Karpathy Stops Typing Code as Orchestration Becomes the New IDE

Coding Agents Alpha Tracker

2026-03-21

Karpathy Stops Typing Code as Orchestration Becomes the New IDE

By Coding Agents Alpha Tracker • March 21, 2026

Karpathy's near-100% delegation was the clearest workflow signal today, and multiple practitioners now agree the developer workspace is shifting from one editor window to orchestration surfaces built for parallel agents. Also inside: Cursor Composer 2's disclosed training stack, honest Codex vs. Opus field notes, and the most copyable workflow patterns from people shipping with agents.

TOP SIGNAL

Andrej Karpathy says his day-to-day has already crossed from AI pair programming to operating a small fleet: he hasn't typed code since December, now delegates non-interfering features to parallel agents, and thinks in "macro actions" over repos instead of line edits [1]. The bigger pattern is showing up from multiple angles: Addy Osmani argues the IDE is being de-centered into orchestration surfaces, and Theo says current editors break down because agentic work spans multiple projects, terminals, browsers, and worktrees at once [2, 3].

TOOLS & MODELS

- **Cursor Composer 2**: built on **Kimi k2.5**, which Cursor says was the strongest base on its perplexity-based evals. Cursor then did continued pretraining plus a **4x** high-compute RL scale-up on top, using Fireworks for RL and inference; Aman Sanger says only about **1/4** of final-model compute came from the base and full pretraining is planned later [4, 5, 6]. Cursor also says it missed crediting Kimi in the initial blog and will fix that next time [4].

- **The control plane is becoming the product**: Osmani's current stack includes Conductor, Claude Code Web/Desktop, GitHub Copilot Agent, Jules, Vibe Kanban, and cmux; his framing is that the editor is still critical, but no longer the front door [2]. He also flags Claude Code's new Swarm/agent-teams direction and notes that developer reaction to Cursor Glass was basically "this feels more like an agent orchestrator than an IDE" [7, 2].
- **Task-level model notes, not universal benchmarks**: Theo says Opus spent over an hour on a new feature and still got the implementation entirely wrong; Codex did the same feature correctly in 15 minutes [8, 9]. Karpathy, meanwhile, says Claude's coding agent has a better teammate-like personality while Codex feels dry, but his latest gripe is broader than model choice: agents still bloat abstractions, copy-paste, and ignore AGENTS.md style instructions [1, 10].

WORKFLOWS & TRICKS

- **Run repo work in macro-actions, not prompt-by-prompt**

  1. Split work into non-interfering feature chunks.

  2. Hand separate chunks to parallel agents across checked-out repos/workspaces.

  3. Use other agents for planning and research in parallel.

  4. Review output proportionally to how much you care about that path. Karpathy points to Peter Steinberger's setup with roughly **10 Codex agents** as the visual form of this pattern; each high-effort task runs about **20 minutes**, then you top them up and keep moving [1].

- **Treat unused quota as lost throughput**: Karpathy says if one tool/provider hits quota, switch to another; his default when agents fail is not "the capability isn't there" but "bad instructions, memory, or tooling" [1].
- **Set objective metrics and boundaries, then get out of the way**: Karpathy's AutoResearch loop improved a nanoGPT repo overnight by finding weight-decay/value-embedding and Adam-beta interactions he had missed; his Program.md is just a markdown attempt to describe how the autoresearcher should search [1].
- **Design for async review**: the stable loop across Osmani, Theo, and Copilot-style tooling is isolated workspaces/worktrees, task-state UIs, background execution, and attention routing so humans only re-enter when an agent actually needs them [2, 3].

  "specify intent → delegate → observe → review diffs → merge" [2]

- **Use model progress to change product process**: @_catwu's team now plans in short sprints, builds demos/evals instead of docs, revisits "too hard" features after each model release, and removes scaffolding once new models make it unnecessary. Also: keep agentic systems as simple as possible because failures compound with complexity [11, 12, 13, 14].
- **Make self-checks cheap**: Dreamer's coding loop does plan → build → test → fix, and David Singleton says TypeScript works especially well because compile-time errors give the agent loop immediate feedback on mistakes [15, 16]. Theo's Kernel demo shows the same philosophy on browser auth: one cloud-browser sign-in flow, including 2FA, can then be reused across agent instances for private GitHub access [3].

