

Cloudflare Pushes Code-Writing Agents as Claude Code and Cursor Tighten the Loop

Coding Agents Alpha Tracker

2026-03-25

Cloudflare Pushes Code-Writing Agents as Claude Code and Cursor Tighten the Loop

By Coding Agents Alpha Tracker • March 25, 2026

Cloudflare’s Dynamic Workers and Code Mode signal a more important shift than another model release: coding agents are moving from brittle tool menus toward fast, disposable code execution. Also inside: Claude Code auto mode, Cursor’s Composer 2 report, Riley Brown’s Codex/Xcode workflow, and production multi-agent patterns worth stealing.

TOP SIGNAL

The sharpest infra shift today is Cloudflare’s push toward **code-writing agents**. Dynamic Workers are now in Open Beta with secure sandboxes that start ~100x faster than containers and use 1/10 the memory, and Kent C. Dodds says he’s already using them to power his own AI assistant with CodeMode [1, 2]. The companion **Code Mode** thesis matters just as much: expose tools as an SDK and let the model write code to select and compose them, instead of stuffing more JSON-style tools into the context window [3, 4].

“Agents should interact with the world by writing code, not tool calls.” [1]

TOOLS & MODELS

- **Claude Code — auto mode.** Claude can now decide file-write and bash permissions on the user’s behalf, with safeguards checking each action before it runs. That replaces the old choice between constant approvals and `--dangerously-skip-permissions` [5, 6].
- **Cursor Composer 2 — training report out.** Cursor says Composer 2 was built with three main efforts: continued pretraining, reinforcement

learning, and benchmark development to emulate the Cursor environment. The practical takeaways: continued pretraining kept improving downstream coding performance, the RL phase was critical enough that simple approaches worked best broadly, and CursorBench is their internal benchmark for more realistic software engineering problems. The report also covers distributed training / RL infrastructure and kernels they open-sourced. Read: Composer 2 report [7, 8, 9, 10, 11, 12, 13].

- **Cloudflare Dynamic Workers + Code Mode.** Dynamic Workers entered Open Beta for paid Workers users, with disposable sandboxes that start ~100x faster than containers and use 1/10 the memory. The adjacent Code Mode posts surfaced three agent-facing building blocks: `codemode` for chewing through multiple tool calls in one/few shots, `shell` for a local filesystem backed by DO/D1/SQL/R2, and `worker-bundler` for packaging npm deps and serving full apps. Read: Dynamic Workers and Code Mode [1, 14, 3].
- **Cursor + Figma.** Cursor can now create new components and frontends in Figma using your team's design system, including variables, tokens, and naming conventions. This landed alongside Figma's open beta for AI agents designing directly on canvas via the `use_figma` MCP tool [15, 16, 17].
- **Devin as reviewer, not just author.** The clearest practitioner comparison today came from Gauri Gupta and swyx: both say Devin code review catches bugs Claude and Codex miss, and swyx says the review pass saves him 3-8x/day [18, 19].

WORKFLOWS & TRICKS

- **Replicable mobile-agent build loop from Riley Brown**
 1. Create the app shell in Xcode, then open the **top-level** project folder in Codex and start a dedicated thread [20].
 2. Paste the full product spec and hit **Shift+Tab** for Plan mode [20].
 3. Run design in parallel in Paper.design via MCP while Codex builds [20].
 4. Steer inline while the thread is running, then paste API keys, Xcode logs, and screenshots back into the chat when builds fail [20].
 5. Test on a physical iPhone, and keep the generated app preview open so you can keep iterating without leaving the app [20].
 - Bonus: Brown adds Whisper-based voice input so he can keep refining the preview without opening the keyboard [20].
 - *Timeless pattern:* split planning, design, implementation, and debugging across parallel agent threads instead of forcing one giant chat [20].
- **Turn prompts into loops, not one-shots**
 - Karpathy's Auto Research runs on a git feature branch: propose experiment, write code, run it, analyze results, commit improvements, repeat [21].

- Matthew Berman says he applies the same shape to overnight fine-tuning of small open models against an Opus 4.6 baseline until a local model wins [21].
- Minimax’s **M2.7** harness adds skills, memory, guardrails, and eval infra; it handled 30-50% of the workflow and reported a 30% performance gain on evals [21].
- Anthropic says similar Claude loops now power almost all of its major agent loops, with humans reviewing solutions before final refinements [21].
- **Production pattern worth stealing from Moda**
 - Use a cheap Haiku triage node to classify the job and preload task-specific Markdown skills [22].
 - Inject those skills as human messages, with caching breakpoints after the system prompt and after the skills block [22].
 - Keep only 12-15 core tools always in context; activate ~30 more on demand with `RequestToolActivation` [22].
 - For big projects, pass a high-level summary and let the agent pull details through tools; then use traces to tune prompt/tool changes, cost, cache hits, and failures [22].
- **Add a critic, not just another worker.** swyx says he won’t merge without a Devin review now, because it catches bugs Claude and Codex miss and acts like fresh eyes on a PR. Related design note: he points to Windsurf’s “**smart friend**” pattern as a model for subagents that are more critical instead of weaker [18, 19].

