Treat verification as infrastructure: Addy Osmani's agent-manager playbook and Theo's case for "gardening" your repo

Coding Agents Alpha Tracker

2026-11-09

Treat verification as infrastructure: Addy Osmani's agent-manager playbook and Theo's case for "gardening" your repo

By Coding Agents Alpha Tracker • November 9, 2026

Two practitioner guides agree on the main point this period: agents improve when they have fast, definitive checks to run against their own work. Also covered: Simon Willison's pattern for long AI-written PR descriptions, Theo's warning about agent-built MCP auth, how Monaco structures agent work, and a counterpoint on MaxxBench.

Give agents a pass/fail check, then remove what slows it down

Addy Osmani frames running agents as managing people. His main rule: give the agent something that returns a definite pass or fail (tests, build exit codes, linters, diff scripts, screenshots) and have it show the result. Don't accept "looks done" [1]. He notes the command that runs these checks is "the single most useful line" in CLAUDE.md [1]. Ways to wire it into Claude Code: - **Ask:** "Run the auth tests after, and continue until they pass." - **`/goal`:** a separate model checks a condition after every turn. - **Stop hook:** blocks the turn from finishing until, for example, the typecheck passes. - **`/code-review`:** sends the diff to a fresh subagent. Scope it to correctness and tell it to ignore style, or it will always find something [1].

His other rules: - **Correct twice, then write it down.** Put it in CLAUDE.md or a shared skill [1]. - **Use hooks or deny rules only for hard requirements.** Drop ALL-CAPS instructions, "think harder" and "don't be lazy" [1]. - **Plan, then split the work.** Agree a plan first. Use `/batch` to split a mechan-

ical change into 5–30 worktree tasks, each with its own PR. Check the slices don't overlap and land shared interfaces first [1]. - **Reset after two failed corrections.** Start a fresh session with a better prompt [1]. - **Answer five questions before an unattended run:** What does done look like? How does it verify itself? What are its boundaries? What turn or spend limits apply? Where does output go, and who approves it? Pilot on three files first [1].

Theo's new video ("Coding Isn't About Code Anymore") explains why this matters more for agents than for people. A 5-minute dev-server start blocks an agent "insanely deeply" and, with 5-minute cache TTLs, wastes tokens on missed cache reads. Slow CI limits how much work you can run in parallel [2]. His examples of this upkeep: - **He deleted about half of T3 Code's unit tests,** roughly 250,000 lines. He restored 4 of 11,000 and reports no regressions the deleted tests would have caught [2]. - **When a bad test reappears,** the contributor goes back to the agent thread and asks the agent why it added the test. The team then fixes the root cause [2]. - **A "thread transfer impact" check runs on every PR.** It replays real Codex and Claude thread data through a simulated pipeline and flags data-transfer regressions in the PR, where a babysitting agent will see them. It takes under 45 seconds [2].

This adds to last period's verify-before-review material. The check is part of the repo itself, and it tells agents what to optimize for.

Long AI-written PR descriptions: write a short human intro, collapse the rest

Simon Willison treats PRs opened by Claude Code or Codex as his unit of work [3]. To keep the agent's description readable, he opens with a short paragraph he has vetted, sometimes adds a screenshot, and puts the full agent-written text inside `<details><summary>`. GitHub Markdown supports this. The summary line is the model name and effort level, such as "Claude Opus 5.5 Medium" [3].

He also shared a prompt sequence worth copying. He asked Opus 5.5 in regular Claude chat to compare a SQLite `parse_number()` extension written in C and in Rust. The Rust builds were 298–321 KB against 14 KB for C, but they were faster [4]. After he asked it to write a custom C decimal parser, the new C version ran at 16–19 ns per row, against 32–55 ns for the Rust versions. He then had it package the work as a report in simonw/research [4].

