

Graph Context, Video Proofs, and Repo-Scale Agent Fixes

Coding Agents Alpha Tracker

2026-07-01

Graph Context, Video Proofs, and Repo-Scale Agent Fixes

By Coding Agents Alpha Tracker • July 1, 2026

Practical workflows from Jason Zhou, Simon Willison, Mercari, LATAM, Kent C. Dodds, and Theo: graph-backed context, agent-generated demos, repo-scale patching, and the real tradeoffs showing up around Sonnet 5.

TOP SIGNAL

Today's clearest edge was **better scaffolding, not more prompt poetry**. Jason Zhou says a graph-backed map across three repos made his coding agent *a lot smarter* and cut token use by roughly 50%, while Mercari used Sourcegraph's Agentic Batch Changes to take a GitHub Actions security fix from two repos to around 80 potential repos because the agent could reason per repo instead of doing brittle find-and-replace [1, 2, 3].

Same theme showed up elsewhere: Simon Willison is turning `--help` output into agent-usable instructions, and LATAM cut roughly 15% latency/token overhead by simplifying which agent formats the final answer [4, 5].

TRY THIS

- **Give the agent a repo map before you ask for edits (Jason Zhou).**
 1. Install Codebase Memory MCP and let it auto-index on first use.
 2. Start with `get_architecture`, then use `search_graph + trace_pass` for call chains and blast radius.
 3. Add a pre-tool-use hook so plain grep results get enriched with graph relationships even when the agent forgets the MCP-specific call.

Jason says this setup cut token use about 50%; he also shows graph queries like `files calling X without tests`, and says giant codebases can index in minutes while smaller ones finish in seconds [2]. Study his setup skill: AI Builder Club skills [1].

- **Make the agent ship a demo, not just a diff (Simon Willison).** Use this prompt skeleton inside the target repo: `Review the changes on this branch → cd to ~/dev/shot-scraper and run uv run shot-scraper video --help → use that command to record a video demo of the new features against a local dev server and demo DB` [4]. Then have the agent write a `storyboard.yml`, start a local dev server, and record the feature flow with `shot-scraper video`. The timeless trick: make `--help` rich enough that it works like a bundled `SKILL.md` for the agent [4].
- **For org-wide fixes, seed the agent with one real repo pair, then fan out (Mercari on Sourcegraph).** Mercari first fixed a GitHub Actions security issue on the Help Center frontend and backend, then extended the task to around 80 potential repos. The important part is *not* scripted search/replace: let the agent reason about each repo's setup, react to CI, and push follow-up commits when needed [3]. Announcement: sourcegraph.com/agent-ic-batch-changes [3].
- **Treat out-of-scope traffic as product signal, not user failure (LATAM).** Log real conversations from day one. LATAM used LangSmith to find two wins: moving final formatting to the supervisor cut roughly 15% latency/token overhead, and digging into the 13% out-of-context bucket revealed that 95% of those chats were legitimate needs like check-in, baggage, and benefits; adding a customer-care agent dropped that bucket to 1% and improved return rate by 6 points [5].

WHAT SHIPPED

- **Claude Sonnet 5 is landing fast.** It is now in Cursor, where Cursor says it scores 57% on CursorBench versus Sonnet 4.6 at 49%; full rankings are at cursor.com/evals [6, 7]. Simon Willison's doc readout: 1M context window, 128k max output, adaptive thinking on by default, same tool surface as Sonnet 4.6, no `temperature/top_p/top_k`, and a tokenizer that produces roughly 30% more tokens than 4.6 [8].
- **Early Sonnet 5 verdict: more agentic, not automatically better.** One same-day workflow recommendation suggested making it the default in Hermes/OpenClaw and Claude Code, while reserving Opus 4.8 for ultra-complex tasks [9]. But in Theo's same-prompt game rebuild test, Opus 4.8 finished in about 26 minutes with the best result, GLM 5.2 took 35–40 minutes with no vision, and Sonnet 5 took roughly 2–2.5 hours, spawned many subagents, and produced the messiest build; his advice is

to use smarter models for top-level routing and cheaper models for smaller subtasks so orchestration does not turn into token bloat [10].

