Agent-First Coding Spreads as Teams Tighten Tests, Context, and Permissions

Coding Agents Alpha Tracker

2026-03-23

Agent-First Coding Spreads as Teams Tighten Tests, Context, and Permissions

By Coding Agents Alpha Tracker • March 23, 2026

Practitioners are moving from coding-agent experiments to agent-first workflows: DHH starts new customer work with agents, Tibo says Codex is helping refactor Codex, and the strongest tactics today were faster harnesses, explicit Skills for fresh releases, and safer enterprise rollout patterns. Also inside: Cursor Composer 2 pricing and speed details, GPT-5.4's frontend gap, and the clips worth watching.

TOP SIGNAL

The strongest signal today: coding agents are becoming the default starting point for real work, not just a side tool. DHH says all new customer work now starts with agents that he steers and calibrates, while OpenAI engineer Tibo Sottiaux says the Codex team is using Codex to help refactor an end-to-end systems rethink that would otherwise take months [1, 2].

The practical edge is shifting away from raw model IQ and toward loop design: faster harnesses, fresher context, and safer deployment controls [3, 4, 5].

TOOLS & MODELS

- **Cursor Composer 2** — Cursor's new code-only model is built from open-weight **Kimi 2.5** and then heavily post-trained/RL-tuned on Cursor's own data and harnesses [6]. Theo says it beats Opus on multiple coding evals, including Terminal Bench 2 in the **4.5-4.6** range, while running at **80-100 tokens/sec** and pricing at **$0.50/M input** and **$2.50/M output** [6]. Cursor is going through **Fireworks** for inference, and Theo says that path handles the attribution/license requirements for large-scale commercial use of the Kimi base [6].

- **GPT-5.4** — Theo's field report is blunt: it's an **incredible model for coding**, but still **a generation behind on frontend design**. His read on OpenAI's frontend best-practices post: useful advice, but not proof the design gap is solved [7, 8].
- **Claude Skills / Claude Code for web** — Simon Willison used **Claude Skills** to teach Claude the minor breaking changes in **Starlette 1.0**, because the model wasn't familiar with the release yet [4]. His caveat: Claude chat has an **"add to skills"** flow, but Claude Code for web apparently does not [9].
- **Devin** — swyx says Devin usage has grown **>50% MoM every month this year** [10]. More important than the growth chart: his deployment note that enterprise rollouts need permission models that won't terrify compliance and IT teams across **10,000s of engineers** [5].

WORKFLOWS & TRICKS

- **Let the agent write the first draft. Keep yourself on steering.** DHH says he writes no fresh customer code himself now; new work starts with agents, and he handles direction and calibration [1].
- **Patch fresh-release blind spots with explicit context.** If the framework version is newer than the model's knowledge, write the breaking changes into a Skill or context file before asking for edits. Simon did this for **Starlette 1.0** [4]. Read: Simon's writeup [4].
- **Use agents as ops translators, not just coders.** DHH says he injects agents into Linux systems and uses them constantly to decode obscure error messages. If you know "some Linux" but not enough to debug quickly, this is a high-ROI use case [1].
- **Speed up the harness before you scale the loop.** Peter Steinberger focused on tests and cut OpenClaw's harness runtime from roughly **10 minutes** to **2 minutes** [3]. Faster evals mean more agent iterations per day and less dead time between runs.
- **Split logic and tests into separate files/domains before you unleash automation.** Geoffrey Huntley hit **50 open PRs from automation** and says merge conflicts become a major source of waste if logic and tests are entangled [11].
- **Give non-engineers agent access — but only with safe permissions.** swyx argues designers should get direct access to coding agents, and extends that to PMs and analytics via Slack-style workflows [12, 13]. Pair that with enterprise-safe permission controls, not shortcut flags, if you expect the setup to survive real IT review [5].

  "Give your designer access to your coding agent. It is imperative…" [12]

PEOPLE TO WATCH

- **David Heinemeier Hansson** — high-signal because this is long-time operator commentary, not bench-racing: agent-first for new customer code, heavy use in Linux ops, and experiments with hold-to-talk voice input in his Linux setup [1].
- **Tibo Sottiaux** — short post, big signal: the Codex team is using Codex to help refactor its own system during an end-to-end scalability rethink [2].
- **Simon Willison** — still one of the best examples of practical context management. Today's lesson: when model knowledge lags a fresh OSS release, teach the model explicitly before trusting edits [4, 9].
- **Theo** — worth tracking because he separates "strong coding model" from "strong frontend model," and his Composer 2 breakdown added concrete cost/speed details instead of generic hype [7, 6].
- **swyx** — useful pulse on where coding agents are spreading inside orgs: designers, PMs, analytics teams, and enterprise deployment staff — not just core engineers [12, 13, 5].

WATCH & LISTEN

- **15:25-15:57 — DHH on using agents as Linux translators.** Short clip, practical point: this is the cleanest real-world case today for using an agent to decode obscure infra errors when you're not a deep Linux expert [1].

David Heinemeier Hansson: SaaS som vi kender det er FORBI.. (15:25)

- **49:54-50:20 — DHH on hold-to-talk voice prompting.** He describes a voice-to-model flow inside his Linux setup where a button press turns dictation into clean text. Worth watching if your next bottleneck is input speed, not model quality [1].

David Heinemeier Hansson: SaaS som vi kender det er FORBI.. (49:54)

PROJECTS & REPOS

- **OpenClaw** — the strongest repo signal today was operational, not social: Peter Steinberger got the harness from ~**10 min** to ~**2 min** by focusing on tests [3]. Separate signal: another user used **Claude Code** plus Google's live browser control to interact with the OpenClaw web dashboard for debugging [14].
- **Starlette 1.0** — not an agent project, but a useful OSS release case for agent users: Simon had to explicitly teach Claude the **1.0** breaking changes because the model lagged the release [4]. Expect this pattern on newly shipped framework versions.

Editorial take: the edge is moving from "which model is smartest?" to "who has the tightest loop" — fast tests, explicit fresh context, safe permissions, and agents in more hands [3, 4, 5, 12].

Sources

1. David Heinemeier Hansson: SaaS som vi kender det er FORBI..
2. X post by @thsottiaux
3. X post by @steipete

4. X post by @simonw
5. X post by @swyx
6. Did Cursor really just rebrand Kimi???
7. X post by @theo
8. X post by @theo
9. X post by @simonw
10. X post by @swyx
11. X post by @GeoffreyHuntley
12. X post by @swyx
13. X post by @swyx
14. X post by @steipete