Composer 2 Arrives as Cross-Agent and Test-Hardened Workflows Mature

Coding Agents Alpha Tracker

2026-03-20

Composer 2 Arrives as Cross-Agent and Test-Hardened Workflows Mature

By Coding Agents Alpha Tracker • March 20, 2026

Cursor's Composer 2 and Glass launch drove the release chatter, but the strongest practitioner signal was elsewhere: cross-tool agent orchestration, contained optimization loops with brutal tests, safer shell sandboxes, and honest task-by-task model comparisons.

TOP SIGNAL

Today's highest-signal workflow came from Redis creator Salvatore Sanfilippo: use LLMs to take on *self-contained* optimizations, not architectural sprawl [1]. His rule is simple: first make the test suite brutally hard to pass, then use the model on a contained data-structure or algorithm change that keeps the external API stable but can materially improve speed or memory use [1]. He argues that's where AI is genuinely changing programming economics [1].

TOOLS & MODELS

- **Cursor — Composer 2**: now live in Cursor, priced at **$0.50/M input + $2.50/M output** on standard and **$1.50/M input + $7.50/M output** on fast [2, 3]. Cursor says its first continued pretraining run improved quality and lowered cost to serve, giving it a stronger base for RL [4]. Founders frame it as frontier-level and explicitly coding-only after a year of model-training effort [5, 6, 7]. More: Composer 2 blog [8]
- **Early Composer 2 read from practitioners**: Kent C. Dodds says it is not quite as good as **GPT 5.4**, but it is much **faster and cheaper** [9]. Theo says it is already very good [10], while @koylanai says it is especially strong at **long, grounded, tool-mediated** research/context work and beat **Opus 4.6** and **GPT 5.4** on a transcript-to-reading-list

task [11]. Jediah Katz adds one sleeper feature: ask Cursor about your **past conversations** [12].

- **Cursor — Glass alpha**: Cursor also opened an early alpha of **Glass**, its simplified interface [13]. Kent says it feels like a marriage between the web portal and the local IDE and is likely where most agentic coding tools are heading [14]. Theo agrees the UI reset was overdue and likes the **ACP** support [10]. More: Glass alpha [13]
- **Claude surfaces are spreading**: **T3 Code** now supports the **Claude Code CLI** for users who already have it installed and signed in [15]. Anthropic also released **Claude Code channels** for controlling sessions through **Telegram** and **Discord**, including from your phone [16]. At the same time, **opencode 1.3.0** stopped autoloading its Claude Max plugin after Anthropic legal pressure, and the plugin was removed from GitHub / deprecated on npm [17].
- **Hard-debugging signal**: in a real Ghostty/GTK case, **Codex 53 extra high** solved a bug Mitchell Hashimoto's team had struggled with for **over six months** from a vague prompt, while lower Codex reasoning levels and **Opus 46** failed; the Opus run reportedly cost **$4.14** and took **45 minutes** [18].
- **Do not over-generalize from one model story**: Simon Willison says **Opus 4.5** earned his trust on familiar tasks like JSON APIs, and that **Opus 4.6 / Codex 5.3** feel close to one-shot reliable for many routine jobs [19]. Theo, meanwhile, reports letting **Opus** run for over an hour on a new feature only to learn 20 minutes later that the whole implementation was wrong [20].

WORKFLOWS & TRICKS

- **Cross-agent handoff**: Kent built a personal assistant agent that works across **ChatGPT, Claude, Cursor**, and any MCP-compatible interface [21]. His demo workflow was practical: ask **Claude in the browser** to create a GitHub issue, then have it fire off a **Cursor cloud agent** to solve it [21]. If you are punting work for later, he also recommends dumping all current context into the GitHub issue so resumption is trivial [22]. Under the hood, his setup uses Cloudflare's Dynamic Worker Loader so the agent can write code, plus capability search and reusable skills [23, 24].
- **Teach through repo files**: Kent says linking `testing-principles.md` from `agents.md` was enough to get his agent using `Symbol.asyncDispose` correctly for test setup [25]. Simon's version of the same idea is structural: start from **cookiecutter** templates with tests, CI, and README in place so the agent copies the right patterns from the first commit [19].
- **Contained optimization loop**: Salvatore's playbook is worth stealing for hot paths: **(1)** harden the test suite until wrong code is brutally hard to sneak through, **(2)** let the model handle a contained algorithm/data-structure change, and **(3)** only pay added complexity when the win is material and the subsystem API stays stable [1].

- **Human 20% still matters**: the Mitchell Hashimoto GTK story is a clean pattern for ugly bugs. Let the agent do the tedious repo archaeology across issues, patches, and source trees, then do the targeted code review, failure-mode questions, and cleanup yourself [18].

