

The Five-Day Agent Rebuild—and the Systems Behind It

Coding Agents Alpha Tracker

2026-09-24

The Five-Day Agent Rebuild—and the Systems Behind It

By Coding Agents Alpha Tracker • September 24, 2026

A practitioner's reported five-day rebuild puts agent-visible tests, persistent project context, and role-based orchestration at the center; Cursor is also extending coding agents into PR security review and production monitoring.

TOP SIGNAL

Overment says he scrapped a four-year-old product and rebuilt its app, website, and API in five days; he describes it as a tool people relied on daily and says it had helped generate about \$750,000 through campaigns. [1] His account puts the emphasis on the system around the agents: unified logs, an app-connected test surface, persistent project context, and role-based Pi instances steered through Grok Bot. [1]

TRY THIS

- **Build the harness around the app.** Overment unified API/Rust/Svelte logging and connected a test environment to the app's webview so agents could interact with it, take screenshots, and collect measurements. He kept a project vision (`vision.md`), style guide, board, and feature/bug specifications available to agents; Coordinators spawn Researchers to draft specs and manage Workers. He recommends narrow tasks and short agent threads. [1]
- **Use formal methods on the risky state machine, not the whole repo.** Boris Cherny's follow-up describes the loop: model a race-prone or otherwise tricky part, find counterexamples, reproduce the suspected bugs, then fix them. He explicitly says this does not amount to formal verification of the entire codebase. [2]

- **Give computer-use agents a concrete repo-to-PR endpoint.** Theo describes Opus 5.5 finding the repository on his machine, recognizing that his local clone was stale, updating it, creating a worktree, opening a PR, verifying the fixes, and merging. It's a reported successful run, not a general reliability guarantee. [3]
- **Start a codebase audit with an explicit checklist:** "I want you to do an audit around security, performance, accessibility, maintainability, scalability, architecture, documentation, testing, automation, etc." Kent C. Dodds says Opus 5.5 found a significant security issue other models missed; treat that as one practitioner's report, not a controlled comparison. [4]

WHAT SHIPPED

- **Cursor harness efficiency:** Cursor reports 7% lower token costs without reduced agent quality. Its engineering write-up says it trimmed roughly 66% of its system prompt as models improved, moved most built-in tools to on-demand loading (most were used in fewer than 20% of conversations), and A/B-tested tool configurations against token use, cost, latency, errors, and overall agent use. On GPT-5.6, explicit cache breakpoints cut cold cache misses by 20%; numbering every tenth line in file reads cut cache-read tokens by 1.6% with no quality loss. Write-up. [5]
- **Cursor Rollouts:** Tracks a change from PR opening through production. It reads the diff and drafts an editable monitoring plan before merge, then compares post-deploy signals with the baseline; depending on configuration, it can notify the author, pause a progressive rollout, or prepare a revert PR for approval. Feature-flag ramping is still listed as coming soon. Available on Teams and Enterprise, with 10 days of trial credits. Details. [6, 7]
- **Cursor Security Reviewer:** Runs on every PR, examines changes in whole-codebase context, and reports vulnerabilities with severity, an attack path, and a proposed fix. Cursor reports average review time falling from 4.8 to 3.8 minutes and comment acceptance rising from 45–50% to 60–70%. [6, 8]
- **Claude Code cloud sessions are out of research preview.** Sessions run on Anthropic-hosted infrastructure, so work continues with the laptop closed; start one with `claude --cloud` or from the app. GitHub must be connected. Existing subscribers get a one-time \$100 Pro or \$250 Max credit, claimable through Oct. 7. [9, 10, 11]
- **LangChain Managed Deep Agents can run on a schedule:** add a file under `schedules/` with a cron expression and prompt, then deploy; the agent runs itself. Docs. [12]

GO DEEPER

- **Video — DHH on coding with agents for Linux:** DHH says agents have been writing his serious production code and that implementations improve when models critique one another. Useful as a firsthand workflow report, not an independent measure of code quality. [13]



DHH: I Raised \$20M and Have No Idea How to Spend It! (13:32)

- **Repo — Overment's `limen`:** A custom Pi extension that injects the project vision and style guide and steers Coordinator, Worker, Reviewer, and Researcher instances. [1]

Editorial take: The repeatable pattern is a closed feedback loop: give agents persistent context and a testable product surface, verify changes at the PR boundary, then check production signals after deploy. [1, 6]

Sources

1. X article by @megadevhq
2. X post by @bcherny
3. Anthropic Actually Fixed Opus
4. X post by @kentcdodds
5. Improved token efficiency for longer agent runs
6. Bots for the last mile: Rollouts, Security Review
7. X post by @cursor_ai
8. X post by @cursor_ai

9. X post by @ClaudeDevs
10. X post by @ClaudeDevs
11. X post by @ClaudeDevs
12. X post by @LangChain
13. DHH: I Raised \$20M and Have No Idea How to Spend It!