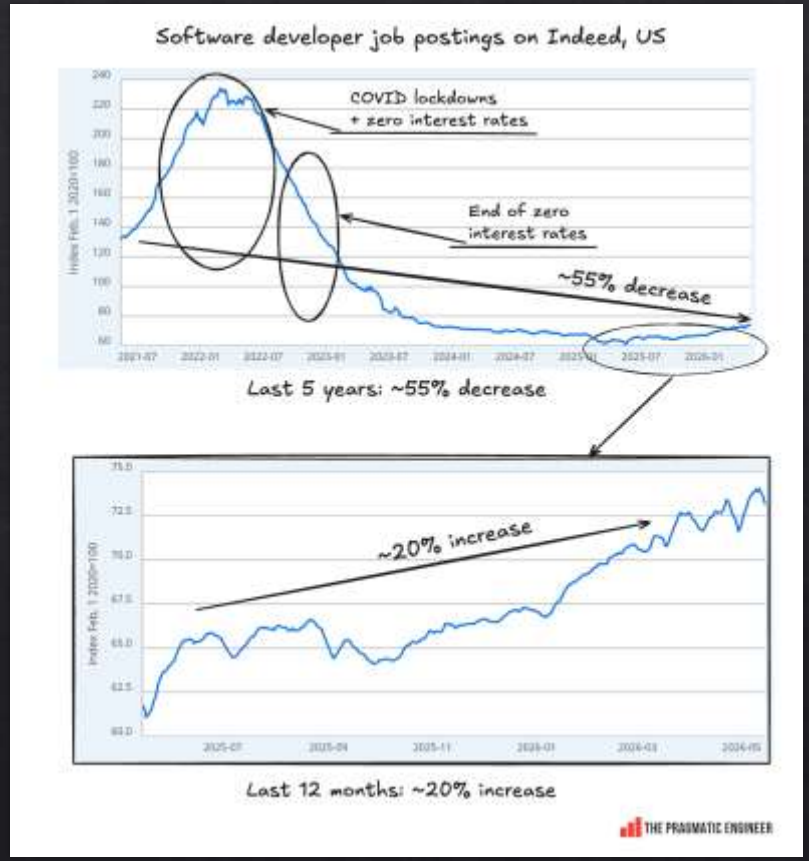
- Windows Server feature hallucination
- Code based on outdated API versions
- Just one more prompt will fix it…
- Each solution adds 5 new packages to your project
- Junior developer off the rails

AI Cost Optimization

- Use smaller models for simpler tasks
- Avoid verbose agent steering files and instructions
- Be careful with multi-agent workflows
- Compact Context: request a compaction of context
- New Chat with Summary: ask for summary and start a new chat based on the summary (e.g., handoff documents)
- Prework and exploration outside of IDE/CLI (e.g., M365 Copilot)
- GitHub Copilot CLI `/chronicle cost-tips` command

Software Engineering Outlook

- Additional productivity by Software Engineers is increasing not diminishing their demand. [TechCrunch]
- AI Washing: CEOs blaming AI for downsizing
- Boomerang Hire: 35% of new hires from previous worker pool
- Amount of internal company code is growing exponentially
- AI coding introduces 30-40% more technical debt [source]



https://newsletter.pragmaticengineer.com/p/state-of-the-job-market-2026

Explore AI Viewpoints

Proponents

- Scientific Optimists
  - Fei-Fei Li
  - Geoffrey Hinton
- Economic & Productivity Optimists
  - Tyler Cowen
  - Erik Brynjolfsson
  - Marc Andreessen

Critics

- Technical Critics
  - Gary Marcus
  - Emily Bender
- Economic and Hype-cycle Critics
  - Ed Zitron
  - Cory Doctorow