Anthropic's Internal Claude Code Playbook and GPT-5.5's First Real Trials

Coding Agents Alpha Tracker

2026-04-24

Anthropic's Internal Claude Code Playbook and GPT-5.5's First Real Trials

By Coding Agents Alpha Tracker • April 24, 2026

Anthropic's Cat Wu dropped the clearest operating model of the day: how Claude Code and Cowork actually fit into shipping, PMM, sales, and applied AI work. This brief also covers GPT-5.5/Codex field reports, Simon Willison's plan-first build loop, production agent architecture patterns, and the repos worth trying.

TOP SIGNAL

Anthropic's internal Claude Code/Cowork playbook is the highest-signal drop today. Cat Wu says AI has pulled many Claude Code feature timelines from six months to one month, one week, or even one day, and that the team ships almost all Claude Code features first as **Research Preview** to lower commitment and get feedback fast. The interesting part is the operating system around it: weekly metrics readouts, explicit team principles, engineers dropping ready features into a launch room, and docs/PMM turning those into launches the next day; PRDs are mostly reserved for infra-heavy or ambiguous work. [1]

> "We want to remove every single barrier to shipping things." [1]

TOOLS & MODELS

- **OpenAI GPT-5.5** — live in Codex and ChatGPT today; API is coming soon. Reported evals: **82.7% Terminal-Bench 2.0**, **73.1% Expert-SWE**, **58.6% SWE-Bench Pro**, **78.7% OSWorld-Verified**. Codex gets **400K context**, with **1M** in the API. [2, 3]
- **Codex app update** — full browser use, computer use, in-app docs/PDF viewer, non-dev mode, global dictation, and a new **auto-review** mode. Auto-review uses a guardian agent to vet higher-risk actions so Codex can

keep running tests, builds, long tasks, and automations with fewer manual approvals; Alexander Embiricos says it is now his default mode. [4, 5, 6, 7, 8]

- **Early GPT-5.5 read from actual users** — Aaron Friel says the new Codex harness caused a "tidal wave" of PRs and has engineers running single tasks for **40+ hours**; Will Koh says it handles ambiguous tasks with less prompting, finds the right code paths, and uses DB + telemetry tools in novel ways inside Ramp's Inspect harness. On frontend, Tylernotfound and Thibault Sottiaux both say 5.5 is the first OpenAI model that feels like a real programming partner. [9, 10, 11, 12]
- **But the caveats are real** — Theo calls GPT-5.5 the best code-writing model he has used, but also says it needs stricter instructions, explores when under-specified, is hard to steer back once it goes off track, and is expensive at **$5/M input** and **$30/M output**. Matthew Berman and Riley Brown both argue the right evaluation metric is task quality + time + total tokens/cost, not price-per-token alone. [13, 14, 15, 16]
- **Claude Code quality fix** — Anthropic says the recent quality regressions were **three harness bugs**, fixed in **v2.1.116+**, with subscriber usage limits reset. The standout bug repeatedly cleared older thinking after sessions sat idle for more than an hour, which made Claude look forgetful and repetitive; Simon Willison says "stale" sessions are a huge part of his real usage. [17, 18]
- **Support-ticket-to-PR automation is already showing up in the wild** — Jason Zhou says an AI agent read a support ticket, checked the database, found the root cause of a broken customer credits issue, and submitted the fix PR in **10 minutes** with **no human intervention**; he says the usual path takes about **two weeks** through sprint prioritization. [19]

WORKFLOWS & TRICKS

- **Choose the Anthropic surface by output type.** Cat Wu's split is clean: **CLI** for one-off coding and newest features; **Desktop** for frontend work, live preview, and a graphical control plane across sessions; **web/mobile** to dispatch work on the go; **Cowork** for non-code outputs like docs, inbox/slack cleanup, and decks. [1]
- **Steal Cat Wu's async deck workflow.** First connect the data sources relevant to your role (Slack, Gmail, Calendar, Drive). Then give Cowork the narrative, draft links, and constraints; ask it to propose an outline first; lock the outline; let it run for an hour or a few hours; then do the last-mile editing yourself—mainly trimming text and picking the final story. [1]
- **Build narrow internal apps for repetitive work, not demo-ware.** Anthropic's sales team used Claude Code to build a web app that pulls **Salesforce + Gong + customer notes** and spits out tailored decks in seconds instead of 20-30 minutes. Cat Wu's explicit advice: pick something you do constantly, push the last 5-10% to **100% reliability**, and

build apps you will actually use every day. [1]

