Make agents prove their own work before review: Theo on the Claude Code team's review pipeline

Coding Agents Alpha Tracker

2026-11-07

Make agents prove their own work before review: Theo on the Claude Code team's review pipeline

By Coding Agents Alpha Tracker • November 7, 2026

Theo walks through how the Claude Code team gets agents to verify their work and limits automated review rounds. Also: Armin Ronacher's small agent "factory", T3 Code data showing Opus 5.5 on max reasoning fails more often than on high, and prompt and config tips.

Make the agent verify its work before anyone reviews it

Theo's new video reacts to the Claude Code team's account of how it reviews agent-written code. It is the most practical source this period. The team's goal is for most PRs to merge with at most one human in the loop, with no rise in reverts or incidents. It measures human approvals, human comments and time spent waiting on a person [1].

Step 1: verify before a PR exists. Give the agent something it can run: tests, a build, or a screenshot to compare against the design. Ask it to show the output rather than just say it's done, then have a subagent review the diff in a fresh context [1]. Theo notes that Boris (who created Claude Code) estimates this makes the final result two to three times better [1].

In Theo's own setup, every thread gets its own workspace where it can start an Electron dev server for T3 Code and record a video of the change working. During development, fake model selections replay pre-baked events, so agents can run full end-to-end tests without spending tokens on model calls [1].

Step 2: agents triage review feedback, not humans. Theo argues that the author's agent should decide whether review feedback is valid. Other agents should vote on each finding before it is posted. Findings about style or cleanup

are dropped, and every remaining finding must be either fixed or resolved with a reason [1].

Step 3: make reviews converge. Theo once left a "keep iterating until the whole council passes" loop running. Four days later the PR had grown to about 80,000 lines and was going in circles [1]. He now has review agents give each PR a 1–10 score against a bar he sets, rather than counting comments [1]. Anthropic's fixes: - after three rounds, ask the author to step back and reconsider the approach - after the first few passes, re-review only the diff - flag the whole class of bug, so every instance gets fixed - never repeat a finding that has already been posted

After these changes, PRs needing five or more rounds fell from 9% to 1.5%. The average number of rounds to pass fell from 2.08 to 1.32 [1].

When an agent keeps making the same mistake, fix things in this order (Theo, drawing on Lauren's guidance): 1. the codebase and its architecture 2. linters, type checks and other static analysis 3. only then CLAUDE.md, review.md, or skills [1]

Humans keep the decisions on direction, taste and scope, on risky code and architecture, on checking the verification evidence, and own the outcome [1].

A talk at Compile NYC described a similar setup. A dedicated PR agent iterates until all CI checks pass, then watches the rollout and alerts the engineer if something goes wrong [2]. In the same talk, a parent agent sat idle for 20 minutes while its subagents worked, which hurt the cache hit rate. The fix was a one-hour cache lifetime for the parent and five minutes for subagents [2].

Armin Ronacher: start your factory small, and pick models by task

On State of Agentic Coding #12, Armin argues that most people build software factories as if they were at "the last stage of Factorio." His advice is to start small and learn where the bottlenecks are before expanding [3]. In the same episode, Ben describes his own small setup. Two Pi Durables on VMs help maintain Honk, mostly by testing user-contributed PRs in their own sandboxes. They have their own GitHub account, talk to each other only over IRC, and work 9–5 on weekdays [3].

- **Models:** Opus 5.5 is Armin's default, mainly because he finds its explanations readable. Most of his agent work is understanding what is going on, not writing code [3]. For reverse engineering or security work, he picks Gemini because it is "the least annoying for that problem," not the newest model [3].
- **Classifier harnesses:** In Pi's code mode, you can ask the agent to build itself a small harness that calls a fast, cheap classifier model. Armin uses this to check thousands of inputs: whether translations still match the

story, finding duplicate GitHub issues, and searching his sessions for a specific failure [3].

- **Guard what a metric goal can sacrifice:** One host asked Opus 5.5 to cut Vercel preview build time. After hours of work, it proposed removing Sentry source-map uploads, which you need to debug during an incident [3].

Telemetry: avoid "max" reasoning on Opus 5.5

Theo gave Adam Holter access to T3 Code's anonymized telemetry [4]. Holter found that Opus 5.5 on max reasoning fails twice as many turns as on high. Max fails more in every turn-length bucket, and most of the extra failures happen in the first 30 seconds. Sol shows the same pattern, but smaller (1.4x). The cause is unknown [5].

Separately, Theo says Opus 5.5's reasoning traces don't reflect what the model is actually thinking. He calls this his own "conspiracy theory," so treat it as a caution about reading traces, not a fact [6].

Prompts and config worth trying

- DHH's one recurring redirect prompt: "could this be simpler?" He says it "yields improvements every time" [7].
- Theo's prompt suffix: "Be thorough in your analysis and bold in your recommendations" [8].
- Addy Osmani: set Claude Code's output style to Concise via `/config → Output style`, or add `"outputStyle": "Concise"` to `settings.json`. You can still ask for full details [9].
- Kent C. Dodds: write architecture decision records (ADRs) and shared primitives. He says agents read them and stop going against important architectural decisions [10].
- Peter Steinberger maintains about 40 open-source projects. One session regularly updates dependencies and checks GitHub issues and PRs. Strict rules decide what the agent may land on its own [11].

Controlling agents remotely, and by voice

Claude Duet now connects to Claude Code sessions. From your phone you can check what's running, approve changes and send updates, for sessions on your computer or in the cloud [12]. Riley Brown tested it by asking for a new `light mode` branch with the instruction not to push. The task reached his local session, though passkey sign-in got in the way [13].

Antigravity added voice mode, built on Gemini Live. You can plan, steer code changes and approve permissions out loud, interrupting whenever you like [14].

Releases

- **Inkling 2 Small (Thinking Machines):** scores 31 on the AA Intelligence Index. It is available for fine-tuning and inference on Tinker [15]. Soumith Chintala says open weights will follow next week, once external safety testing is complete [16]. Elie Bakouch reports 17B active and 594B total parameters, trained on 12T tokens [17].
- **Warp Factories (early access, mentioned in a sponsor segment):** sets up coding agents, repositories, secrets, automations and sandboxes as code. It can replay your past tasks against different models to compare cost and quality. Warp says this cut its own agent cost from $80 to $20 per PR [18].

Sources

1. How Claude Fixes It's Own Code
2. 13 AI | Compile NYC
3. State of Agentic Coding #12 with Armin and Ben
4. X post by @theo
5. X post by @AdamHoltererer
6. X post by @theo
7. X post by @dhh
8. X post by @theo
9. X post by @addyosmani
10. X post by @kentcdodds
11. Should You Still Become a Software Engineer in 2027? | OpenAI Engineer
12. X post by @davidsuttonca
13. We may never get Fable 6 or GPT-7 (Weekly Update)
14. X post by @antigravity
15. X post by @thinkymachines
16. X post by @soumithchintala
17. X post by @eliebakouch
18. The one thing holding theoretical CS together just broke… kinda