Coding Agents Crush Prototypes While Human Finishing Becomes the Bottleneck

Coding Agents Alpha Tracker

2026-04-02

Coding Agents Crush Prototypes While Human Finishing Becomes the Bottleneck

By Coding Agents Alpha Tracker • April 2, 2026

Today's strongest signal is a process change, not a model drop: teams are using coding agents to generate working specs and fast prototypes, then handing them to humans for architecture, ergonomics, and final integration. Also inside: Open SWE, LangSmith Sandboxes, Claude Code's no-flicker renderer, AI Studio workflow hacks, and a practical vuln-hunting loop.

TOP SIGNAL

The clearest production pattern today comes from DHH's Basecamp 5 work at 37signals: coding agents radically compress *exploration*, not *finishing*. He says the cost of getting from 0→75% is now 100-1000x lower, but the 75→100% backlog can actually get worse—so the team is using AI to create working specs and prototypes, then handing those to programmers for the PR-quality, production-ready version [1].

> "agents don't finish beautiful, ergonomic, desirable software… that human finishing at the end is not just necessary, it's essential." [1]

TOOLS & MODELS

- **Open SWE** — LangChain's new open-source framework for internal coding agents packages the architecture Stripe, Ramp, and Coinbase independently landed on: isolated cloud sandboxes, curated tools, subagent orchestration, and tight workflow integration. You can fork and deploy it yourself [2].
- **LangSmith Sandboxes** — private preview for locked-down, temporary code-execution environments with granular access and resource controls.

Practical if your agent needs to run code without getting full machine access [2].

- **LangGraph v1.1 + Deploy CLI** — v1.1 adds type-safe streaming/invoke plus Pydantic and dataclass coercion, and the new CLI does one-step terminal deploys to LangSmith Deployment [2].
- **deepagents v0.5.0 alpha** — async subagents and multimodal support landed. If you're building multi-worker agent setups, that's the release to note [2].
- **Claude Code NO_FLICKER** — enable with `CLAUDE_CODE_NO_FLICKER=1 claude`. Boris Cherny says the new renderer removes flicker/jumping, holds CPU and memory steady as conversations grow, supports mouse interaction, and cleans up text selection; tradeoffs are new search/copy behavior and scrolling that still needs tuning [3, 4, 5]. Docs: fullscreen mode [6].
- **AI Studio vs Anti Gravity** — Logan Kilpatrick frames AI Studio as the batteries-included web path for fast sharing and model hookup, with Anti Gravity as the IDE-first step up when you want deeper code control. AI Studio also auto-routes to models suited to the task, like Lyria for music and live models for audio, unless you override it in the prompt [7].
- **Claude Opus 4.5, in one real repo** — Theo says it added Clerk auth to his T3 Canvas web+mobile monorepo on the first try, including synced experiences, user account UI, and organizations [8].

WORKFLOWS & TRICKS

- **Basecamp's prototype → production loop**
  1. Let designers or junior devs push prototypes much further with agents [1].
  2. Use the agent's endless patience to iterate until you know what you actually want [1].
  3. Hand the working AI-coded prototype/spec to programmers [1].
  4. Expect the human finishing pass—architecture, ergonomics, and durability—to be the hard part [1].
- **AI Studio screenshot-to-code**
  1. Start ambitious; Logan says the model and infra are now good enough to break down more complex asks than people assume [7].
  2. Drop in a screenshot and prompt `copy this UI exactly` [7].
  3. For higher fidelity, use the Chrome extension to capture the full page/DOM [7].
  4. Ask for multiple treatments or feature variants, then move the winning output into the real codebase [7].
  5. For edits, use **Annotate** to draw/request changes or **Focus** to directly change selected elements [7].
  6. Same pattern also works for quick dashboard apps from raw CSVs or even table screenshots [7].
- **Remix shared apps without charging the original builder**

1. Open the shared AI Studio link [7].
2. Attach your own API key [7].
3. Hit **Remix** to apply suggestions or new prompts; the remixed version uses the recipient's key, not the sender's [7].

