

Ask for the outcome, not the feature: Theo's prompt-first workflow, Anthropic's VCS rebuild for agent scale, and Daybreak in security loops

Coding Agents Alpha Tracker

2026-10-19

Ask for the outcome, not the feature: Theo's prompt-first workflow, Anthropic's VCS rebuild for agent scale, and Daybreak in security loops

By Coding Agents Alpha Tracker • October 19, 2026

Theo argues developers should get reports, visualizers and support flows by prompting their agents instead of building features. Anthropic replaced Git after agent output reached 30,000 merges a day, and Huntley and OpenAI describe defensive security loops built on Daybreak.

Stop building features you can ask for

In a new video, **Theo** argues that engineers under-use agents because they “think in code”. They plan features when they could simply ask their agent for the result [1]. He shows several concrete patterns:

- **Audit your own agent history.** A friend asked T3 Code for a feature that breaks down tool-call durations. Theo instead pasted the request into a thread (“break down all my tool calls... on this machine”), and the agent built a report page on its own. It found one home-directory search that ran for 41.7 minutes and another that ran almost 10 [1]. Earlier audits showed that most of Claude's tool-call failures on his machines came from ripgrep commands written for bash that broke under zsh [1].
- **Ask for a context-window visualizer.** Doing this showed that, because of how his proxy was set up, Railway had been adding 70,000 tokens of context to every request for two months [1]. Turn any request you keep repeating into a skill or a CLAUDE.md entry [1].
- **Describe the delegation in the prompt instead of waiting for a subagent picker.** His example: “spin up two opus 5.5 delegates to

explore performance wins in T3 code, summarize their thoughts, and then start a new thread using Grock 4.7 to implement your findings” [1].

- **Write support flows as a markdown playbook.** `npx t3 triage` collects basic machine facts and passes a prompt to Claude Code or Codex. The playbook tells the agent to ask what went wrong, get repro steps and screenshots, read the facts and the source, check whether the bug is already fixed upstream, strip secrets before filing, and comment on an existing issue rather than open a duplicate [1]. A feature he expected to take “thousands of lines” became five lines of code and about 30 lines of markdown [1].
- **Ask the agent what it can do.** He had the Codex desktop agent describe its computer-use capabilities, then pasted that description into Opus and asked it to “clone as much of this as you can”. He says that work is still in a PR [1].

His larger point is to spend engineering time on plumbing that agents can use. He connected email to ChatGPT, has Dot watch for an AWS bill and investigate the costs, and connected T3 Code’s MCP so Codex and Dot can start T3 Code cloud threads that ship changes [1].



Make Your Prompts Dumber (14:11)

Agent-scale throughput is breaking human-scale infrastructure

Anthropic went from about 2,500 merged changes a day in May to more than 30,000 a day in September. It says this left it bottlenecked by developer infrastructure built for humans, so it built its own version-control system and fully moved its monorepo off Git in five months (write-up) [2]. Theo’s reading is that the bottleneck was GitHub and that “a lot of us are gonna end up where they are today” [3]. In a smaller example, Jullerino says his agent merged almost 400 web-performance PRs in 30 hours. He reports 86% fewer renders when another thread is streaming, up to 35% less JavaScript CPU and a heap about 10% smaller [4].

Geoffrey Huntley has a “spicy” fix for merge conflicts at that volume: throw away the active agent’s working copy and make it rebuild and reapply its changes from the context it still holds in memory [5]. Kent C. Dodds noted that, by Anthropic’s own figures, 10% of its code is still written by hand: “What are you guys doing!?” [6]

Defensive security loops with Daybreak

Huntley is running loops with OpenAI’s Daybreak Blue to audit all of Overalls and says little is coming out. He adds that a clean result doesn’t prove the code is secure [7]. His operating assumption is that autonomous attackers will look at every commit of released source, possibly with more inference capacity than you have. So what matters is how fast you respond. He recommends running self-hosted tools on private Tailscale or WireGuard networks and doing autonomous blue-team exercises on your own code [7].

OpenAI’s Tibo says OpenAI worked with OpenCode to get OpenCode users who are also OpenAI API customers working again after an attack, and to refund any credits the attack would have used. He urges teams to request Daybreak Blue access before they have an incident [8]. Individuals can sign up at chatgpt.com/cyber, and organizations can apply through trusted access [9].

Overalls and inference routing

- **Overalls** (Huntley’s self-hostable agent platform) can now pause and resume Firecracker VMs. Thread-created machines suspend after 4 hours with no tool calls. A new prompt, or an incoming VS Code web, SSH or web-app request, wakes them up. Machines can be forked along with their live memory state, and they move to another bare-metal node when capacity runs out [10]. It can now also run Android as a Firecracker guest, displayed in the browser [11].
- **geoclass early result:** at 2 p.m. in Australia, capacity was short. Routing to Germany got around it, and he plans to switch back once Germany wakes up [12]. His Underclass usage of OpenAI models over about 7 days

came to roughly \$39,314 at retail pricing [13].

T3 Code: performance demo and a disputed benchmark

In Theo’s speed demo comparing T3 Code with Codex, the T3 Code threads ran on remote VPSs and Linux boxes, while the Codex side was mostly local. The two setups weren’t the same [14]. A benchmark fight is unresolved. David Ondrej alleges that Theo paid for a v2 run and asked for Cloudroom to be hidden after it beat T3 Code in v1 [15]. Theo denies this. He says the evaluator had tested a T3 Code version more than three months old, and that he offered to fund a real benchmark [16]. Treat both rankings with caution until a methodology is published. Theo also says that over the last two weeks, more than 60% of T3 Code users who used Codex also used Claude Code [17].

Smaller items

- **Kody:** Kent C. Dodds removed the inline UI generation built on MCP Apps, expecting agents to ship their own version. He is positioning Kody as “the shared cloud for your teams agents” [18].
- **llm-echo 0.5**, Simon Willison’s fake model that repeats prompts back without calling an LLM, now counts tool calls, system prompts and reasoning tokens in its simulated token usage. Try `uvx --with llm-echo llm -m echo 'hi there' -o thinking 1 --usage` [19]. **llm-tools-datasette 0.1.1** fixes a break caused by LLM 0.33’s switch to `httpx2` [20].
- **pi-durable** will get proper hibernation on serverless platforms such as Cloudflare Durable Objects and Vercel, plus telemetry, next week [21].
- In the desktop app, you can now edit earlier messages in a thread, per Tibo [22].
- One user plugged a Kindle in over USB and had Claude Code jailbreak it, strip out Amazon’s bloat and add Tailscale [23].

Sources

1. Make Your Prompts Dumber
2. X post by @AnthropicAI
3. X post by @theo
4. X post by @jullerino
5. X post by @GeoffreyHuntley
6. X post by @kentcdodds
7. X post by @GeoffreyHuntley
8. X post by @thsottiaux
9. X post by @fouadmatin
10. X post by @GeoffreyHuntley
11. X post by @GeoffreyHuntley

12. X post by @GeoffreyHuntley
13. X post by @GeoffreyHuntley
14. X post by @theo
15. X post by @DavidOndrej1
16. X post by @theo
17. X post by @theo
18. X post by @kentcdodds
19. llm-echo 0.5
20. llm-tools-datasette 0.1.1
21. X post by @badlogicgames
22. X post by @thsottiaux
23. X post by @adriandlam__