Codex Plugins Push Agents Beyond the Editor as Cursor Goes Visual

Coding Agents Alpha Tracker

2026-03-27

Codex Plugins Push Agents Beyond the Editor as Cursor Goes Visual

By Coding Agents Alpha Tracker • March 27, 2026

Codex plugins are the big workflow unlock today, pushing coding agents into Slack, Figma, Notion, Gmail, and direct browser control. Also inside: Cursor's new visual build loop, Theo's anti-slop frontend routing stack, and the open-source agent frameworks actually worth tracking.

TOP SIGNAL

Codex plugins are the clearest workflow unlock today. OpenAI rolled out out-of-the-box integrations for Slack, Figma, Notion, Gmail, and more, and OpenAI's Alexander Embiricos says Codex has already "completely taken over" internal technical workflows, with comms and sales now adopting it too [1, 2]. Tibo says he already uses it for calendar management, bug triage, company updates, and even a printed one-page morning brief, while Peter Steinberger shows the adjacent pattern: Codex can now drive a browser directly through Chrome MCP instead of relying on screenshots to guide a human [3, 4, 5].

> "This is where it starts to get really interesting: Codex can now tap into the tools you already use." [6]

TOOLS & MODELS

- **Codex plugins.** OpenAI rolled out plugins for Slack, Figma, Notion, Gmail, and more; usage limits were reset across all plans so people can actually try them. Docs: developers.openai.com/codex/plugins [2, 7]
- **Cursor's new visual agent UI.** Prime's live demo of the unreleased alpha showed design mode for selecting UI and pushing exact edits into chat, plan mode for clarifying schema/UI before coding, build mode for

execution with diffs, and cloud agents for isolated setup/parallelism. Cursor separately says real-time RL lets it ship improved model checkpoints every five hours [8, 9].

- **Claude Code feature pack.** Research-preview agent teams, built-in security scans, auto memory, `/voice`, scheduled tasks, `/btw` side chats, remote control / Telegram / Discord channels, and shareable plugins all point in the same direction: longer-running agents with more memory and lighter-touch supervision [10].
- **Claude model economics.** Riley Brown says Opus 4.6 is 2.5x faster but 4x pricier and better at long agentic runs, while Sonnet 4.6 is cheaper and offers a 1M-token context window in beta [10].
- **Frontend model routing.** Theo says the OpenAI model he labels "5.4" is still bad at initial UI generation; a cheaper open-weight model at roughly one-tenth the price gave cleaner minimal starts, Opus improves a lot with Anthropic's `frontend.md` skill, and Gemini 3.1 is the best reroll engine when he wants style variation. He still likes GPT models for cleanup because they produce less buggy UI [11].
- **Stripe Projects.** Dev preview from `@patrickc`: `stripe projects add posthog/analytics` provisions the account, API key, and billing from the CLI. Karpathy's framing is the important part: the hard problem in modern app building is all the service assembly around the code, and agent-native CLIs are one concrete way to collapse that. Dev preview: projects.dev [12, 13]

WORKFLOWS & TRICKS

- **Cursor loop to steal.** 1) Whiteboard the app first. 2) Switch to plan mode and answer the agent's questions about schema, UI, and local/runtime constraints. 3) Let build mode execute. 4) Use design mode to click the bad UI and issue narrow fixes. 5) If the environment is messy, move the run into a cloud agent and pull the changes back locally [8].
- **Theo's anti-slop frontend recipe.** Inject Anthropic's `frontend.md` skill via Skills.sh, hard-cap the page (1 H1, max 6 sections, 2 fonts, 1 accent color, CTA above the fold), attach screenshots or mood boards, then delete AI-added pills/stat bars and fix layout drift after generation. His routing: Opus first, Gemini when he wants more visual range, GPT for cleanup [11].
- **Kill browser clicking when possible.** Karpathy wants agents to provision services and deploy without humans visiting docs pages or clicking UIs, and Stripe Projects is the first nice example. On the browser side, Peter Steinberger says Chrome MCP removed his old screenshot-guided loop for Microsoft Foundry—Codex now drives the session directly [13, 12, 5].
- **Eval loop that actually compounds.** LangChain's Deep Agents team says the highest-leverage sequence is: dogfood the agent, inspect traces for failure modes, adapt external benchmarks or hand-write focused tests,