  "Poor quality code from an agent is a choice that you make." [19]

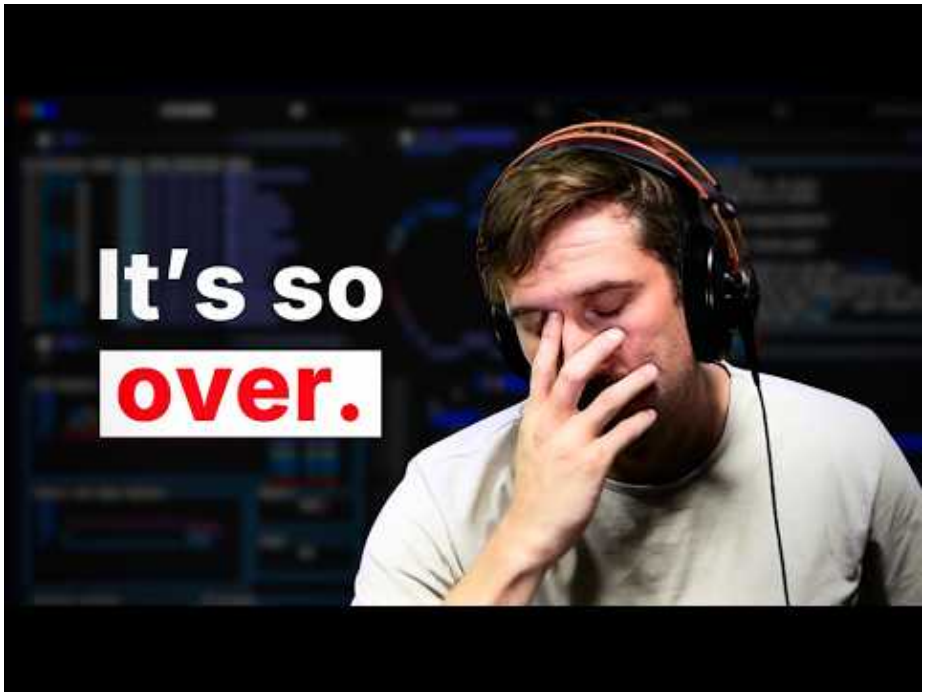
- **Never hand the agent real Bash if a fake shell will do**: **Just Bash** gives agents a Bash-like environment in TypeScript with an in-memory filesystem inside a JavaScript VM because agents are good at shell interactions but real shell access is risky and expensive [26]. Its defense-in-depth disables dangerous JS execution paths and checks for prototype-pollution-style escapes; the broader rule is simple: put agents in a sandboxed runtime, not your host OS [26].
- **Long context may be less broken than the discourse suggests**: Kent says he does **not** see the often-repeated failure in the last **40%** of context when using **Cursor** mostly with **GPT 5.4** or long **ChatGPT** threads, and credits Cursor's compaction for holding up [27]. He also notes he does not use **Claude Code** or **Open Code** much, so his exposure may be narrower [28].

PEOPLE TO WATCH

- **Kent C. Dodds** — one of the clearest operator feeds right now: cross-tool MCP orchestration, GitHub issues as context handoff, repo-guided agent behavior, and a useful counterpoint on long-context reliability [21, 22, 25, 27].
- **Simon Willison** — still the best mix of daily-driver pragmatism and security realism. He says he now writes more code on his phone than on his laptop [19], trusts Opus on familiar tasks [19], and keeps hammering on prompt injection, the lethal trifecta, and sandboxing [19].
- **Theo** — worth following because he ships tools and does not hide the misses: positive on Glass and T3 Code's Claude support, bluntly negative when a model wastes an hour, and generally honest about where the UI is headed [10, 15, 20].
- **Salvatore Sanfilippo** — the most thoughtful systems-programming take of the day. He is not talking about toy app scaffolding; he is talking about when LLMs make complex data-structure work worth attempting in production code [1].
- **swyx** — useful security signal: he argues identity-based authorization is the key way to break the binary between **HITL everything** and **dangerously skip permissions**, and points to Keycard plus similar work from WorkOS/Auth0/Cloudflare [29, 30].

WATCH & LISTEN

- **1:30-4:35 — Codex on a six-month GTK bug**: best proof today for AI as a research mule. The agent works through the issue, patches, and finally the **GTK4** source before proposing the fix the other runs missed [18].