PEOPLE TO WATCH

- **Kent C. Dodds** — strongest practitioner confirmation on today’s Cloudflare story: he’s been using Dynamic Workers for a week to power his own AI assistant, and separately backs the “tools as SDK + agent-written code” pattern as the more effective direction [2, 4].
- **Riley Brown** — one of the most concrete end-to-end Codex workflows in the wild right now: Xcode + Codex + Claude Agent SDK + Vibe Code CLI + Paper.design, with screenshots and logs fed back into the loop [20].
- **swyx** — short posts, strong signal: Devin review outperforming Claude/Codex in his day-to-day, plus the “smart friend” subagent pattern callout [18, 19].
- **Ravi Parikh / Moda** — rare production detail instead of benchmark chatter: triage, dynamic tool loading, prompt caching, context scaling, and tracing from a real multi-agent design product [22].
- **Anthropic Engineering** — new post on a multi-agent harness for front-end design and long-running autonomous software engineering. Worth reading because it’s from the team running the stack, not a third-party teardown: harness-design-long-running-apps [23].

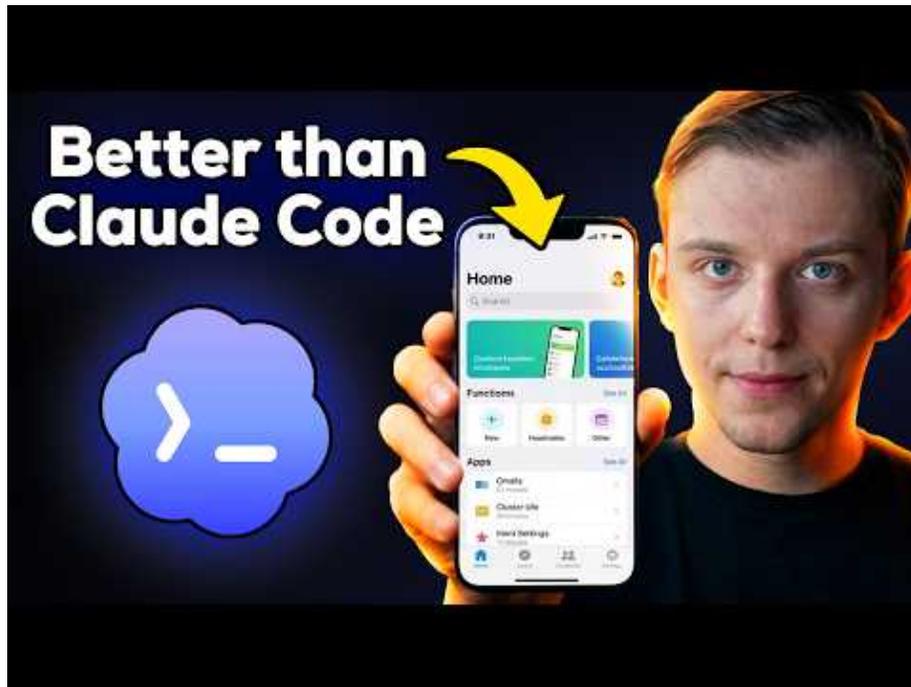
WATCH & LISTEN

- **1:56-4:28** — **M2.7’s harness loop.** Best compact explanation today of the human/harness/agent split: human steers direction, the harness supplies skills/memory/guardrails/evals, and the agent writes, runs, analyzes, and improves experiments [21].



Hard Takeoff has started (1:56)

- **3:36-5:43** — **Riley Brown’s Codex setup for “Jerry.”** Watch this if you want a replicable “agent builds the app that builds apps” flow: create the Xcode project, point Codex at the right folder, then fan out into parallel threads [20].



Building a Mobile App with OpenAI Codex (Better than Claude Code) (3:36)

- **25:16-26:17 — In-app preview + voice iteration.** Shorter clip, but high leverage: Riley tweaks the generated app while the preview is open and adds Whisper-based voice input so the iteration loop never leaves the phone UI [20].



Building a Mobile App with OpenAI Codex (Better than Claude Code) (25:16)

PROJECTS & REPOS

- **karpathy/auto-research** — GitHub repo for autonomous research loops on a git feature branch. Practical signal: Berman frames it as bringing frontier-style experiment loops to solo developers, and points to an overnight run that reached the fastest reported time to train one of these models [21].
- **langchain-ai/deepagents** — better adoption signal than most new repos because Moda is already using Deep Agents in production for its Research Agent and Brand Kit Agent, with the Design Agent migration under evaluation. Repo: Deep Agents [22].

Editorial take: today's edge is loop design — faster execution sandboxes, smaller always-on tool sets, explicit review passes, and tighter human feedback beat one giant prompt. [1, 22, 19, 20]

Sources

1. X post by @KentonVarda
2. X post by @kentcdodds
3. X post by @jpschroeder

4. X post by @kentdodds
5. X post by @claudeai
6. X post by @alexalbert__
7. X post by @cursor_ai
8. X post by @cursor_ai
9. X post by @cursor_ai
10. X post by @cursor_ai
11. X post by @cursor_ai
12. X post by @cursor_ai
13. X post by @cursor_ai
14. X post by @threepointone
15. X post by @cursor_ai
16. X post by @cursor_ai
17. X post by @figma
18. X post by @gauri__gupta
19. X post by @swyx
20. Building a Mobile App with OpenAI Codex (Better than Claude Code)
21. Hard Takeoff has started
22. How Moda Builds Production-Grade AI Design Agents with Deep Agents
23. X post by @AnthropicAI