Harness Discipline, Grok 4.5's Arrival, and Claude Code's /checkup

Coding Agents Alpha Tracker

2026-07-09

Harness Discipline, Grok 4.5's Arrival, and Claude Code's /checkup

By Coding Agents Alpha Tracker • July 9, 2026

The sharpest signal today is that coding-agent leverage is coming from better harnesses and leaner context, not just new model launches. This brief covers Grok 4.5's early real-world use, Claude Code's new cleanup command, and concrete loops you can copy today.

TOP SIGNAL

Harness quality is now a real performance lever, not an implementation detail. Matei Zaharia says the simple Pi harness matched vendor harness success rates with Opus and GPT 5.5 at roughly half the cost because it sent smaller inputs [1]. The same theme showed up across tools: Claude Code shipped `/checkup` for pruning unused skills/MCPs/plugins and breaking up bloated `CLAUDE.md` files [2], Ben Tossell got Claude Code's starting prompt down to 13K tokens by inspecting the payload and stripping cruft [3], and Jason Zhou reports Pi extensions that cut tool tokens by 80–96% [4].

TRY THIS

- **Audit a live launch PR against its own report** *(Theo on Lakebed with Grok 4.5 in Cursor).* Paste the current PR plus the audit/report and ask `Have we resolved them? Are there other things worth solving before launch?` [5]. When multiple numbered lists exist, prefix which list you're referring to; Theo says Grok 4.5 handled duplicate issue numbers cleanly in that setup [5]. Then tell it which findings to ignore and which deserve separate PRs; in his run it opened two PRs, answered follow-up questions, and produced a launch to-do list in one pass [5]. After review comments land, ask it to address both PRs and let

Cursor's `babysit` skill monitor follow-ups; Theo also got visual fixes by pasting a screenshot of the comments [5].

- **Use failing tests as the task contract** *(LangChain/NemoClaw + Decode).* Create the smallest repro you can with one missing function and a failing unit test [6]. Start `decode` and paste `inspect this tiny Python project, fix the failing test by making the smallest reasonable change, run the test and summarize what changed` [6]. Approve only the expected actions—here it asked to create the function and run the tests [6]—then rerun `python3 -m unittest` yourself before trusting the diff [6]. The pattern is durable: inspect workspace, make the minimal change, validate, summarize [6].
- **Run `/checkup` before blaming Claude Code for being slow or forgetful** *(Boris Cherny).* It can clean unused skills/MCPs/plugins, dedup local vs checked-in `CLAUDE.md`, break a root `CLAUDE.md` into nested files + skills, turn off slow hooks, update Claude Code, enable auto mode, and pre-approve frequently denied read-only commands [2]. It confirms before making changes [2].
- **Feed reference implementations, not style adjectives** *(Kent C. Dodds + Romain Huet).* If you already have a canonical pattern, tell the agent `Just do it like I did it in the Epic Stack` instead of re-describing the architecture from scratch [7]. Huet's Codex advice is the same at a higher level: talk to the model like a smart coworker and spend effort on giving full context, not on clever prompt magic [8].

WHAT SHIPPED

- **Grok 4.5 in Cursor.** Cursor says it partnered with SpaceXAI to train the model; it's live now with double usage for the first week, and `Composer 2.5` remains a separate weight class with more models of that size coming [9, 10, 11]. Theo's firsthand use on Lakebed and a 3D game found strong multi-step orchestration/context handling and unusually good 3D spatial reasoning, but weaker sub-agent delegation than Fable 5 or GPT-5.6 [5]. Sualeh Asif says it's the first model he personally prefers over Opus for single-agent iteration, and Jediah Katz says Ramp's internal evals found frontier-level shell discipline [12, 13]. Theo cites pricing at `$2/M` input and `$6/M` output plus benchmark references close to GPT-5.5/Fable, with one code-benchmark view showing 2M tokens for Grok 4.5 versus 3.5M for GPT 5.5 on medium [5]. Read: Cursor blog [10].
- **Claude Code `/checkup`.** New command for cleaning unused skills/MCPs/plugins, deduping and splitting `CLAUDE.md`, turning off slow hooks, updating the client, enabling auto mode, and pre-approving common read-only commands—all with confirmation before edits [2].
- **NemoClaw Deep Agents Blueprint.** LangChain and NVIDIA intro-

duced an open reference stack with Nemotron 3 Ultra, a Deep Agents harness layer for planning/tool use/memory/long-running tasks, and an inspectable OpenShell runtime; LangChain's pitch is enterprise ownership/customization, benchmark-leading performance, and over `10x` lower inference costs [14, 15]. Related demos showed `Decode`, the open-source model-agnostic terminal agent with skills, subagents, MCP, goal mode, and LangSmith traces [16]. Read: LangChain blog [14].