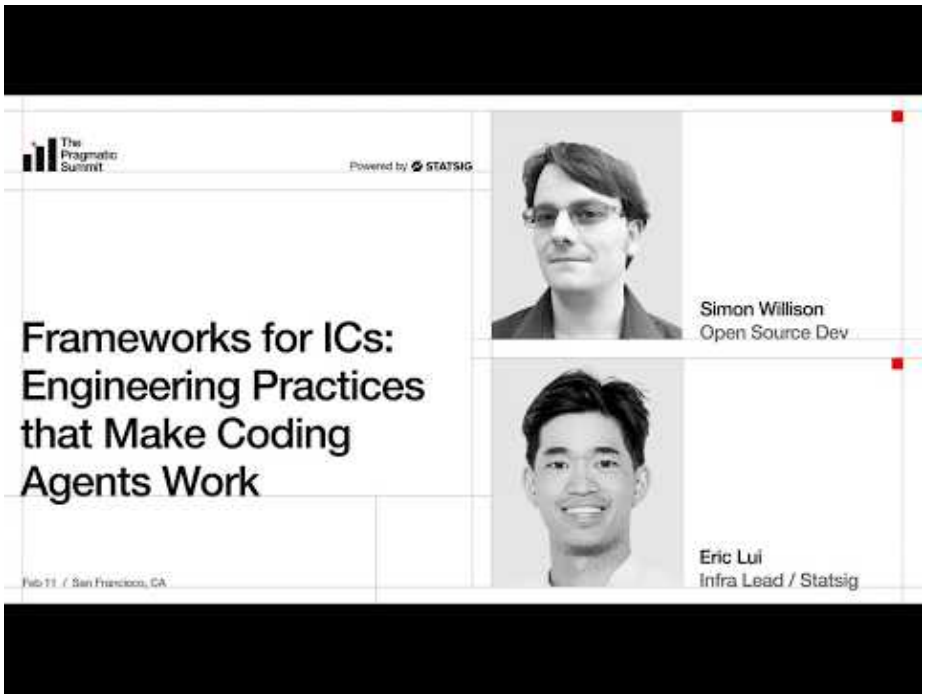


we're so back (1:29)

- **9:11-11:35 — Salvatore on self-contained optimization**: if you work near hot paths, watch this. He lays out when added complexity is worth paying now that LLMs can help shoulder implementation and corner-case load [1].

Le semplificazioni avventate sono … (9:10)

- **1:45-2:28 — Simon's tiny benchmark prompt**: one short prompt — `run a benchmark and then figure out the best options for making it faster` — got his Python WebAssembly engine a **45-49% Fibonacci speedup** [19].



Simon Willison: Engineering practices that make coding agents work - The Pragmatic Summit (1:45)

PROJECTS & REPOS

- **Just Bash / Cloudflare Shell** — the strongest open-project signal today. Vercel's **Just Bash** gives agents a Bash-compatible environment in TypeScript with an in-memory filesystem [26]. Cloudflare's Sunil Pye praised it, Cloudflare forked it into **Cloudflare Shell**, and Dane says he is already using it for an internal CTO agent [26].
- **Showboat** — Simon Willison's new tool is only about **48 hours old** at recording, but the use case is excellent: agents can run manual API checks with `curl` and produce a Markdown log of each step and output [19].
- **Keycard for Coding Agents** — worth watching because it targets a real failure mode: coding agents inherit your credentials and many identity systems cannot distinguish you from the agent acting in your name [31]. swyx says Keycard now supports all coding agents and frames identity-based authz as the most important security direction here [29].
- **uv / ruff / ty** — not new, but increasingly relevant agent tooling. Simon says fast linting and type-checking resonate with coding agents, and he has made `uv run` an essential part of his workflow; he is skeptical that these tools need to live *inside* the agent as opposed to being called by it [32].

Editorial take: the durable edge today was not a single model release — it was

tighter loops: hard tests, contained complexity, safer sandboxes, and agents that can hand work to each other.

Sources

1. Le semplificazioni avventate sono …
2. X post by @cursor_ai
3. X post by @cursor_ai
4. X post by @cursor_ai
5. X post by @mntruell
6. X post by @sualehasif996
7. X post by @amanrsanger
8. X post by @cursor_ai
9. X post by @kentcdodds
10. X post by @theo
11. X post by @koylanai
12. X post by @jediahkatz
13. X post by @cursor_ai
14. X post by @kentcdodds
15. X post by @theo
16. X post by @trq212
17. X post by @thdxr
18. we're so back
19. Simon Willison: Engineering practices that make coding agents work - The Pragmatic Summit
20. X post by @theo
21. X post by @kentcdodds
22. X post by @kentcdodds
23. X post by @kentcdodds
24. X post by @kentcdodds
25. X post by @kentcdodds
26. Vercel accuses Cloudflare of stealing
27. X post by @kentcdodds
28. X post by @kentcdodds
29. X post by @swyx
30. X post by @swyx
31. X post by @KeycardLabs
32. Thoughts on OpenAI acquiring Astral and uv/ruff/ty