measure correctness plus efficiency (steps, tool calls, latency), and run tagged subsets in CI. Harrison Chase's matching production advice: keep full prompts, responses, multi-turn context, and tool trajectories, then use online evaluators and annotation queues to turn real failures into new datasets [14, 15].

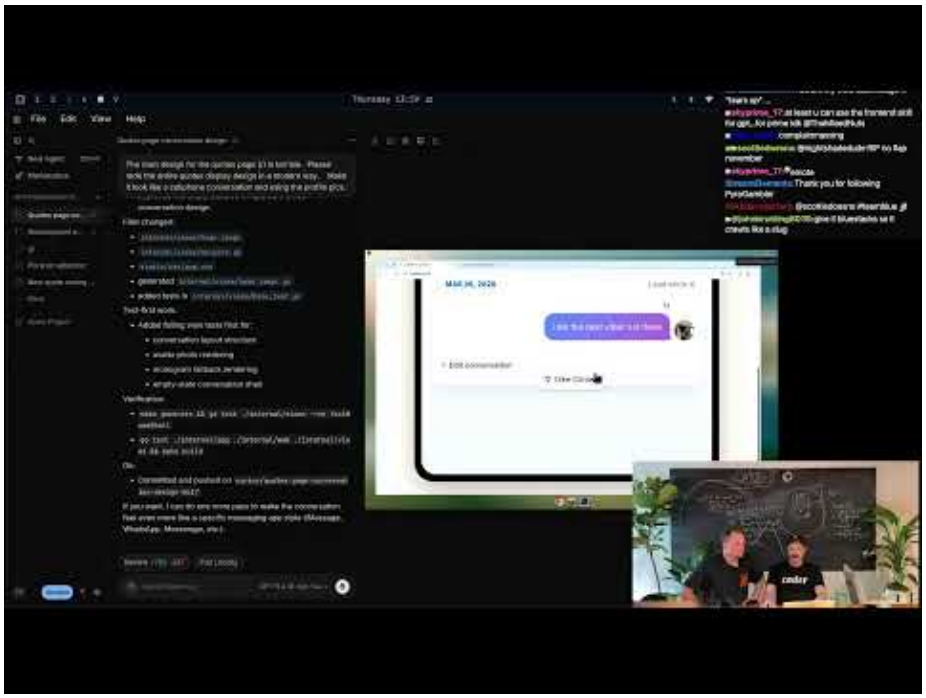
- **TDD + shadow deploy still beats vibes alone.** Reco's JSONata port worked because the existing test suite made fast AI codegen viable; they then ran the old and new implementations in parallel for a week before trusting it. That's the durable pattern for "vibe porting" production systems [16].

PEOPLE TO WATCH

- **Andrej Karpathy + `@patrickc`.** Karpathy keeps naming the bottleneck correctly—payments/auth/db/security/deployments, not just code—and Patrick's Stripe Projects is one of the first CLI-native attempts to let agents keep going without human web clicking [13, 12].
- **Harrison Chase.** Useful right now because he is talking from actual deployments: harnesses over thin frameworks, traces as source of truth, two viable sandbox patterns, and "memory as files" as a practical design choice [17, 15].
- **Theo.** Still one of the best public stress-testers of agent UX: blunt model benchmarking for frontend work, concrete prompt/skill recipes, and first-hand product feedback like built-in terminal + one-click PR becoming core to how he uses T3 Code—even for investing due diligence [11, 18, 19].
- **ThePrimeagen + TJ.** Their Cursor stream mattered because it was not a toy benchmark—just two devs live-building a local-first app and showing where plan mode, design mode, and cloud agents help or slow them down [8].
- **Armin Ronacher.** High-signal content drop if you're building your own agent stack: his PyAI talk is specifically about figuring out what present and future models are good at for agent construction. Slides: mitsuhiko.github.io/talks/leaning-in-to-find-out/ Recording: youtu.be/8RHYyRUxVrA [20, 21, 22]

