

Models are skipping the edit tool: Ronacher and Huntley on code mode, stale AGENTS.md files and harness design

Coding Agents Alpha Tracker

2026-10-17

Models are skipping the edit tool: Ronacher and Huntley on code mode, stale AGENTS.md files and harness design

By Coding Agents Alpha Tracker • October 17, 2026

State of Agentic Coding #11 reports that frontier models now write their own code to edit files instead of using harness tools, and asks what that means for harnesses, visibility and agent instructions. Also: bounded overnight loops, continuous review, and new Cursor and Codex releases.

Models are routing around the edit tool

In State of Agentic Coding #11, **Armin Ronacher** reported that Pi still provides an edit tool, but with Astra and Opus 5.5 it goes unused in about 80% of cases. The models write one-off Python to edit files instead [1]. His explanation is that training rewards success by whatever means, so the models fall back on ad hoc code. Pi already offers “code mode”, which runs JavaScript in the harness loop rather than bash where your files live, and will probably make it the default [1].



State of Agentic Coding #11 with Geoff, Armin, and Ben (60:04)

Three consequences for developers follow:

- **Readability.** Astra’s code written on the fly reads like code golf: single-character variables and no indentation [1]. Ben Tossell pointed out the tension: very compact code suits a tool loop but isn’t what you want in application code [1]. Watch for it leaking into your codebase.
- **Visibility.** Pi will probably hide tool output soon, because when the model runs unreadable code, explaining it would take a second LLM [1]. Ronacher calls this a regression. For critical work you need checks along the way, not just the agent’s word at the end, and he floats a separate commentary stream from another model [1].
- **Model divergence.** Huntley says Kimi still reaches for the edit-file tool, while the frontier American models write code over bash to get around it [2].

Huntley’s write-up adds a hot take you can act on: delete and redo your AGENTS.md and many skills after every model release, or at least every couple of months, because “best practice” goes stale [2]. The same reasoning applies to harnesses. Most agents copied Claude Code’s loop: transcript, tool call, tool result. That loop fits background compaction and multi-person Slack threads poorly, and an agent will “happily build you a decent-looking agent” on top of it that is unreliable underneath [2]. Pi Durable, which resumes exactly after a harness restart, including with sub-agents talking to each other, took four

months for about 10,000 lines. The team solved the problem around twenty times, about two hours each, before the trade-offs felt right [2].

Two other ideas from the episode:

- **Huntley’s Lisp demo** connects an LLM to a running Lisp image with just two tools, evaluate an expression or define a function, and lets you roll changes back. You change the running program by chatting with it instead of editing files and recompiling [2].
- **Ronacher uses fast classifier models** each morning to check open issues against his own criteria and see how they changed since the day before [1].

Overnight and parallel patterns from High Taste Testers

On Latent Space’s new High Taste Testers show, one practitioner said Opus 5.5 is finally good enough to start agents at 10 p.m. and review at 9 a.m. across about 10 prototypes. He also warned that agents running non-stop on production software create more code than you can absorb [3]. His method is a “poor man’s Ralph loop”: queue 5–10 prompts, such as implement, compact, reread the plan, keep improving, so the run stays bounded [3].

Other patterns from the episode:

- A “detach mode” skill runs CLI agents in the background, up to 40 in parallel, with logs to monitor, pause, improve and rerun [3].
- Codex staff reportedly said subagents beat separate sessions because of cache optimizations [3]. One user set 48 subagents in their config and burned through their Astra limits in about two hours; another keeps it at about 15 [3].
- For very hard problems, run the main thread in ultra and give it only one job, dispatching subagents. Put timers on them and push any that stall [3].

OpenAI’s Tibo says Codex & Work now gives more effective usage by reusing context across subagents, keeping context cached longer, and routing requests to get more cache hits [4]. ChatGPT’s GPT-6 Astra Pro can now run several agents in parallel on parts of one task, for Pro users in the Chat tab [5].

Review and codebase hygiene

The Claude Code team (Addy Osmani, Morgan Westlee Lunt and Qing Wang) says “did a human read the code?” is the wrong question. The real one is what could go wrong with a change and which check, automated or human, should catch it. Review now runs continuously “like spellcheck”, and Claude has reviewed and fixed most changes at least once before a human looks [6]. Boris Cherny says Claude Code Review runs on all his PRs and catches almost every bug before it goes in [7].

Geoffrey Huntley points out that agents copy a messy codebase’s bad patterns everywhere. He suggests periodically having the model rename things and restructure modules, files and directories. He calls this a “janitorial Roomba” job, suited to a skill run by cron or by a GitHub Actions job on trunk [8]. His personal-bot write-up has one design rule: build thin bots, one per function, that call each other and chain together like Unix pipes [9]. His Overalls bot sends coding tasks to short-lived Firecracker VMs on his own infrastructure [9]. He also flags Claude Code’s `crossSessionInbound=accept` setting as apparently the key primitive behind “hive swarms” [10].

Tool releases

- **Cursor:** cloud agents can build interactive canvases, such as dashboards or reports, that you share by link [11]. The Environments API (beta) lets you create and manage environments programmatically, including the repos they clone and their `environment.json` setup [12]. Environments now support up to 500 repos, and large multi-repo environments start 5x faster (changelog) [13].
- **LangSmith AI Search:** structured filters narrow the traces first, then a small model checks each remaining one in parallel, usually in under half a second each [14]. LangChain checked 150 conversations from its Open SWE coding agent. The median ran 84K tokens, about 1.5K came from the user and the final reply, and roughly three-quarters was tool output [15].
- **Managed Deep Agents** adds your credentials at a proxy outside the sandbox, and only on calls to services you’ve allowed, so a malicious package can’t steal them [16].
- **Riley Brown’s** hands-free setup: walk with AirPods and talk to his Dot coordinator, which starts several Codex sessions, such as company research, a landing page and a website update. He says to connect your tools so it has context [17]. He also calls GrokBot through a webhook skill from Claude Code to search X without an X API key, and says setup takes under two minutes [17].
- **Fai** (repo) is Krzysztof Cieslak’s functional language for exploring software engineering tools designed for agents [18].
- Theo says OpenAI has lately given him significantly more cybersecurity risk flags and refusals than Anthropic [19].

Sources

1. State of Agentic Coding #11 with Geoff, Armin, and Ben
2. state of agentic coding #11: the curiosity test, lisp, and labs that can’t fail
3. High Taste Testers: OpenAI Shipped Everything. What Actually Works?
4. X post by @thsottiaux

5. X post by @OpenAI
6. Quoting Addy Osmani, Morgan Westlee Lunt, and Qing Wang
7. X post by @bcherny
8. X post by @GeoffreyHuntley
9. revisiting some of my priors about openclaw
10. X post by @GeoffreyHuntley
11. X post by @cursor_ai
12. X post by @cursor_ai
13. X post by @cursor_ai
14. X post by @LangChain
15. X post by @LangChain
16. X post by @LangChain
17. Gemini 4 and Muse Wonder Dropped... But Fable 5.5 Is TERRIFYING
18. X post by @k_cieslak
19. X post by @theo