- **Skip plan mode if you just ignore the plan**
  - Peter Steinberger says he never uses Codex plan mode: `just talk with your agent`. kr0der's reason is practical—plan mode kept generating giant plans he didn't read, while direct implementation discussion felt easier [9, 10].
- **Phone → laptop session handoff**
  - Use Claude mobile's Code tab from the iOS/Android app, start a session on the go, then teleport it into the local CLI later. Anthropic engineers say the same flow also works for quick code changes directly from the iPhone app [11, 12].
- **A clean agentic security loop**
  1. Run the same Claude prompt across each source file in a repo [13].
  2. Write each finding to `Vuln.md` [13].
  3. Verify exploitability in a second pass [13].
  4. Randomize the starting file to add stochastic coverage, and keep the task narrow so the model gets more context per file [13].
  5. Use this where success is objectively testable; the agent can keep iterating on success/failure without getting bored [13].

PEOPLE TO WATCH

- **DHH** — one of the best current firsthand accounts of what changes inside a real product org when agents get good: design/prototyping accelerates, but senior review becomes the bottleneck [1].
- **Logan Kilpatrick + Amar Reshi** — high signal because they're not just demoing AI Studio; they're using screenshot cloning, remix, and CSV-to-dashboard workflows on their own product work [7].
- **Boris Cherny** — worth following for concrete Claude Code ergonomics work, especially when he explains renderer tradeoffs instead of shipping a black-box UI tweak [3, 14, 15, 6].
- **Armin Ronacher** — useful two-sided signal: Clanker helped generate migration docs for a real OSS release, but it still hallucinated badly on a trait-bound discussion. That's the right kind of grounded maintainer feedback [16, 17, 18].
- **Salvatore Sanfilippo** — worth watching if you care about security-oriented agent loops and where `search forever until the exploit works` beats normal human attention spans [13].

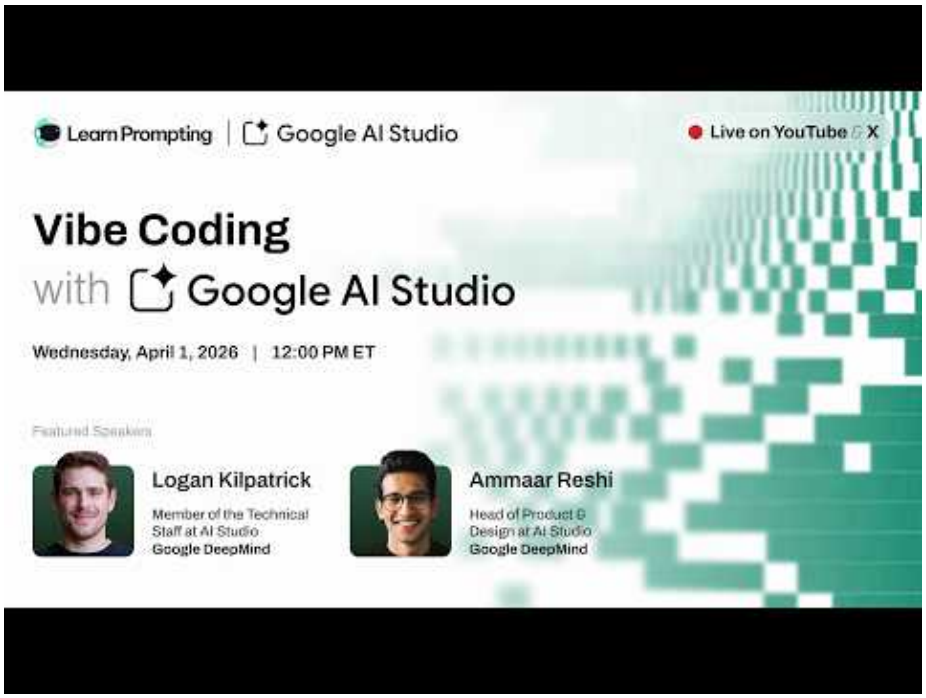
WATCH & LISTEN

- **8:21–9:42 — DHH on the new waterfall.** Best short segment today on the practical split: use AI to co-create a working feature spec, then

hand that prototype to programmers for production-ready code [1].

