Turn agent failures into a score: Theo shrinks T3 Code's tool descriptions, LangChain picks Stagehand by eval

Coding Agents Alpha Tracker

2026-11-04

Turn agent failures into a score: Theo shrinks T3 Code's tool descriptions, LangChain picks Stagehand by eval

By Coding Agents Alpha Tracker • November 4, 2026

Several practitioners describe the same loop: collect agent failures, turn them into a measurable eval, and let agents improve against it. Also: Addy Osmani's verification habits, Claude Code 2.1.317's game-rule fix, and a fake T3 Code site that ChatGPT kept recommending.

Turn failures into a score, then let the agent improve it

Several practitioners this period describe the same move: collect the agent's actual failures, turn them into something measurable, and let agents improve against that number.

Theo (T3 Code MCP). Models kept misusing T3 Code's MCP tools. For example, they opened a new thread when he wanted a subagent. He went through his own history, categorized every case where a model did the wrong thing, and turned them into an eval. Sonnet 5.5 initially got roughly 60 of 88 right [1]. He then gave the agent two jobs in order: 1. Reach a perfect score on every model he cares about, and commit. 2. Cut the tool descriptions to a third of their size while keeping the perfect score [1].

The result was 52% fewer tokens from T3 Code, or 78% with tool search on. It is in nightly now, with stable coming soon [1]. His general rule: if you can turn a problem into a program that outputs a score, "your agents are going to make that score go up" [1].

For subjective UI work he uses a "council" instead. Opus 5.5, Muse and Codex

GPT-6.1 each get the request and context, review read-only, and score out of 10. A fresh review after fixes counts as a new round [1]. He warns the scores are noisy: the same screenshot can get three different scores from one model [1]. Context also moves them. When he added Claude Code and Codex screenshots for comparison, Codex rated a PR it had earlier scored 9 at only 7.4 [1].

LangChain (browser tools). LangChain tested seven browser toolsets on 100 WebVoyager tasks with Claude Sonnet 5.5. Stagehand passed 90% for $1.98 total, at a median 17.4 s per task. Playwright MCP passed 78% for $26.71 and took about three times as long [2]. What made the difference: - **Code mode:** the `run` tool batches a chain of actions into one call. It averaged 10 steps per task versus 26 for Playwright MCP, and a navigation returned a median 117 characters versus about 17,000 from Browserbase's hosted tool [2]. - **Playwright-style code:** let the model write Playwright-style code, which models write best. A `run` tool that took Stagehand's native API instead passed 5.3 points fewer tasks and cost 19% more per passed task [2]. - **Fewer fixed sleeps:** traces showed agents writing fixed sleeps like `page.waitForTimeout(8000)`. Ending each wait once the page has been quiet for 500 ms cut median HardBench time from 182 s to 124 s (together with two other changes), with accuracy within noise [2].

To use this in open-source Deep Agents, add `StagehandMiddleware` from `langchain-stagehand` [2].

Warp (team scale). Zach Lloyd defines one "factory" per product surface in a YAML file listing repos, MCPs and workflow steps [3]. He tracks human interventions per merged PR and cost per PR [3]. Tuning (less Opus where it wasn't needed, fewer redundant subagents) cut cost per PR from about $80 to roughly $15–20. PR throughput rose 40% over the month with no increase in reverts [3]. He puts the floor at roughly 5–10 engineers; below that, "use the cloud code sub or the codec sub" [3].

Armin Ronacher adds that your harness already stores sessions on disk. After a month you have plenty of traces to point an agent at, for example to ask why it keeps failing to use `uv` [4]. Two more tips from his talk: - If you edit files outside the agent, tell it. Otherwise it gets confused, and in his sandbox test it claimed to have validated a fix without running any tools [4]. - For restricted mini-languages, return errors that say which construct is unsupported so the agent can adjust [4].