- **shot-scraper 1.10** adds a `video` command that records browser demos from `storyboard.yml` via Playwright. Simon says GPT-5.5 xhigh in Codex Desktop built the feature, docs, and demo YAML from a prompt inside a repo checkout [4]. Docs: [shot-scraper video](#). Release: [shot-scraper 1.10](#) [4].
- **Sourcegraph Agentic Batch Changes** is in public beta: agent-driven batch edits across repos, free on Sourcegraph Cloud during beta, with self-hosted support coming July 8 in Sourcegraph 7.5 [3]. Mercari used it in preview to scope and begin patching a GitHub Actions security issue across around 80 potential repos, and Canva used it to split batch changes by code ownership in a Bazel monorepo [3].
- **Kody** is Kent C. Dodds' new OSS layer for turning existing coding agents into safer, deterministic integration and automation assistants rather than replacing them [11, 12]. Repo: github.com/kentcdodds/kody [13, 14]. Kent says it pairs well with Cursor cloud agents because those agents get a full machine environment; he used Kody's Cloudflare + Kit connections to let Cursor fix missing SPF records and newsletter deliverability issues [15, 16, 17].
- **Harbor** now integrates with Deep Agents, LangSmith Sandboxes, and LangSmith Observability for real, reproducible, isolated agent runs in parallel, with a deterministic check at the end [18].
- **Claude Desktop on Linux** is now in beta for Ubuntu and Debian, adding desktop access to Claude Code, Claude Cowork, and chat on paid plans. Download: code.claude.com/docs/en/desktop-linux [19, 20].

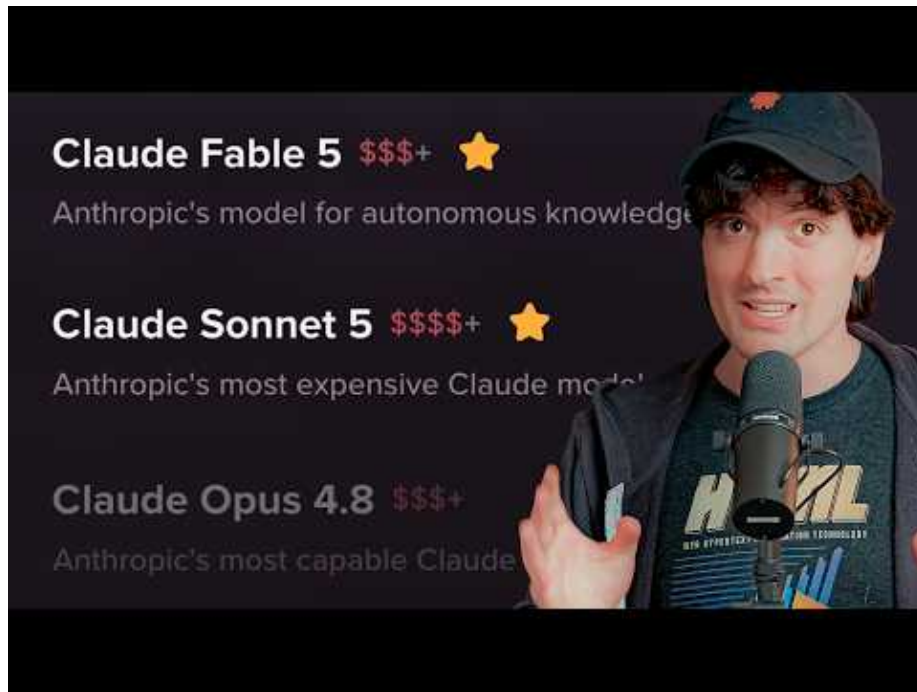
GO DEEPER

- **2:48–3:28 — Jason Zhou on the useful query surface for graph memory.** If you only watch one code-context clip today, watch this: `get_architecture`, `search_graph`, `trace_pass`, and a graph query for files calling `X` without `tests` in under a minute [2].



I was giving my coding agent context the wrong way... (3:57)

- **1:04–2:28** — Theo’s scheduled Devin regression loop. Good watch if you want agents doing daily QA, not just one-off coding: one top-level agent fans out page checks, subagents record evidence, and the whole run can be scheduled every day [10].



FABLE IS BACK! (And Sonnet 5 is here too) (1:04)

- **6:22–7:29** — **LATAM’s supervisor-only formatting fix.** Short production lesson: if every specialist formats final answers, you may be paying a hidden tax in latency and tokens [5].



How LATAM Airlines Built Intelligent Agents in Aviation | Interrupt 2026 (6:21)

- **Repos and resources worth reading end-to-end.** AI Builder Club skills for Jason Zhou’s codebase-setup pattern [1]; Kody for deterministic agent automations [13, 11]; shot-scraper video docs for agent-generated proof-of-work demos [4].

Editorial take: today’s alpha was structural — graph context, reproducible environments, and proof-of-work loops beat raw model churn.

Sources

1. X post by @jasonzhou1993
2. I was giving my coding agent context the wrong way...
3. X post by @Sourcegraph
4. Have your agent record video demos of its work with shot-scraper video
5. How LATAM Airlines Built Intelligent Agents in Aviation | Interrupt 2026
6. X post by @cursor_ai
7. X post by @cursor_ai
8. What’s new in Claude Sonnet 5
9. X post by @AlexFinn
10. FABLE IS BACK! (And Sonnet 5 is here too)
11. X post by @kentcdodds

12. X post by @kentdodds
13. X post by @kentdodds
14. X post by @kentdodds
15. X post by @kentdodds
16. X post by @kentdodds
17. X post by @kentdodds
18. X post by @LangChain
19. X post by @ClaudeDevs
20. X post by @bcherny