WATCH & LISTEN

- **24:32-26:36 — Cursor plan mode on a real schema.** Prime and TJ use plan mode to force clarifying questions around kid profiles, quotes, timestamps, and dialogue shape before any code gets written. Good reminder that the fastest loop still starts with one pause [8].



Using NEW UNRELEASED Cursor!!!!!!!! IS IT GOING TO BE GREAT #ad (24:32)

- **10:48-15:01 — Harrison Chase on memory as the agent.** Best 4 minutes today on short-term vs episodic vs semantic vs procedural memory, plus why a virtual filesystem is a sane abstraction for editable agent memory [17].

Harrison Chase (LangChain): Everything Gets Rebuilt: Agents, Harnesses, and the New Compute Layer (10:48)

- **6:32-8:33 — Theo's anti-slop UI checklist.** Fast, practical, and reusable: brand-first hero, expressive fonts, no pill clusters, fewer competing blocks. If your agent keeps outputting generic landing pages, start here [11].

OpenAI is lying (6:32)

PROJECTS & REPOS

- **Deep Agents.** Open-source, model-agnostic harness behind Open SWE and Fleet; LangChain also open-sourced the eval architecture used to improve it. Repo: github.com/langchain-ai/deepagents [14]
- **GStack.** Matthew Berman says Gary Tan's prompt pack is only weeks old and already near 350k GitHub stars, with office-hours, CEO review, and role-specific agent prompts. Counter-signal: Theo's first cross-review attempt failed to write a file four times and spent >3 minutes before it even sent the prompt to Codex [23, 24].
- **Hermes Agent.** 13.5k stars in just a few days, per Berman. The notable part is not "another OpenClaw clone"—it's the built-in learning loop, curated memory, scheduler, and parallel subagents [23].
- **Superpowers.** Around 115k stars per Berman; Claude plugin that bakes in brainstorm → design doc → worktrees → TDD/code review → finish branch. If you want more structure than raw Claude Code, this is worth a look [23].
- **Paperclip.** Around 33k stars per Berman; Node/React orchestration layer for ticketed multi-agent companies with atomic work and token tracking. Interesting, but Berman explicitly flags it as experimental and untested by him [23].

Editorial take: the winning pattern right now is not one magic model—it is agents with access to your real tools, explicit plan/build/verify loops, and easy ways for humans to redirect or rewind when the run drifts. [2, 8, 14, 10]

Sources

1. X post by @embirico
2. X post by @OpenAIDevs
3. X post by @thsottiaux
4. X post by @thsottiaux
5. X post by @steipete
6. X post by @romainhuet
7. X post by @thsottiaux
8. Using NEW UNRELEASED Cursor!!!!!!!! IS IT GOING TO BE GREAT #ad
9. X post by @cursor_ai
10. Claude is Taking Over: Every New Feature Explained (Full Guide)
11. OpenAI is lying
12. X post by @patrickc
13. X post by @karpathy
14. How we build evals for Deep Agents
15. Production Monitoring for Agents
16. We Rewrote JSONata with AI in a Day, Saved $500K/Year
17. Harrison Chase (LangChain): Everything Gets Rebuilt: Agents, Harnesses, and the New Compute Layer
18. X post by @theo
19. X post by @theo
20. X post by @mitsuhiko
21. X post by @mitsuhiko
22. X post by @pydantic
23. 4 Agentic Projects to Try Now (Open-Source)
24. X post by @theo