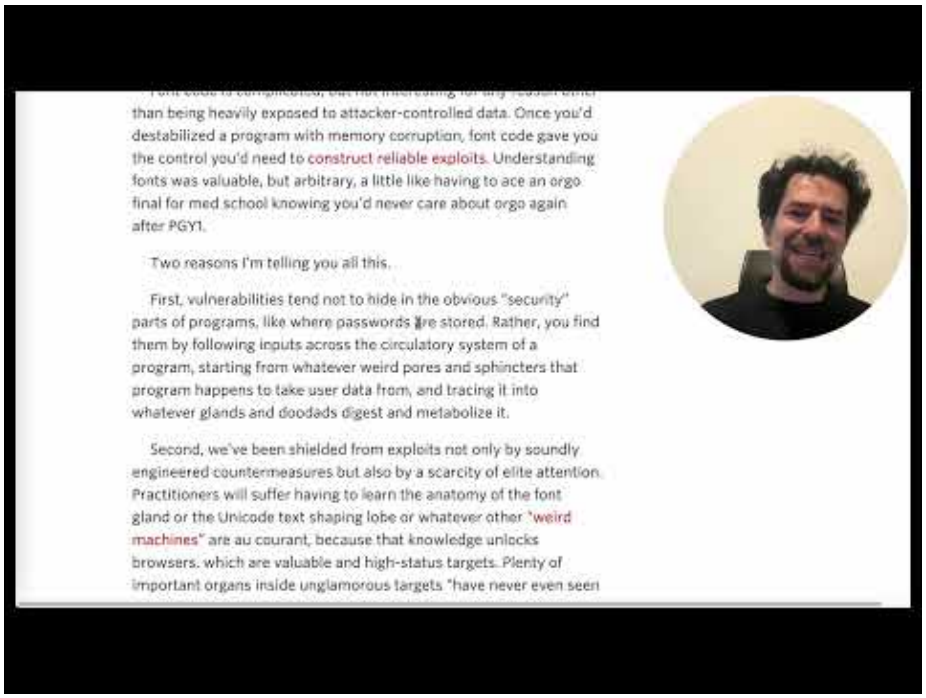
Pencils down – REWORK (8:21)

- **44:18–45:39 — Logan Kilpatrick on screenshot-first product work.** He shows the literal prompt pattern—drop in a screenshot, say `copy this UI exactly`, ask for variants, then move the winning build into the real AI Studio codebase [7].



Vibe Coding with Google AI Studio (44:18)

- **20:25–21:42 — Salvatore on the vuln-report loop.** A clean demo of an agent workflow with objective feedback: same prompt per file, write reports, then verify exploitability [13].



La sicurezza informatica è bollita? (20:25)

PROJECTS & REPOS

- **Open SWE** — new open-source framework for internal coding agents. The signal here is not stars; it's that LangChain says Stripe, Ramp, and Coinbase independently converged on the same architecture it now packages [2].
- **deepagents v0.5.0 alpha** — notable open agent-orchestration update: async subagents plus multimodal support [2].
- **LangGraph v1.1** — type-safe streaming/invoke, Pydantic/dataclass coercion, and a new Deploy CLI make this a meaningful release for teams operationalizing agents [2].
- **similar** — Armin Ronacher shipped a major update to his Rust diffing library with histogram/hunt diffs, semantic inline-diff fixes, `no_std` support, and performance work; Clanker generated the migration docs in UPGRADING.md [16, 17].

Editorial take: the best teams are not asking agents to finish the feature—they're using them to make exploration, prototyping, and bounded search absurdly cheap while keeping humans responsible for taste, verification, and final integration [1, 7, 13].

Sources

1. Pencils down – REWORK
2. March 2026: LangChain Newsletter
3. X post by @bcherny
4. X post by @bcherny
5. X post by @bcherny
6. X post by @bcherny
7. Vibe Coding with Google AI Studio
8. I got DMCA'd by Anthropic (not a joke)
9. X post by @steipete
10. X post by @kr0der
11. X post by @_catwu
12. X post by @bcherny
13. La sicurezza informatica è bollita?
14. X post by @bcherny
15. X post by @bcherny
16. X post by @mitsuhiko
17. X post by @mitsuhiko
18. X post by @mitsuhiko