Verification-first coding agents: Composer 1.5, spec-driven dev, and why proof beats diffs

Coding Agents Alpha Tracker

2026-03-07

Verification-first coding agents: Composer 1.5, spec-driven dev, and why proof beats diffs

By Coding Agents Alpha Tracker • March 7, 2026

Today's signal across Cursor and Augment is clear: as agents generate bigger diffs, the winning teams shift from "review code" to "verify outcomes" with agent-run tests, spec-driven loops, and guardrails. Plus: Cursor's Composer 1.5 details, Codex Security/OSS programs, and a concrete cloud-agent PR that shipped in 15 minutes.

TOP SIGNAL

Verification is becoming the core product feature of coding agents—not an afterthought. Cursor argues cloud agents won't scale until the model can *test its own code and prove it works* (otherwise you return to humans a giant diff they can't trust) [1]. Augment is converging on the same idea via **spec-driven development + a dedicated verification agent + robust CI/CD** [2].

TOOLS & MODELS

- **Cursor — Composer 1.5 model release**
  - Cursor describes Composer 1.5 as **between Sonnet 4.5 and Opus 4.5** in capability, trained "almost entirely" with lots of RL [3].
  - Design goal: **fast, engaging** usage—not "press Enter and go to sleep" [4].

[1] Lessons from Building Cursor
[2] The Future Live | 03.06.26 | Guests from Augment Code, NEAR Protocol, and Modular!
[3] Lessons from Building Cursor
[4] Lessons from Building Cursor

  - Integrated capabilities Cursor wants *inside* the model: better **GREP**, strong **semantic search** for large codebases (finding the right place in **1–3 queries vs tens**) and training toward **recursive subagents** to resolve most queries in **<2–3 minutes** [5].
- **Cursor — Cloud agents need product step-changes, not UI polish**
  - Cursor says cloud agents today feel worse than local (slow setup/boot, hard to see changes) and highlights the core failure mode: you come back to a **1000-line diff** and it's still your job to determine mergeability/correctness [6].
  - Reported adoption signal: when the agent can **test its own code and prove correctness**, they've seen cloud agent usage jump by **10×** [7].
  - Cursor's mental model: cloud-agent compute is ~**1%** of local today; getting to **90%** implies **1000×** growth, which likely requires step-function capability changes [8].
- **OpenAI — Codex Security (research preview)**
  - OpenAI introduced **Codex Security**, an application security agent that finds vulnerabilities, validates them, and proposes fixes for you to review and patch [9].
  - Positioning: helps teams focus on "vulnerabilities that matter" and ship faster [10].
  - Link: https://openai.com/index/codex-security-now-in-research-preview/ [11]
- **OpenAI — Codex for Open Source**
  - New program aimed at OSS maintainers: use Codex to **review code, understand large codebases, and strengthen security coverage** [12].
  - Apply: https://openai.com/form/codex-for-oss/ [13]
  - Docs: http://developers.openai.com/codex/community/codex-for-oss [14]
- **Codex usage/cost notes (from @thsottiaux)**
  - **/fast mode**: **1.5× inference speed** at **2× token usage** [15].
  - **GPT-5.4** token cost is advertised as **30% higher** than GPT-5.2 and GPT-5.3-Codex; they say they're not seeing evidence of additional excess usage beyond that [16].

[5] Lessons from Building Cursor
[6] Lessons from Building Cursor
[7] Lessons from Building Cursor
[8] Lessons from Building Cursor
[9] post by @OpenAIDevs
[10] post by @OpenAIDevs
[11] post by @OpenAIDevs
[12] post by @OpenAIDevs
[13] post by @romainhuet
[14] post by @OpenAIDevs
[15] post by @thsottiaux
[16] post by @thsottiaux

  - Investigating reports of unexpected higher drain when **WebSockets** are enabled [17].
- **GPT-5.4 capability anecdotes worth calibrating against your own evals**
  - Mark Chen: giving GPT-5.4 a raw dump of **GPT-2 weights** and asking for a **<5000 byte C program** to run inference succeeded in **under 15 minutes** [18]; a similar exercise in a previous paper took **days** [19].
  - QuixiAI (shared by Greg Brockman): GPT-5.4 showed a boost in "understanding and ability to solve problems quickly and completely," including building a compiler where **Claude Code** was "pretty much stumped" [20].
  - Hanson Wang: GPT-5.4 and GPT-5.3-Codex perform strongly on **Terminal-Bench**, with GPT-5.4 solving a previously-unsolved hard task ("gpt2-codegolf") [21].
- **Language targeting anecdote (Claude/Opus)**
  - DHH: in a language shoot-out for Claude code generation, **Opus + Ruby** produced the best output (fewest tokens, fewest LOCs, fastest completion) [22].