Addy Osmani's checklist for reviewing agent work

- **Before the task:** write down your own hypothesis [5]. Brief the agent in your own words with the audience, what's out of scope and the definition of done. If two rounds of corrections haven't worked, rewrite the prompt [5].
- **Define "verified" first.** Decide it before the work starts. When the agent reports it's finished, check the command it ran and the output before

you read its summary [5]. When a check passes, ask what it proved and what it didn't [5].
- **Self-critique every time:** have a fresh-context subagent review the work on every run, then ask "Did I miss anything?" [5]
- **Review decisions, not just the diff:** ask the agent to list its decisions and assumptions, then quiz yourself. One rule he quotes: "I only merge after I pass the quiz perfectly." [5]

Releases

- **Claude Code 2.1.317:** you can name an existing game to describe the mechanics you want, and Claude should build an original game. If you ask for a copy, it offers something similar instead [6]. Theo calls the copyright-steering prompt changes "lame as hell," but that post was not among the sources read here.
- **Claude plugins:** Team and Enterprise admins can set up company plugins so teammates connect automatically on first use [7]. Plugin builders need MCP servers that accept Claude as a trusted issuer [8].
- **T3 Code:** fixes that steer subagents to the right tools and stop them blocking on questions, performance wins for big repos and long threads with many subagents, and inline notices when you switch models or effort levels [9].

Security: check where you download agent tools

A user asked ChatGPT for the T3 Code download and was sent to `t33.codes`, a near-pixel-perfect clone registered two weeks earlier. The real site is `t3.codes` [10]. Theo says ChatGPT recommends the scam clone "constantly" [11].

Separately, Fireship reports that a Cursor-shipped Claude Code extension scanned machines for keys and tokens and set up a backdoor. It was live for about 11 hours and reached around 5,000 installs [12].

Shorter notes

- **Spawn's fully autonomous loop:** Simon Willison relays Jacob Sansbury's account. Agents ship to production hourly, an hour without a ship counts as a SEV-0, and rollback takes 30 seconds. Sansbury says agents only keep a few hours of work in their heads, so shipping fast keeps them coherent [13]. They found Opus 5.5 not quite good enough, and GPT-6 Astra "fixed" a security review by disabling every feature [13]. Willison doubts this suits higher-stakes software [13].
- **Triage AI-assisted PRs with an agent:** Willison uses Opus 5.5 or GPT-6 Astra to triage incoming PRs and treats them as "bug reports with attached examples." His latest LLM release included fixes from 12 authors [13].

- **Agentic QA on Omarchy:** ThePrimeagen has agents explore and test Omarchy with role prompts like "You are the theme tester" [14]. He reports the first bug found this way [15]. DHH's goal is to automate the chain from bug fix to PR to QA to release [16].
- **Self-hosted vs frontier models:** a viral post claims Revolut moved about 2,000 engineers to self-hosted GLM on 56 B200s, at about a tenth of its previous API cost [17]. Nicholas Moy says his team made a similar bet and lost, calling it "picking up pennies in front of a steamroller" [18].
- **Cat Wu leaves Anthropic:** in her farewell post she notes that early Claude Code users approved every permission prompt by hand. Today the default is to trust Claude to auto-approve, and it handles large migrations [19].

Sources

1. Agents Sucked At UI (Until Today)
2. How We Optimized Browser Use for Deep Agents
3. The 2026 Software Factory Infra Stack — Zach Lloyd, Warp.dev
4. Leaning In To Find Out | Armin Ronacher | PyAI Conf 2026
5. X article by @addyosmani
6. X post by @addyosmani
7. X post by @bcherny
8. X post by @ClaudeDevs
9. X post by @theo
10. X post by @doriandeacu
11. X post by @theo
12. The worst week in cybersecurity history… so far
13. Qwen 3.8 27B on my laptop can interpret model weights, too
14. X post by @ThePrimeagen
15. X post by @ThePrimeagen
16. X post by @dhh
17. X post by @guilleflorvs
18. X post by @thenickmoy
19. X post by @_catwu