

Coding Agents Are Becoming Workflow Systems

Coding Agents Alpha Tracker

2026-08-24

Coding Agents Are Becoming Workflow Systems

By Coding Agents Alpha Tracker • August 24, 2026

Riley Brown’s mailbox-to-worker delegate, Kody’s inspectable factory loop, and Omarchy’s local-model plugin sketch the current practical frontier: explicit handoffs and controls around the model.

TOP SIGNAL

The practical unit of agentic coding is shifting from a single chat to a supervised handoff. Riley Brown’s CURSOR DELEGATE makes Grokbot the mailbox and Cursor the worker, gives delegated agents a private scratch repository, and returns the cloud agent’s full response; Kody v2026.08.23 makes its factory loop pauseable, scrollable, and inspectable with inline tool calls. [1, 2]

For practitioners, separate the control and handoff surface from execution, and make state and tool activity visible rather than burying them in one chat. [1, 2]

TRY THIS

- **Build the mailbox-to-worker bridge.** Riley Brown’s posted pattern uses literal triggers such as “Send this to Cursor,” “Ask 5.6 Sol,” and “Delegate this.” Route the request to `gpt-5.6-sol`; only set `model_params.reasoning` to `high` when the user asks for “high” or “5.6 Sol high.” Keep the delegated agent’s scratch work in a private repository and return the full Cursor response, not a paraphrase. [1]
- **Use Codex’s side chat as a second control lane.** While the main thread is running, open `/side chat` to stage the next question, test review, or follow-up instead of waiting; Derrick Choi calls the feature underrated, and Riley Brown says he uses side chat more than main chat. [3, 4]

- **When the sandbox lacks hardware, move the test—not the goal—to CI.** Simon Willison gave Claude Fable 5 this task:

```
Put https://smolmachines.com through its paces as a
fast secure sandbox. Explore what it would take to
use this to run untrusted Python and JavaScript code
in a way that is limited in what RAM and CPU time
it can take up (protection against "while true")
with no network access and filesystem access only to
designated files
```

```
Goal is to be able to use this to execute user-provided
tasks for things like data transformations
```

When the web container had no `/dev/kvm`, the agent detected the limit, ran the real test battery in a temporary GitHub Actions workflow on the branch, collected logs, and removed the workflow in the final commit. [5]

- **Clamp verbosity before changing models.** Boris Cherny says Opus is strong at long-running work and coding but has a known verbosity quirk; the immediate Claude Code workaround is `claude /config outputStyle=concise`. [6]

WHAT SHIPPED

- **Codex usage reset and efficiency fixes landed.** Tibo identified excess usage from images in long sessions with multiple compactions, high p95+ usage from Computer History, and conversation-title generation; the promised full reset covered all paid subscriptions. The follow-up says the reset propagated and fixes for those issues landed. [7, 8]
- **Kody v2026.08.23** — the homepage now plays an auto-progressing agentic conversation that can be paused, scrolled, and inspected, with tool calls opening inline. The release also makes Kody the stated system of record for memories, email, secrets, packages, and connected surfaces; adds scoped `packages.invoke` specifiers; and prevents UI deploys from resetting MCP connections or running jobs. [2]
- **OmaPilot entered alpha.** SpencerGBull released the first OmaPilot plugin alpha for OmarchyLinux and is seeking testers; DHH says the prototype runs on a local Qwen 27B model and that a version is planned for Omarchy 4.1. Treat it as a harness experiment, not a production-ready agentic OS. [9, 10]
- **The surrounding Omarchy ecosystem crossed 1,000 plugins.** DHH announced the milestone at omarchyplugins.com, a useful adoption signal for the platform in which OmaPilot is being developed. [11]
- **Client-artifact warning:** @b_nnett reports that Cursor’s Grok bot 0.18.0 shipped with runtime source maps enabled and links reconstructed

source and downloads. Audit source-map exposure and other shipped artifacts before putting a cloud-agent client in a sensitive workflow. [12]

GO DEEPER

- **Talking Postgres — 23:01: “Could I explain this to somebody else?”**. The episode’s stated production bar is whether you can explain AI-written code to someone else; its show notes frame understanding, rather than code generation, as the bottleneck. Use this as the review-standard segment. [13]
- **Talking Postgres — 09:12: red-green TDD makes agents exercise every line**. The timestamp is a compact entry point for turning an agent’s implementation loop into an executable test loop. [13]
- **Kody v2026.08.23 release notes**. Study the concrete boundary work: a visible factory loop, Kody as system of record, scoped package calls, and session isolation across UI deploys. [2]
- **smolmachines / smolvm sandbox research**. The useful pattern is the escape hatch: detect an environment constraint, move privileged testing to a suitable runner, preserve logs, and clean up the temporary automation. [5]

Editorial take: The winning loop today is not “more autonomy”; it is an explicit handoff, a recoverable environment, and an inspection surface around every consequential run. [1, 5, 2]

Sources

1. X post by @rileybrown
2. Release Kody v2026.08.23
3. X post by @derrickcchoi
4. X post by @rileybrown
5. Conceptual integrity and counting lines of code
6. X post by @bcherny
7. X post by @thsottiaux
8. X post by @thsottiaux
9. X post by @SpencerGBull
10. X post by @dhh
11. X post by @dhh
12. X post by @b_nnett
13. How AI is changing software development with Simon Willison