

# AI Augmented Coding with Claude

## Code

Transforming how developers build software

# The Rise of Agentic Coding Tools

2021 — *"Fancy autocomplete. I still had to drive."* GitHub Copilot: inline suggestions, you accept or reject

2023 — *"I was the Context Bridge."* ChatGPT: copy-paste code, manually shuttle context back and forth

2025 — *"The Agent drives. I navigate."* Claude Code: reads files, runs commands, executes plans autonomously

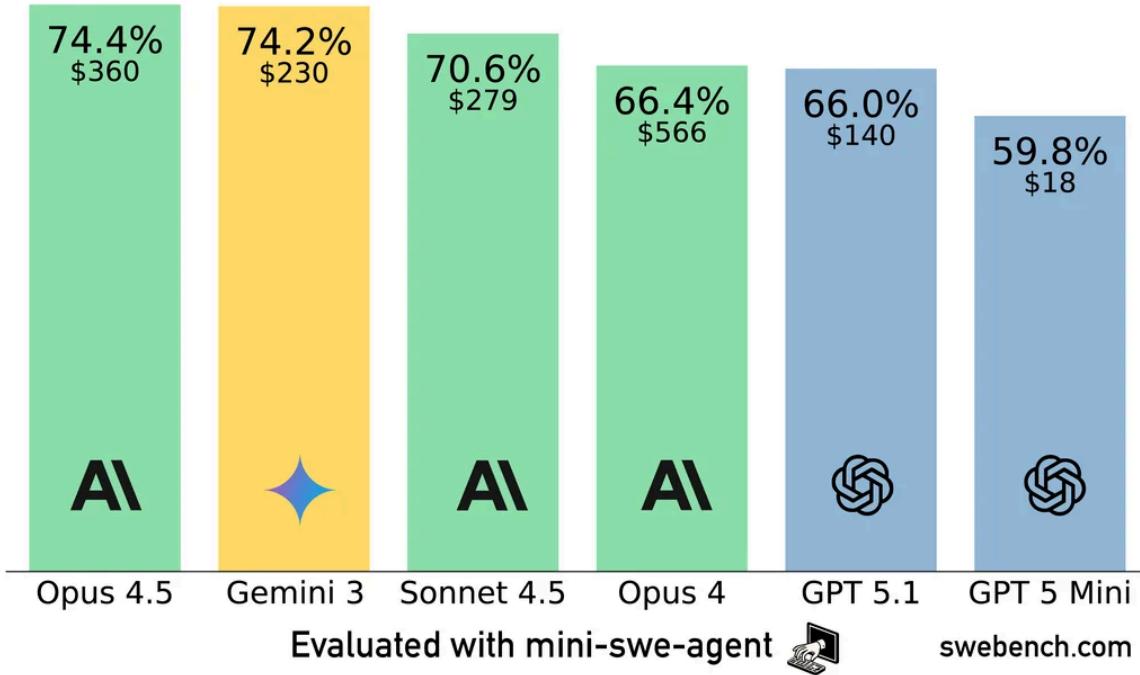
**Why go headless (CLI)?** Treat the LLM as a collaborator, not a plugin. Like messaging a senior engineer on Slack.

**Key shift:** Deep search + agentic loop — runs `grep`, `ls`, reads files, plans, executes.

# Coding Agent Benchmarks



Official SWE-bench Verified leaderboard



# Demo

Fresh Slidev repo + CLAUDE.md setup

# Planning Mode

- Claude Code has a native plan mode
- Claude explores codebase before coding
- Writes plan to file for approval
- Example: PPT notes extraction PR

# Rewinding

- Checkpoints at each change
- `/rewind` to revert
- Safe experimentation, retry with more insights

# Context Window

- Quality degrades with context size
- Progressive disclosure: load only what's needed
- Claude automatically "compacts" to preserve context

# Skills

Real-life examples:

- **MongoDB pipeline:** Complex aggregation built iteratively
- **Playwright e2e:** Flaky test diagnosed and fixed + Gherkin migration

## Feedback Loop > Background Shell

- `type:watch` in hooks config
- Tests run continuously
- Errors fed back to Claude automatically

# Feedback Loop > Hooks

- Pre/post tool execution
- Formatting on file save
- Linting before commit
- Custom validation logic

## Claude Code: Choosing the right Hook

Hook Event	Key Question	When to use this Hook
SessionStart	Do I need to load initial context or set up the environment?	Use when you need to initialize the session with project context, environment variables, or run setup scripts at startup.
UserPromptSubmit	Should I add more context or validate this prompt?	Use when you want to enhance user prompts with additional information or validate/block prompts before Claude processes them.
PermissionRequest	Should I auto-approve or deny this permission request?	Use when you want to automatically handle permission dialogs without user interaction, or log permission requests for auditing.
PreToolUse	Should I allow, modify, or block this tool execution?	Use when you need fine-grained control over tool execution, want to modify parameters, or enforce security policies before tools run.
PostToolUse	Do I need to validate results or provide feedback on what executed?	Use when you want to automatically process tool results (e.g., format code, run linters) or provide feedback to Claude about what just happened.
Notification	Should I notify the user in a specific way?	Use when you want to customize how Claude Code sends notifications (e.g., desktop alerts, sounds, logging).
Stop	Did Claude complete all tasks or should it continue working?	Use when you want to verify task completion, check for errors, or force Claude to continue working on unfinished tasks.
SubagentStop	Did the subagent finish its work or does it need to do more?	Use when you want to validate subagent completeness or ensure subagents fully complete their assigned tasks before stopping.
PreCompact	Do I need to save information before compacting context?	Use when you want to preserve important context or state before Claude compacts the conversation history.
SessionEnd	Should I clean up or save session statistics?	Use when you need to perform cleanup tasks, save analytics, generate reports, or archive conversation data at session end.

## Feedback Loop > MCPs (Model Context Protocol)

- **IDE getDiagnostics**: Type errors in real-time
- **Browser MCP**: Navigate and interact with web apps
- **GitHub MCP**: Pull PR comments directly
- **Support & Monitoring MCPs**: Notion, Sentry, ...

# Commands

- Custom slash commands
- Stored in `.claude/commands/`
- Reusable prompts and workflows
- Team-shareable

# Security Model

- Explicit allowlist / blocklist for commands (incl bash), tools, MCPs
- Sandbox mode? devcontainers? remote dev machines?

# Pricing & Rate limits

## Plans

- **Pro** (\$20/mo)
- **Max 5x** (\$100/mo)
- **Max 20x** (\$200/mo)
- **Team** (\$30/user/mo): Min 5 users, Pro-level limits
- **Team Premium** (\$150/user/mo)

## Rate limits

- **5-hour rolling window:** resets after last message + 5h
- **Weekly cap:** ~150M tokens
- Opus 4.5 is **more efficient:** 48-76% fewer tokens than Sonnet for same tasks

# Interesting Resources

- <https://claudelog.com/>
- <https://www.anthropic.com/engineering/clause-code-best-practices>
- <https://minusx.ai/blog/decoding-claude-code/>

# Transformation of the SWE Role

- From writing code to orchestrating AI
- Architecture (before) and review (after) become critical
- Higher leverage, faster iteration
- New skills: prompting, verification, orchestration, technology watch, tooling, decision making, ...