Coding Agents Have the Stamina—Now Ship the Control Loop

Coding Agents Alpha Tracker

2026-08-02

Coding Agents Have the Stamina—Now Ship the Control Loop

By Coding Agents Alpha Tracker • August 2, 2026

Karpathy's two-hour Opus 5 build and ChatGPT Work's cloud-computer surface point to the same constraint: agent execution is getting cheap, while verification, discoverability, and control remain the work.

TOP SIGNAL

Karpathy's Opus 5 experiment is a sharp capability/quality split: with the first paragraph of *The Lord of the Rings* and a 1M-token budget at roughly $10, the model ran for about two hours and wrote 5,500 lines of procedural three.js code; the result was "janky but fun." It still could not efficiently audit its own work because it cannot natively perceive video or play the game, so it relied on slow screenshots and made mistakes. [1]

The operating rule for coding agents is clear: spend cheap tokens on long-horizon construction, but put verification checkpoints and live intervention outside the model. Greg Brockman describes ChatGPT Work's cloud browser in exactly those terms—watch what the agent is doing and intervene in the live application when needed. [2]

TRY THIS

- **Use `/goal` + `/loop` for explicit autonomy, not as a default mode.** swyx still uses `/loop` and `/goal` when he wants the right mix of steerability and autonomy, or an open-ended "loop that generates loops" without specifying the path in advance. Start with a concrete goal, then let the loop explore; keep ordinary, tightly scoped tasks out of the loop. [3]

- **Separate work from metawork.** Keep the main chat focused on implementation and use a `/side` chat for supervision questions such as "are you stuck?" while continuing to prod the main thread. The split—"doing the work" versus "doing metawork"—keeps status checks from polluting the execution context. [4, 5]
- **Turn repeated prompts into skills.** Ben Ilegbodu's tip, relayed by Kent C. Dodds, is to notice the guidance you keep retyping and make it a reusable skill instead. That matches Dariush's year-in-review: skills became first-class across coding tools, but patient, collaborative delegation still matters more than dumping work on the model. [6, 7]
- **Treat recurring cloud work like supervised cron.** Start with the literal workflow—"ask ChatGPT Work to do any recurring task"—then monitor the cloud browser and intervene when the live application goes off course. Simon Willison reports that Work can take screenshots and deploy web apps to Cloudflare Workers, making the control loop more useful than a fire-and-forget scheduled prompt. [8, 2, 9]

WHAT SHIPPED

- **ChatGPT Work's cloud-computer surface is becoming practical, but it is hard to discover.** Simon Willison found a browser, screenshots, and Cloudflare Worker deployment ("ChatGPT Sites"); Agent Native describes Work as an agent with a computer in the cloud and a built-in tool surface. Riley Brown says the difficult part is understanding both the capability set and the difference between the app and desktop versions, while Willison calls discoverability "way too hard." [9, 10, 11, 12]
- `datasette-apps 0.2a0` adds `app_debug()` and `app_list()` for Datasette Agent workflows. `app_debug()` opens an app invisibly, executes agent-provided JavaScript in a sandboxed iframe, smoke-tests the app, and can measure element dimensions; the mechanism uses `context.browser_task()` from `datasette-agent 0.4a0`. [13] (release)
- **Metaharnesses are moving from discussion to hands-on selection.** In swyx's bake-off of flue, eve, and Matei Zaharia's Omnigent, he says Omnigent "won hands down" for his needs, with the caveat that it is container/Python-centric. The Databricks description positions Omnigent as a shared harness for coding and custom agents, with contextual security policies and spend controls; swyx has now started work on Forge agents and links a "one repository, one agent" design. [14, 15, 16, 17]
- **Kody** `v2026.08.01` adds a per-user quota meter: daily entitlement checks move to a dedicated per-user store that boots from the shared database once and stays warm. Kent C. Dodds says the change is preparation for wider release. [18, 19] (release)
- **Codeberg's anti-vibe-code policy is now a real hosting constraint.**

The *TheStandup* discussion reports a rule against projects mostly consisting of generative-AI code, including Claude and OpenAI Codex, citing unclear copyright and weak safeguards against harmful code. The panel also describes infrastructure strain, single-use repositories, and maintainer-trust erosion, but argues that "mostly" is difficult to enforce and that the more workable rule is human-reviewed, non-slop pull requests rather than banning code by provenance. [20]

- **Astra comes with unusually concrete verification artifacts.** Simon Willison reports OpenAI's claim that an internal Astra model solved ten problems with no progress on the main result for at least a decade, at less than $2,000 per problem at GPT-5.6 Sol prices. OpenAI published Lean 4 formalizations in `openai/ten-proofs`, but the number of failed attempts—and the prompts used—remain undisclosed, so this is a proof-artifact and cost signal, not a coding benchmark. [21]

GO DEEPER

- **OpenAI and Anthropic think it's time to stop — internal coding-loop segment.** The useful section relays OpenAI's reported figures: research compute devoted to internal coding inference grew 100-fold in six months, internal agentic token use grew about 22-fold, models reached 58% on an AI-research benchmark, and Codex was heavily involved. Treat these as reported lab figures, not an independent evaluation. [22]

OpenAI and Anthropic think it's time to stop (12:33)

- **The Codeberg Situation | TheStandup — policy rationale and enforcement debate.** The segment is useful for the practical distinction between banning LLM provenance and enforcing review quality: it covers the platform-cost and trust arguments, then asks who decides whether a project is "mostly" AI-generated. [20]

The Codeberg Situation | TheStandup (4:44)

- **Repo to study:** `brendanlong/cot-controllability-experiment`. Brendan Long found that louder, more detailed, repetitive prompts had essentially no effect on GPT-OSS-20B's chain-of-thought formatting: discrete prompts scored 0% in the developer and user positions, while a soft-prompt blend scored 68% and 49%. It is a useful warning not to treat hidden-reasoning formatting instructions as a reliable agent control surface; the author calls the experiment quick-and-dirty. [23]

Editorial take: The edge is no longer making an agent run longer; it is making the work inspectable—with reusable skills, separated supervision context, browser/test checkpoints, and harnesses that enforce security and spend boundaries. [6, 13, 15]

Sources

1. X post by @karpathy
2. X post by @gdb
3. X post by @swyx

4. X post by @swyx
5. X post by @agrimsingh
6. X post by @kentcdodds
7. One Year with AI For Developers
8. X post by @gdb
9. X post by @simonw
10. X post by @agentnative_
11. X post by @rileybrown
12. X post by @simonw
13. datasette-apps 0.2a0
14. X post by @swyx
15. X post by @latentspacepod
16. X post by @swyx
17. X post by @swyx
18. X post by @kodykoala
19. X post by @kentcdodds
20. The Codeberg Situation | TheStandup
21. Ten advances in mathematics and theoretical computer science
22. OpenAI and Anthropic think it's time to stop
23. I Couldn't Prompt GPT-OSS-20B to Control Its CoT