

OpenClaw Tightens Agent Loops as Codex, Claude, and Cursor Split in Real Workflows

Coding Agents Alpha Tracker

2026-04-12

OpenClaw Tightens Agent Loops as Codex, Claude, and Cursor Split in Real Workflows

By Coding Agents Alpha Tracker • April 12, 2026

OpenClaw's new strict-agentic and pluggable Codex harness settings were the day's clearest practical development. The bigger pattern across today's sources: coding-agent performance is increasingly determined by harness design, context handling, and clean workflow handoffs—not just model choice.

TOP SIGNAL

Peter Steinberger's OpenClaw update was the clearest practical signal today: a new `strict-agentic` execution contract for GPT-5.x that forces the agent to keep reading code, calling tools, making changes, or returning a real blocker instead of stopping at a plan [1]. He is also exposing Codex as a swappable harness plugin, which lines up with Harrison Chase's LangChain argument that context and memory behavior live in the harness more than the model [2, 3, 4].

TOOLS & MODELS

- **OpenClaw: strict-agentic mode for GPT-5.x.** Set `agents.defaults.embeddedPi.executionContract` = `"strict-agentic"` to force continued work instead of a plan-only stop. Steinberger says it is GPT-gated for now, but easy to modify on a hackable install. Docs: [providers/openai](#) [1, 3]
- **OpenClaw: native Codex harness plugin.** Enable `plugins.entries.codex.enabled` = `true`, set `agents.defaults.model` = `"codex/gpt-5.4"`, and `agents.defaults.embeddedHarness` = `{ runtime: "codex", fallback: "none" }`. In this setup, Codex owns threads, resume, compaction, and app-server execution; Steinberger says the tradeoff is weaker personality but better longer-horizon persistence. Docs: [plugins/codex-harness](#) [2, 3]

- **Cursor 3 / Composer 2.** Cursor says v3 is simpler, more powerful, and designed for a world where agents write the code. Separately, it doubled Composer 2 usage in the new Agents Window this weekend and removed hourly limits. Announcement [5, 6]
- **Codex 5.4 vs Claude in real work.** Riley Brown says Codex x high built a Swift iOS app he described as basically a Replit clone with sandboxes, database, and live edit/preview in one shot in 40 minutes after an hour of prompt work. Then a minor UI change took 3 hours and he switched to Claude [7].
- **Closed harnesses are starting to look like memory lock-in.** LangChain argues memory is a harness responsibility, not a plugin, and warns that closed/stateful systems—including Claude Managed Agents and Codex encrypted compaction summaries—can trap state inside a provider. Its recommended counterexample is the open-source, model-agnostic Deep Agents harness [4].

WORKFLOWS & TRICKS

- **Kill lazy-agent stops with stricter execution contracts.** In OpenClaw, `strict-agentic` explicitly pushes GPT-5.x to keep reading code, calling tools, making edits, or returning a concrete blocker instead of ending with a plan [1].
- **Treat harness choice as a behavior knob.** OpenClaw’s plugin architecture means you can swap Pi for Codex or another custom harness. Steinberger’s practical takeaway: Codex may weaken personality, but it can improve longer-horizon persistence [3].
- **Replicable end-to-end shipping loop from Kent C. Dodds.** 1) Ask Kody to build the full app in Claude desktop. 2) Let it deploy to Cloudflare Workers and generate the OG image in the same conversation. 3) If the prototype deserves to live, have Kody create a GitHub repo with CI/tests. 4) Hand follow-up tweaks to Cursor Cloud Workers. Live example: `shape-color-match.kentcdodds.workers.dev` [8, 9, 10]
- **Bake observability into agent-built apps early.** Kent says he configured Sentry integration with Kody so new Kody-built apps can ship with Sentry error reporting without a manual Sentry login step [11].
- **Claude Code works for lightweight, mobile-first repo tasks too.** Simon Willison used Claude Code on his phone to compile SQLite’s Query Results Formatter to WebAssembly and build a playground UI. Playground: `tools.simonwillison.net/sqlite-qrf` [12]
- **Audit context handling explicitly.** Theo says Claude Code’s inject dynamic context pattern is useful enough to belong in a broader skills standard across Codex CLI, Pi, and Cursor. LangChain’s matching audit frame is the right checklist: how do `AGENTS.md` or `CLAUDE.md` get loaded, what survives compaction, how is skill metadata shown, and how much filesystem context is exposed? [13, 4]

PEOPLE TO WATCH

- **Peter Steinberger** — High signal because he is posting concrete OpenClaw configs and runtime tradeoffs, not abstract agent talk [1, 2, 3].
- **Kent C. Dodds** — Worth following for public, end-to-end workflows: build, deploy, add CI/tests, then productionize with integrations like Sentry [8, 10, 11].
- **Riley Brown** — Useful because he shares the annoying edge cases too: Codex 5.4 crushed a large one-shot app build, then stumbled on a trivial UI tweak [7].
- **Theo** — Still a strong barometer for fast-moving coding-agent behavior: he is surfacing both durable context tricks and blunt regression/model-comparison feedback from live use [13, 14, 15].
- **Simon Willison** — Consistently good for small, real repo-level examples that show what coding agents can do in day-to-day engineering, not just staged demos [12].

WATCH & LISTEN

- **Kent's full Kody build session** — X post. *Timestamp not specified in source.* Shows the full Claude desktop conversation that built the game, deployed it to Cloudflare Workers, and generated the OG image [8, 9].
- **Theo's Claude Code regression demo** — X post. *Timestamp not specified in source.* Short video attached to his claim that Claude Code had regressed to the point of being basically unusable for him [14, 16].
- **Cursor 3 launch video** — X post. *Timestamp not specified in source.* Quick product read on Cursor's agent-first IDE direction [5, 6].

PROJECTS & REPOS

- **Deep Agents** — LangChain's open harness pick: open source, model agnostic, based on open standards like `agents.md` and `skills`, with memory plugins for Mongo, Postgres, and Redis plus self-hosted deployment options [4].
- **OpenClaw / Pi harness stack** — Worth tracking because LangChain lists Pi and OpenClaw among notable harness examples, while Steinberger is turning OpenClaw harnesses into swappable plugins [4, 3].
- **Simon Willison's SQLite QRF playground PR** — A concrete repo-level example of an agent turning an upstream library into a runnable WebAssembly playground fast. Related PR: github.com/simonw/tools/pull/266 [12].

Editorial take: the biggest leverage point today was not raw model IQ; it was the harness knobs that control persistence, context, and clean handoffs between tools. [1, 3, 4]

Sources

1. X post by @steipete
2. X post by @steipete
3. X post by @steipete
4. Your harness, your memory
5. X post by @cursor_ai
6. X post by @cursor_ai
7. X post by @rileybrown
8. X post by @kentcdodds
9. X post by @kentcdodds
10. X post by @kentcdodds
11. X post by @kentcdodds
12. SQLite 3.53.0
13. X post by @theo
14. X post by @theo
15. X post by @theo
16. X post by @theo