- **Simon Willison's plan-first build loop is worth copying verbatim.** He started by probing the repo/problem in regular Claude chat, pasted the findings into `notes.md`, had Claude Code write `plan.md`, iterated the plan, then used `build it.` plus **Playwright red/green TDD**, queued prompts, and **small commits**. He ran `npx vite` for live preview, used a **separate Claude session** for GitHub Actions + Pages deploy, and finally used **GPT-5.5/Codex** as a second-model verification pass. [20]
- **For large unstructured datasets, hardcode the expensive parallel ops.** Listen models research data as a **table** (rows = responses, columns = extracted features), exposes classification as a hardcoded map-reduce tool, and can spawn ~**500 constrained subagents** with small models for quantitative passes. They keep a **sandboxed Python fallback** for the 20% long tail, and they split **live** vs. **async** runs: faster/smaller models with minimal thinking live, fuller scans with more thinking asynchronously. [21]
- **Improve agents from traces, not vibes.** Cat Wu asks the model to explain its own failures so the team can fix prompts or harnesses; she says even **10 good evals** can be enough to make progress visible. Listen goes deep on one or two traces at a time and runs a reviewer agent that checks reports for unsupported claims, while LangChain summarizes the broader loop as: traces -> evals/feedback -> reusable datasets -> iteration. [1, 21, 22]
- **Route reasoning effort by task.** Multiple testers are landing on the same pattern: use **low/medium** reasoning for most work, save **high/xhigh** for long builds, and compare models on **quality, tokens, time, and total cost** on the exact task you care about. [15, 23]

PEOPLE TO WATCH

- **Cat Wu** — rare operator-level walkthrough of how a frontier lab is actually using Claude Code and Cowork across PM, sales, applied AI prep, and launch operations. [1]
- **Simon Willison** — still the cleanest source for reproducible agent workflows: today he published the LiteParse build notes and a Codex-subscription plugin for `llm`, both with exact commands and verification steps. [24, 25]
- **Romain Huet** — useful if you want the shortest path from launch post to practitioner feedback; his GPT-5.5 posts pair official claims with concrete reactions from OpenAI and Ramp builders. [2, 26, 9, 10]
- **Logan Kilpatrick** — one of the few people talking concretely about how agent-written code gets into a production codebase: CI green, test pass, then handoff to engineers who steward the final merge and infrastructure. [27]
- **Theo** — high-signal skeptic on GPT-5.5: he calls it the best code-writing model he has used, while documenting the practical annoyances around

context contamination, steering, and cost. [13, 14, 28]

WATCH & LISTEN

- **36:08-40:50 — Cat Wu on Cowork as an async PMM assistant.** Best concrete non-code agent workflow today: connect the right sources, ask for an outline first, let it research launches and internal channels, then do the last-mile edit yourself. [1]

How Anthropic's product team moves faster than anyone else | Cat Wu (Head of Product, Claude Code) (36:08)

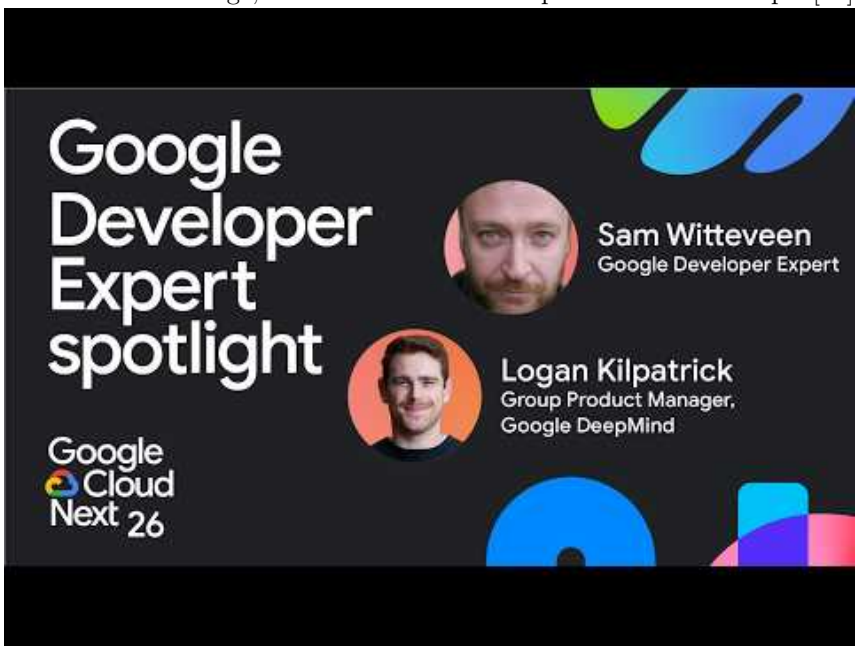
- **15:08-18:04 — Riley Brown stress-tests GPT-5.5 browser use by making it build a canvas app and draw a flowchart.** Good visual proof of spatial reasoning plus the new browser loop: build, inspect, click,

