DHH Goes Agent-First, PI Pushes Minimalism, and GPT-5.4 Wins a Brutal Benchmark

Coding Agents Alpha Tracker

2026-04-09

DHH Goes Agent-First, PI Pushes Minimalism, and GPT-5.4 Wins a Brutal Benchmark

By Coding Agents Alpha Tracker • April 9, 2026

DHH's production workflow is the clearest practical signal today: start with agents, review diffs, and keep two models running in parallel. Also: PI's minimalist harness philosophy, a punishing GPT-5.4 vs Opus benchmark, and cost-conscious automation tricks from Kent C. Dodds.

TOP SIGNAL

DHH has gone from disliking autocomplete-style AI to running an **agent-first** workflow in production: he now starts new work with an agent draft, reviews diffs in Neovim/Lazygit, and keeps a second model running in parallel for harder problems [1]. The broader takeaway from today's sources is even more useful: the best practitioners are not converging on full autonomy; they're converging on **lean harnesses, explicit review loops, and senior engineers as validators/redirection layers** [1, 2, 3].

> "Now I start with the agent. Now it'll give me the draft. I'll review the draft, and I'll make alterations if need be." [1]

TOOLS & MODELS

- **Model winner depends heavily on task + harness.** DHH says Opus 4.5 was the inflection point that made agent-first coding viable for him, and he still reaches for Opus on hard problems [1]. In a very different setup, Salvatore Sanfilippo's multi-day reverse-engineering benchmark had GPT-5.4/Codex doing **99.5-100%** of the work while Opus 4.6 mostly spun its wheels [2]. Theo's smaller prompt example points the same way: he says GPT/Codex treats prompts like instructions, while Opus sometimes treats them like a vibe [4].

- **PI** is the most interesting open-source harness signal today: ~4 built-in tools, a ~20-line system prompt, no automatic AgentMD/MCP clutter, self-customization via editing its own source and `/reload`, plus a TypeScript extension system for building custom agents/TUIs on top [3].
- **Cursor** shipped two concrete agent updates: remote agents you can kick off from your phone onto a devbox, and BugBot code review that learns from PR activity and says **78%** of the issues it finds are resolved by merge [5, 6, 7].
- **Local review loops are getting first-class support.** CodeRabbit's CLI can now be called directly by an agent, returns structured JSON with issues + fixes, and is being framed as a pre-PR review layer by both creator coverage and Theo/Ben's discussion of agent-integrated review [8, 3].
- **OpenClaw** keeps leaning into local-model and provider flexibility: Peter Steinberger added support for `inferrs`, described as a super efficient TurboQuant inference server, and says he has spent significant time making local models easy to use in OpenClaw [9].

WORKFLOWS & TRICKS

- **Copy DHH's dual-model loop.** Run tmux with Neovim on the left, a faster model in one pane, Opus in another, and a terminal strip below. Start the task in an agent pane, watch the diff in Lazygit, then either commit immediately or edit the code yourself if the diff is close-but-not-right [10, 1].
- **Use agents for PR triage, not just greenfield code.** DHH's loop is: pass Claude a PR/issue URL, let it analyze, merge the small minority that are good as-is, ask for a clean-room rewrite when the problem is right but the implementation is wrong, and reject the rest. He says that got him through **100 PRs in 90 minutes** [1].
- **For hard problems, make two models argue before either writes code.** DHH asks one model for a plan, sends that plan to another model for critique, then ping-pongs a couple more rounds before execution [1].
- **Queue your agent-triggered CI.** DHH says concurrent all-core local CI runs from multiple agents were overrunning his machine, so they added a simple **"WAIT YOUR TURN"** line for agents [11].
- **Voice-to-deploy is already real for small projects.** Kent bought a domain on Cloudflare, used Claude speech-to-text to tell Kody what he wanted, and Kody built/deployed a Cloudflare Worker landing page with Kit signup integration and an OG image from Cloudflare Browser Rendering [12, 13, 14].
- **Cost discipline matters.** Kent shut off OpenClaw after it started costing real money, then rebuilt the two features he needed with Kody + Cloudflare infra; his explicit tradeoff is that MCP is more limiting, but it can keep usage inside the AI subs he's already paying for [15, 16, 17].
- **Timeless harness rule: keep the core loop lean.** Ben's PI case is

blunt: fewer tools and smaller prompts work better; don't dump LSP noise into every turn, let the agent finish its generation, then run lint/checks afterward [3].

- **Common agent grammar is converging.** Simon Willison notes that file tools like `view` / `insert` / `str_replace` and "sub-agent as a tool" patterns are showing up outside Claude-style coding harnesses too, which is a good hint at which abstractions may stick [18].