WORKFLOWS & TRICKS

- **Pattern: "Make the agent prove it" (cloud agents + CI)**
  - Cursor's critique of today's cloud agents: they hand you a huge diff and you still have to decide correctness—Cursor says that feels "fundamentally wrong" [23].
  - Cursor's proposed step change: have the model **test its code** and **prove** it did the thing correctly [24].
  - Practical implication for teams: invest in developer experience so agents can act like a new engineer who *doesn't know tribal knowledge* (e.g., service boot order) [25].
- **Spec-driven + verification agent + robust release machinery (Augment's production loop)**
  - Augment describes going fully **spec-driven**, with humans aligning across a **hierarchy of specs**, then having agents refine toward implementation specs [26].
  - They pair this with a dedicated **verification agent** plus CI/CD

[17] post by @thsottiaux
[18] post by @markchen90
[19] post by @markchen90
[20] post by @QuixiAI
[21] post by @hansonwng
[22] post by @dhh
[23] Lessons from Building Cursor
[24] Lessons from Building Cursor
[25] Lessons from Building Cursor
[26] The Future Live | 03.06.26 | Guests from Augment Code, NEAR Protocol, and Modular!

stages (unit/system tests, feature flags, canaries) and treat a robust pipeline as non-optional [27].
  - Code review scaling idea: shift to agents reviewing most changes and escalating a smaller slice to humans (they describe aiming for agents to review ~80% and flag ~10–20% for humans, potentially shrinking further) [28].
- **Agentic manual testing (new chapter from Simon Willison)**
  - Willison's pattern: have agents "manually" try out the code to catch issues that automated tests miss [29].
  - Link: https://simonwillison.net/guides/agentic-engineering-patterns/agentic-manual-testing/ [30]
- **Infra footgun reminder: don't let agents free-fire Terraform**
  - A production incident report: **Claude Code** ran a Terraform command that wiped a production database, taking down the DataTalksClub course platform and deleting **2.5 years** of submissions; automated snapshots were also gone [31].
  - Recovery note (via @simonw): "Thankfully… the full recovery took about **24 hours**" [32].
  - Full timeline + prevention changes (author): https://alexeyondata.substack.com/p/how-i-dropped-our-production-database [33][34]
- **Concrete "cloud agent shipped it" example (Cursor)**
  - Kent C. Dodds: Cursor cloud agents implemented a diff-view upgrade (line diffs → **character-level highlights**) by migrating to diffs.com [35][36].
  - He reports: initial prompt + **7 follow-ups**, "robots" reviewed/iterated, and he merged—**15 minutes of his time** [37][38].
  - PR: https://github.com/epicweb-dev/epicshop/pull/577 [39]
- **A practical "build loop" doc you can copy-paste (Ben Tossell)**
  - Minimal process: create `/spec/` folder, name specs (`00_spec1`), track progress in `progress.md`, enforce a **test gate**, dogfood in an agent-browser before handing you a URL, "debug until green" [40].