correct, repeat. [15]
GPT-5.5: My honest review (15:07)

- **06:55-08:03 — Florian Jungermann on the table + map-reduce pattern.** If you are building agents over lots of messy qualitative data, this is the clip: rows as responses, columns as extracted features, and one tool call fanning out to hundreds of constrained subagents. [21]

Why Listen believes generic AI agents fall short | Florian Juengermann, CTO (6:55)

- **12:03-13:49 — Logan Kilpatrick on getting agent-written code into production.** The important bit is not the coding demo; it is the handoff model: technical staff get the change green, engineering owns the final merge, and the same team improves the next loop. [27]



Vibe coding to production: Logan Kilpatrick on the evolution of AI Studio (12:03)

PROJECTS & REPOS

- `llm-openai-via-codex` — Simon Willison had Claude Code reverse-engineer `openai/codex` auth so his `llm` CLI can use a Codex subscription. Setup: install Codex CLI, `uv tool install llm`, `llm install llm-openai-via-codex`, then `llm -m openai-codex/gpt-5.5 '...'`; it also supports images, chat, logs, and tools. [25]
- `openai/codex` — the open-source CLI + app server behind the Codex ecosystem. OpenAI's Romain Huet says the point is to let ChatGPT subscriptions work in the app, terminal, JetBrains, Xcode, OpenCode, Pi, and even Claude Code. [29, 25]
- `run-llama/liteparse` + web demo — LiteParse is a fast, heuristics-based PDF parser that can be used from the CLI or inside a coding agent; Simon Willison used Claude Code to build a browser version in about an hour. [30, 31, 32]
- **Agentic red/green TDD guide** — Simon's red/green TDD write-up

is still one of the best concrete pattern docs to hand an engineer who is moving from "prompting" to repeatable agent workflows. [20]

Editorial take: the real edge today is not "which model won"—it is whether you can wrap a strong model in plans, evals, trace review, reviewer loops, and narrow internal apps that people use every day. [20, 1, 21]

Sources

1. How Anthropic's product team moves faster than anyone else | Cat Wu (Head of Product, Claude Code)
2. X post by @romainhuet
3. X post by @swyx
4. X post by @OpenAIDevs
5. X post by @thsottiaux
6. X post by @gdb
7. X post by @OpenAIDevs
8. X post by @embirico
9. First impressions of GPT-5.5 from Aaron Friel
10. First impressions of GPT-5.5 from Will Koh
11. X post by @tylernotfound
12. X post by @thsottiaux
13. X post by @theo
14. X post by @theo
15. GPT-5.5: My honest review
16. GPT-5.5 is HERE!
17. X post by @ClaudeDevs
18. An update on recent Claude Code quality reports
19. X post by @jasonzhou1993
20. Extract PDF text in your browser with LiteParse for the web
21. Why Listen believes generic AI agents fall short | Florian Juengermann, CTO
22. X post by @LangChain
23. OpenAI just dropped GPT-5.5… (WOAH)
24. GPT 5.5, ChatGPT Images 2.0, Qwen3.6-27B
25. A pelican for GPT-5.5 via the semi-official Codex backdoor API
26. X post by @romainhuet
27. Vibe coding to production: Logan Kilpatrick on the evolution of AI Studio
28. I don't really like GPT-5.5…
29. X post by @romainhuet
30. X post by @jerryjliu0
31. X post by @simonw
32. X post by @simonw