PEOPLE TO WATCH

- **DHH** — High-signal because he's showing an actual senior-engineer production loop, not just hot takes: agent-first starts, diff review, PR triage, CLI design for agent interoperability, and a clear view on where human review still matters [1].
- **Salvatore Sanfilippo** — One of the few people running long, reproducible agent benchmarks on weird real systems work instead of toy app demos; his GPT-5.4 vs Opus emulator test is worth reading for methodology alone [2].
- **Kent C. Dodds** — Useful because he keeps turning agent talk into concrete side-project automation: NAS scripts, tunnels, Cloudflare infra, voice-driven landing pages, and cost-conscious rewrites when a setup gets too expensive [19, 12, 16].
- **Theo + Ben** — Watch them when you want harsh negative signal on harness design. Their main argument today: Claude Code is bloated, PI is minimal, and model quality is only half the story if the execution layer is wasting tokens and polluting context [3].

WATCH & LISTEN

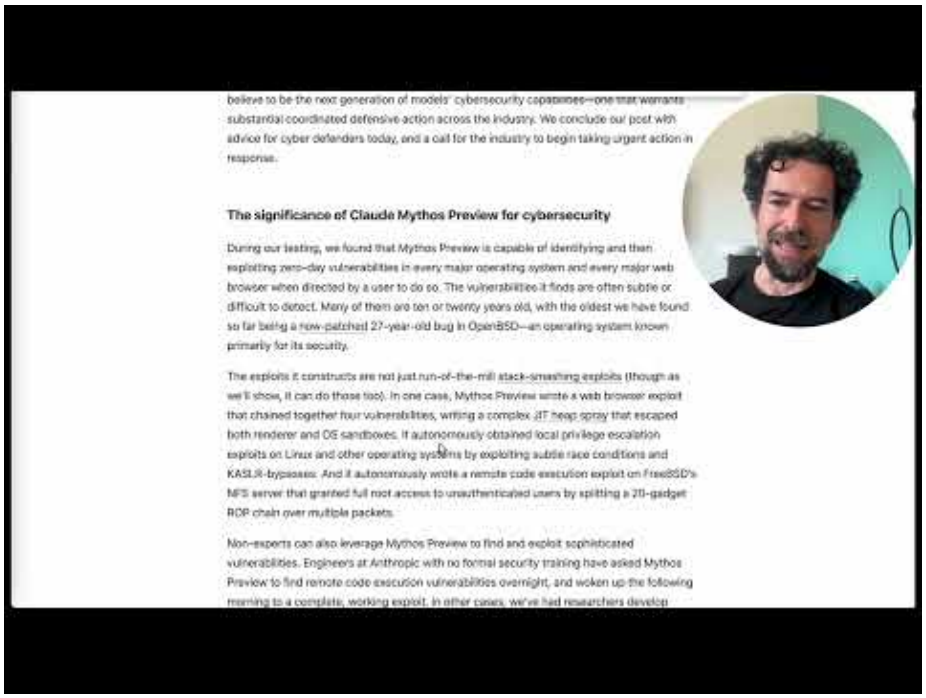
- **45:38-48:06 — DHH's dual-model layout.** One fast model, one stronger model, Neovim in the middle, and human review on the diff instead of autocomplete-driven coding [1].

DHH's new way of writing code (45:37)

- **1:05:22-1:07:15 — DHH's PR triage loop.** `review URL`, merge the clean ones, clean-room the fix when the idea is right but the code is wrong, and move on fast [1].

DHH's new way of writing code (65:22)

- **30:52-32:59 — Human steering still matters.** Salvatore explains how non-technical nudges—not hand-holding, just expert steering—helped GPT-5.4 break out of plateaus in a days-long emulator reconstruction task [2].



Mythos preview, e a seguire un test tra GPT 5.4 e Opus 4.6 (30:52)

PROJECTS & REPOS

- **PI** — Open-source minimal agent harness with a small team behind it; Ben rebuilt his BTCA research agent and custom TUI around its SDK/extensions because it avoids auto-loading extra context and makes custom tools easier to control [3].
- **OpenClaw** — Latest release adds `inferrs` support for efficient local inference, while Peter continues pushing local-model usability. The reality check: users like Kent are also finding that unattended agent setups can get expensive fast [9, 15, 16].
- **37signals' internal CLI push** — Not open source, but worth watching as a project pattern: DHH says they're building CLIs for Basecamp, HEY, and Fizzy so agents can pipe work across tools like Sentry, GitHub, and Basecamp with a clean record of what happened [1].

Editorial take: the edge right now is not "more autonomous agents" — it's better harnesses, tighter review loops, and humans who know when to redirect the model. [1, 3]

Sources

1. DHH's new way of writing code
2. Mythos preview, e a seguire un test tra GPT 5.4 e Opus 4.6
3. Crashing out at Anthropic and getting Pi pilled
4. X post by @theo
5. X post by @cursor_ai
6. X post by @cursor_ai
7. X post by @cursor_ai
8. Google just casually disrupted the open-source AI narrative…
9. X post by @steipete
10. X post by @GergelyOrosz
11. X post by @dhh
12. X post by @kentcdodds
13. X post by @kentcdodds
14. X post by @kentcdodds
15. X post by @kentcdodds
16. X post by @kentcdodds
17. X post by @kentcdodds
18. Meta's new model is Muse Spark, and meta.ai chat has some interesting tools
19. X post by @kentcdodds