PEOPLE TO WATCH

- **Andrej Karpathy** — still the highest-signal operator feed in public: near-total delegation on real repos, strong views on memory/personality, and zero sugarcoating when code quality is bad [1, 10].
- **Addy Osmani** — best current synthesis of the orchestration shift, because it's grounded in the actual tools he uses daily instead of a generic future-of-IDEs take [2, 7].
- **Theo** — worth tracking for honest task-level comparisons and for pushing the "bigger IDE" framing from complaint into product experiments like T3 Code and Kernel demos [8, 9, 3].
- **@_catwu** — useful if your bottleneck is deciding what to ship in a world where model capability changes every release cycle [17, 11, 13].

WATCH & LISTEN

- **4:03-4:54 — Karpathy on 10-agent macro-actions.** Best quick mental model for parallel feature delegation across multiple repos [1].

Andrej Karpathy on Code Agents, AutoResearch, and the Loopy Era of AI (4:03)

- **16:35-19:18 — Karpathy on AutoResearch.** Watch this if you want the cleanest explanation of objective-metric loops and why the human becomes the bottleneck [1].

Andrej Karpathy on Code Agents, AutoResearch, and the Loopy Era of AI (16:34)

- **2:20-2:56 — Addy on orchestration as the skill to learn.** Fast distillation of the move from one-agent chats to fleets, coordination, and context handoff [7].

How to Beat AI FOMO and Stay Ahead in 2026 with Addy Osmani and Tim O'Reilly (2:20)

- **12:38-15:14** — **Theo's bigger-IDE thesis.** Good segment if your workflow collapses the moment you run multiple agents across multiple projects [3].

Andrej Karpathy
@karpathy

we're going to need a bigger IDE

9:22 AM · Mar 11, 2026 · **2.3M** Views

793 1.1K

Everything needs to change (12:38)

PROJECTS & REPOS

- **OpenClaw / ClawHub** — maintainer @magicseth says ClawHub now supports **1M weekly active users** on Convex; Peter Steinberger says the next push is making plugins great. Notable adoption signal for an agent platform [18, 19].
- **lat.md** — early agent integration for keeping spec files synced with implementation. Armin Ronacher finds it interesting, but explicitly wants proof on larger codebases before getting excited [20].
- **Arena + EVO Skill** — Sentient's new open competition for agent harnesses is using **Office QA** as its benchmark and aims to generate open feedback/data about where open harnesses still lag Claude Code. EVO Skill generates multiple candidate skills from eval feedback and keeps the best [21].
- **Dreamer** — not open source, but a project worth watching because its build loop is unusually explicit: Sidekick plans tools/data, builds, tests, exposes code/prompt internals, and exports via SDK/CLI. The platform also pays tool builders by usage and has a **$10k** prize for the best tool added by mid-April [15].

Editorial take: today's real edge wasn't a new chatbot tab — it was running more work in parallel, with cleaner isolation, explicit success metrics, and a higher *skepticism level about agent-written code [1, 2, 10].*

Sources

1. Andrej Karpathy on Code Agents, AutoResearch, and the Loopy Era of AI
2. Death of the IDE?
3. Everything needs to change
4. X post by @amanrsanger
5. X post by @Kimi_Moonshot
6. X post by @leerob
7. How to Beat AI FOMO and Stay Ahead in 2026 with Addy Osmani and Tim O'Reilly
8. X post by @theo
9. X post by @theo
10. X post by @karpathy
11. X post by @_catwu
12. X post by @_catwu
13. X post by @_catwu
14. X post by @_catwu
15. Dreamer: the Personal Agent OS — David Singleton
16. Dreamer: the Agent OS for Everyone — David Singleton
17. X post by @_catwu
18. X post by @magicseth
19. X post by @steipete
20. X post by @mitsuhiko
21. The Future Live | 03.20.26 | Guests from MOTS Podcast, Microsoft, Eliza Labs, and Sentient!