After "coding is solved": plan-first, parallel-agent ops, and sandboxing become the workflow

Coding Agents Alpha Tracker

2026-02-20

After "coding is solved": plan-first, parallel-agent ops, and sandboxing become the workflow

By Coding Agents Alpha Tracker • February 20, 2026

Boris Cherny's strongest claim yet: coding (for his work) is "largely solved," and the real frontier is end-to-end agentic ops—backed by +200% PR productivity and Claude reviewing 100% of PRs. Plus: Cursor's cross-OS agent sandboxing, Claude Code perf/regression signals, and new lightweight OpenClaw clones worth cloning.

TOP SIGNAL

Boris Cherny (Head of Claude Code) is blunt: for the kinds of programming he does, **"coding is largely solved"**, and the frontier is shifting to **adjacent, end-to-end agentic work** (project management, paying tickets, general ops) rather than better IDE autocomplete [1]. In that world, throughput isn't hypothetical: he says Anthropic saw **+200% productivity per engineer (PRs)** [1], and Claude now **reviews 100% of pull requests** (with human review still in the loop) [1].

TOOLS & MODELS

- **Claude Code — stability + performance signals**
  - **v2.1.47**: long-running sessions use **less memory** [2].
  - Team guidance: **keep reporting issues** and they'll fix them [2].
  - Practitioner complaint: Theo reports Claude Code has "**regressed an absurd amount**" with UI/feedback issues (timestamps not updating, missing "thinking," multi-minute hangs with 0 output) [3] and suggests it "needs to be **rewritten from scratch**" [4].
- **Cursor — agent sandboxing shipped across desktop OSes**

  - Cursor says it rolled out **agent sandboxing** on **macOS, Linux, and Windows** over the last three months [5].
  - Mechanism: agents run freely inside a sandbox, only requesting approval when they need to step outside it [5].
  - Implementation write-up: http://cursor.com/blog/agent-sandboxing [5].
- **OpenAI Codex — pricing/availability + compute pressure**
  - @thsottiaux: **Codex is included with a ChatGPT subscription** (even Plus has "very generous" usage) [6]; they attribute this to **gpt-5.3-codex** achieving "SoTA at lower cost" [6].
  - Same source: candidates increasingly ask how much **dedicated inference compute** they'll have, and usage/user is growing faster than user count → compute could be **scarce** [7].
- **Gemini 3.1 Pro — dev-workflow positioning (ramping up)**
  - Google Antigravity: **Gemini 3.1 Pro** is ramping to Google AI Ultra/Pro users, pitched around "advanced reasoning" and "long horizon planning" for dev workflows [8]. Details: https://antigravity.google/blog/gemini-3-1-pro-in-google-antigravity [9].
- **GitHub Copilot → Zed editor (GA)**
  - GitHub: Copilot subscription support in **Zed** is generally available [10]. Changelog: https://github.blog/changelog/2026-02-19-github-copilot-support-in-zed-generally-available/ [10].
- **Model choice drift + self-hosting pressure (reported trend)**
  - Salvatore Sanfilippo says he's seeing excellent programmers **move off US models (Codex, Claude Code)** toward **Chinese open-weight models** like **Kimi 2.5** and **GLM5** [11], often via providers or by building **in-house Nvidia GPU inference** to avoid outages and keep sensitive data internal [11].
  - He frames DeepSeek v4 as a potentially major moment *if* it lands as SOTA (as rumors suggest), putting pressure on OpenAI/Anthropic business sustainability [11].

WORKFLOWS & TRICKS

- **"Plan mode → execute" as a default loop (Claude Code / Boris Cherny)**
  1. Start the task in **plan mode** (he says he does this for ~80% of tasks) [1].
  2. Iterate on the plan (model goes back-and-forth) [1].
  3. Once the plan is good, **let it execute**; he'll **auto-accept edits** after that [1].
  - Implementation detail: plan mode is literally a prompt injection: "please don't write any code yet" [1].
- **Parallel agents, but treat "state" as a first-class problem**
  - Cherny: he runs ~**5 agents in parallel** while working (termi-