Don't trust agents on MCP auth

Theo says that every time he has Opus set up an MCP authentication path, it "makes horrible assumptions" and the result doesn't work [5]. Asked about remote machines, it proposed curling a script, copying codes by hand on mobile and notifying the agent manually [6]. He is writing an "MCP the right way" guide for his agents. Their first draft rendered the MCP URL as a clickable

link, and he warns not to trust anything agents say about MCP or connection strategies [7]. If you build MCP auth with an agent, test the remote case yourself.

On usage, Theo ran 8–25 Opus 5.5 threads most of the day and used less than 20% of a $200 Claude plan [8]. His Codex plans get the heavy use because he reviews with Sol, often while Opus reviews the same code in parallel [9]. He also banned a user whose bot left more than 1,000 "review" comments on a T3 Code PR [10, 11].

Mihail Eric: research the code first, and understand your harness

On Latent Space, the Stanford CS146S instructor argues that large codebases need more than a single prompt: 1. research the code 2. propose alternatives for a human to choose between 3. write a plan/spec 4. implement and test [12]

He teaches the harness itself. Students build an agent loop of about 200 lines, then proxy Claude Code's traffic to read its system, plan and compaction prompts [12]. swyx says context for subagents is still "very hacky": they over-fit to stray comments when given too much [12].

At Monaco, Monocoder (built on Coder) provides isolated environments for long-horizon multi-agent work, and a Slack agent named HAL debugs issues and opens PRs [12]. Their rule is "buy to learn, build to keep." They still use Devin for code review [12].

Claude Duet: how to start new Claude Code sessions

Riley Brown found that asking Duet to create a Claude Code session doesn't work. Use **New Session** on your phone, pick the computer and folder, and type the first prompt [13]. Requests to existing sessions need approval and Face ID [13]. He runs Amphetamine so his Mac never sleeps [13]. He rates Claude Code's computer use well below Codex's [13].

A counterpoint on MaxxBench

Last period's MaxxBench result showed large score drops on modified DeepSWE tasks. Salvatore Sanfilippo (in Italian) argues those drops overstate how much weaker the models are. Some modified specs leave out details the hidden tests check. Models sometimes target the tests they expect rather than the written spec. Smaller models are less able to override what they learned earlier [14]. He proposes about 100 genuinely new tasks of the same difficulty, run on the same harness [14].

Quick notes

- **DeepSeek V4.1 Flash 1017 Ultrafast** is on Vercel AI Gateway. Vercel says it generates over 1,000 tokens per second; the model ID is `deepseek/deepseek-v4.1-flash-1017-ultrafast` [15].
- **TensorFold** recipes are in Omarchy's local-AI catalogue (omarchy-local-ai) for Qwen3.8-27B, Flash Next and Nemotron 3.5 Bolt [16]. DHH says TensorFold will be key to Omarchy's out-of-the-box local agents on NVIDIA [17].
- **Asahi Linux** published an LLM policy that "broadly forbid[s]" generative AI [18]. DHH says Asahi rejects upstreaming his group's agent-accelerated Apple Silicon work [19]. Armin Ronacher defends Asahi's right to the policy [20].
- **Open question:** ThePrimeagen says tasks that took Claude 5.5 10 minutes take 40 as a project grows, and asked for ways to prevent it [21].

Sources

1. X article by @addyosmani
2. Coding Isn't About Code Anymore
3. Hide your slop PR descriptions behind
4. parse_number() as a SQLite extension: C vs Rust
5. X post by @theo
6. X post by @theo
7. X post by @theo
8. X post by @theo
9. X post by @theo
10. X post by @theo
11. X post by @theo
12. [One Million AI Engineer Special] How Stanford Teaches AI-Native Engineering — Mihail Eric, Monaco
13. Claude Duet's NEW Update Is INSANE (Claude Code From Your Phone)
14. MaxxBenched, ma: importa davvero?
15. X post by @vercel_dev
16. X post by @ashxhart
17. X post by @dhh
18. X post by @dhh
19. X post by @dhh
20. X post by @mitsuhiko
21. X post by @ThePrimeagen