[27] The Future Live | 03.06.26 | Guests from Augment Code, NEAR Protocol, and Modular!
[28] The Future Live | 03.06.26 | Guests from Augment Code, NEAR Protocol, and Modular!
[29] post by @simonw
[30] post by @simonw
[31] post by @Al_Grigor
[32] post by @simonw
[33] post by @Al_Grigor
[34] post by @Al_Grigor
[35] post by @kentcdodds
[36] post by @kentcdodds
[37] post by @kentcdodds
[38] post by @kentcdodds
[39] post by @kentcdodds
[40] post by @bentossell

PEOPLE TO WATCH

- **Sualeh Asif (Cursor, "Lessons from Building Cursor")** — unusually specific on what gets trained *into* the model (GREP/semantic search/subagents) and why cloud agents need proof, not diffs [41][42].
- **Vinay (Augment)** — concrete production patterns for agent-first teams: spec hierarchies, verification agents, and treating CI/CD as the real safety net [43].
- **Simon Willison** — keeps the conversation grounded in what actually catches bugs: agent-assisted *manual* testing as a complement to automated suites [44].
- **Kent C. Dodds** — high-signal "minutes-to-merge" cloud agent workflow, with a real PR you can inspect [45][46].
- **@thsottiaux (Codex)** — practical cost/speed tradeoffs and ongoing investigation notes for usage drain with WebSockets enabled [47][48].

WATCH & LISTEN

1) Cursor: why cloud agents are stuck until they can test + prove correctness (05:42–10:13)

Hook: the "1000-line diff" problem, why it's backwards to make humans certify correctness, and why agent-run testing is the step-change.

[41] Lessons from Building Cursor
[42] Lessons from Building Cursor
[43] The Future Live | 03.06.26 | Guests from Augment Code, NEAR Protocol, and Modular!
[44] post by @simonw
[45] post by @kentcdodds
[46] post by @kentcdodds
[47] post by @thsottiaux
[48] post by @thsottiaux

Lessons from Building Cursor (5:41)

2) Cursor: infra for long-running agents (minutes → days) + why Temporal-like systems matter (10:26–12:37)

Hook: agents break the old RPC mental model; monitoring and deploys get weird when tasks run for hours.

Lessons from Building Cursor (10:26)

3) Augment: spec-driven development + integrated verification loops (25:50–28:00)

Hook: how they structure specs so humans align first, agents implement next, and verification runs continuously (not "later").

The Future Live | 03.06.26 | Guests from Augment Code, NEAR Protocol, and Modular! (25:49)

PROJECTS & REPOS

- **T3 Code** (open source, Codex-CLI-based) — released publicly by Theo; designed for running many agents in parallel, and explicitly motivated by CLI scaling limits [49][50].
  - Try: http://t3.codes or `npx t3@alpha` [51]
  - Claude support via Agent SDK is planned; PR is ready but waiting on approval [52].
  - Adoption signal: "Nearing **2,000 users in 1 hour**" [53].
- **OpenAI: Harness Engineering write-up** — "steering Codex" to open/merge **1,500 PRs** with **zero manual coding** for a product used by hundreds of internal users [54][55].
  - https://openai.com/index/harness-engineering/ [56]

[49] post by @theo
[50] post by @theo
[51] post by @theo
[52] post by @theo
[53] post by @theo
[54] post by @OpenAIDevs
[55] post by @OpenAIDevs
[56] post by @OpenAIDevs

- **Agentic manual testing (guide chapter)** — a reusable pattern, not a product launch: https://simonwillison.net/guides/agentic-engineering-patterns/agentic-manual-testing/ [57]

Editorial take: Output is cheap now; the real differentiator is **proof**—verification loops, repo devex, and hard guardrails around what agents are allowed to break. [58][59][60]

Sources

1. Lessons from Building Cursor
2. The Future Live | 03.06.26 | Guests from Augment Code, NEAR Protocol, and Modular!
3. post by @OpenAIDevs
4. post by @OpenAIDevs
5. post by @romainhuet
6. post by @thsottiaux
7. post by @thsottiaux
8. post by @thsottiaux
9. post by @markchen90
10. post by @QuixiAI
11. post by @hansonwng
12. post by @dhh
13. post by @simonw
14. post by @Al_Grigor
15. post by @simonw
16. post by @kentcdodds
17. post by @kentcdodds
18. post by @bentossell
19. post by @theo
20. post by @theo
21. post by @theo
22. post by @theo
23. post by @theo
24. post by @OpenAIDevs

[57] post by @simonw
[58] Lessons from Building Cursor
[59] The Future Live | 03.06.26 | Guests from Augment Code, NEAR Protocol, and Modular!
[60] post by @Al_Grigor