nal/desktop/iOS) [1] and highlights you can run many sessions in parallel [1].
  - Kent C. Dodds: similar "utter chaos" workflow—multiple projects, "a couple cloud agents" each, plus a locally guided agent [12].
  - Failure mode (real): Simon Willison describes "**parallel agent psychosis**"—losing track of where a feature lives across branches/worktrees/instances [13].
  - Recovery trick: after hacking in `/tmp` and crashing, he recovered the code from **`~/.claude/projects/` session logs**, and Claude Code could extract and recreate the missing feature [14].
- **Turn your feedback firehose into PRs (fast iteration loop)**
  - Cherny's pattern: point Quad/Cowork at an internal Slack feedback thread; it proposes changes and opens PRs quickly, which encourages more feedback because users feel heard [1].
  - Bug-fix loop: "as long as the description is good," he can fix a bug in minutes by delegating to Claude [1].
- **Token policy as a productivity lever (especially early)**
  - Cherny recommends giving engineers **as many tokens as possible** early (even "unlimited tokens" as a perk) so they try ideas that would otherwise feel too expensive; optimize/cost-cut after an idea works [1].
- **Avoid over-orchestration: tools + goal > rigid workflows (model-first design principle)**
  - Cherny: don't "box the model in" with strict step-by-step workflows; give it tools + a goal and let it figure it out—he argues heavy scaffolding mattered a year ago but often isn't necessary now [1].
- **"Ephemeral app" mindset + AI-native interfaces (Karpathy)**
  - Karpathy built a one-off cardio experiment dashboard with Claude; it had to **reverse engineer** a treadmill cloud API, process/debug data, and build a web UI; he still had to chase bugs (units, calendar alignment) [15].
  - His takeaway: the app-store model feels outdated for long-tail needs; instead, the industry needs **AI-native sensors/actuators** with agent-friendly APIs/CLIs so agents don't have to click HTML UIs or reverse engineer services [15].
- **Agent "memory" ops in practice (LangSmith Agent Builder)**
  - LangChain's concrete guidance:
    - Tell your agent to **remember what works** [16]
    - Use **skills** to inject specialized context when needed [16]
    - Edit agent **instructions directly** when it's faster [16]
  - Entry point: https://blog.langchain.com/how-to-use-memory-in-agent-builder/?utm_medium=social&utm_source=twitter&utm_campaign=q1-2026_ab-philosophy_aw [16].

PEOPLE TO WATCH

- **Boris Cherny** — production-grade Claude Code habits (plan mode, parallel sessions) + strong claims about where "after coding" goes [1].
- **Andrej Karpathy** — high-signal framing: *ephemeral bespoke apps* + "AI-native CLI/API" requirements for tools and hardware vendors [15].
- **Simon Willison** — the best micro-case study of parallel-agent failure/recovery using session logs as the source of truth [13, 14].
- **Steve Ruiz (tldraw)** — pragmatic company-building: code gets easier, but alignment/positioning/communication get harder—and he's automating the overhead away [17].
- **Theo** — sharp practitioner critique on Claude Code regressions plus continued pressure on "harness vs infra" policy differences across vendors [3, 4].
- **François Chollet** — frames agentic coding as ML optimization (spec/tests as constraints) and asks what the "Keras of agentic coding" will be [18]; **@swyx** suggests **DSPy** as the presumptive community default [19].

WATCH & LISTEN

1) Boris Cherny — "Plan mode" as the default starter move (~1:09:52–1:10:41)

Hook: a simple, copyable workflow: force planning first (no code), iterate the plan, then execute + auto-accept when the plan is solid [1].

Head of Claude Code: What happens after coding is solved | Boris Cherny (69:51)

2) Boris Cherny — “Coding is largely solved... what’s next?” (~0:18:19–0:19:06)

Hook: his thesis on why the frontier is shifting from IDE coding to adjacent operational tasks and general automation [1].

Head of Claude Code: What happens after coding is solved | Boris Cherny (18:19)

3) Steve Ruiz — daily automated release notes from landed PRs (~0:20:35–0:21:02)

Hook: treat agents like scheduled staff: every day, Claude scans the last 24h PRs and drafts “release notes we’d publish if we shipped main today” [17].

Selling SDKs in the era of many Claudes | Steve Ruiz from @tldraw (20:35)

PROJECTS & REPOS

- **NanoClaw** — "Clawdbot" in ~500–700 LOC TypeScript using **Apple container isolation** for sandboxing/security; posted as Show HN [20, 21]. Repo: https://github.com/gavrielc/nanoclaw [20] • HN: https://news.ycombinator.com/item?id=46850205 [20].
- **Nullclaw** — "fastest, smallest OpenClaw clone": **678 KB static binary**, no runtime/VM/framework overhead [22]. Repo: https://github.com/nullclaw/nullclaw [22].
- **tldraw agent starter kit** — Cursor-like agent panel next to a canvas; cloneable starter for agent+canvas UX: https://tldraw.dev/starter-kits/agent [17].

Editorial take: As agents make code cheap, the new edge is **orchestration discipline**: plan-first loops, sandboxing, session-log recoverability, and AI-native interfaces that don't force your agent to "be the computer." [1, 5, 14, 15]

Sources

1. Head of Claude Code: What happens after coding is solved | Boris Cherny
2. X post by @jarredsumner
3. X post by @theo
4. X post by @theo
5. X post by @cursor_ai
6. X post by @thsottiaux
7. X post by @thsottiaux
8. X post by @antigravity
9. X post by @antigravity
10. X post by @github
11. Deepseek v4 potrebbe essere un colpo al cuore alla sostenibilità dell'AI americana
12. X post by @kentcdodds
13. X post by @simonw
14. X post by @simonw
15. X post by @karpathy
16. X post by @LangChain
17. Selling SDKs in the era of many Claudes | Steve Ruiz from @tldraw
18. X post by @fchollet
19. X post by @swyx
20. X post by @betterhn20
21. X post by @swyx
22. X post by @jedisct1