- **Antigravity CLI `1.1.0`.** New interactive execution modes, improved UI, and workspace fixes; `shift+tab` cycles modes and `/changelog` shows updates inside the terminal [17, 18, 19]. Changelog: GitHub releases [20].
- **Pi Agent SDK public builder path.** Jason Zhou says he rebuilt Posia with a core agent of roughly 15 lines, recorded a 17-minute walkthrough on extensions/hosted agents/core SDK, and highlighted extensions that cut tool tokens by `80-96%` [4]. Setup repo: AI-Builder-Club/skills [4].
- **Task routing is getting more specific.** Tim Neutkens says GPT-5.6-Sol has spent 2+ months handling day-to-day Next.js work with short prompts, architecture-aware bug fixing, and end-to-end server refactors tied into failing tests and deployment checks, with some PRs queued until after Next.js `16.3` [21]. Peter Gostev's current split: Fable for architectural discussion/UI/writing, GPT-5.6-Sol for robustness, adherence to existing code patterns, sub-agent handling, multi-day `/goal` runs, and token efficiency [22]. Riley Brown adds that GPT-5.6-Sol passed the Replit benchmark in one prompt [23], while Kent C. Dodds says Fable + Composer `2.5` subagents + Kody + Cursor migrated his site to Cloudflare in one PR [24].

GO DEEPER

- **16:10-18:39 — Theo's Lakebed audit loop.** Best practical Grok 4.5 clip today: PR + report in, launch-gate analysis out, then separate PRs for the fixes that actually matter [5].

Oh no (the new Grok model is good) (16:09)

- **3:11-5:26 — Decode goal mode + long-running build.** Good example of turning a loose prompt into a session contract: declare the goal, let the agent draft acceptance criteria, then inspect the generated app at `localhost:8000` [16].

How to use dcode + Nemotron 3 Ultra (3:11)

- **Repo to study — AI-Builder-Club/skills.** Jason Zhou published his Pi extension setup here after rebuilding Posia with a tiny core agent; worth reading if you want to shrink tool chatter instead of just model-hop [4].
- **Writeup to study — How to kill the bloat in Claude Code's system prompt.** The useful habit here is to inspect the real payload first, then tune prompt overhead like infrastructure [3].
- **Tiny HITL tool — nameplate.sh.** Peter Steinberger built it so agents can show humans a big contextual alert when input is needed instead of tossing no-context dialogs; nice pattern for multi-machine or screen-share-heavy workflows [25, 26, 27].

Editorial take: today's real edge was not a flashy benchmark—it's tighter harnesses, smaller inputs, and agent loops that can prove work through tests, PRs, and auditable traces. [1, 6, 16, 3]

Sources

1. X post by @matei_zaharia
2. X post by @bcherny
3. X post by @mattpocockuk

4. X post by @jasonzhou1993
5. Oh no (the new Grok model is good)
6. NemoClaw + dcode: A governed blueprint for AI coding agents
7. X post by @kentcdodds
8. Inside OpenAI: AGI, Codex & Prompting | Tech Unlocked
9. X post by @cursor_ai
10. X post by @cursor_ai
11. X post by @cursor_ai
12. X post by @sualehasif996
13. X post by @RampLabs
14. X post by @LangChain
15. X post by @LangChain
16. How to use dcode + Nemotron 3 Ultra
17. X post by @antigravity
18. X post by @antigravity
19. X post by @antigravity
20. X post by @antigravity
21. X post by @timneutkens
22. X post by @petergostev
23. X post by @rileybrown
24. X post by @kentcdodds
25. X post by @steipete
26. X post by @steipete
27. X post by @steipete