

Theo turns T3 Code’s tool schema into an eval:
4× better tool use with 79% less context

Coding Agents Alpha Tracker

2026-10-27

Theo turns T3 Code’s tool schema into an eval: 4× better tool use with 79% less context

By Coding Agents Alpha Tracker • October 27, 2026

Theo’s eval-driven tuning of T3 Code’s tools and his newly published review and autonomous-merge skills give the most reusable workflow material. Also covered: OpenAI’s GPT-6.1 Terra, Cursor’s Agent SDK and DeepSeek V4.1 Flash, evidence that the harness changes benchmark results, and a debate over tests and humans in the loop.

Turn your agent’s tool failures into an eval, then hill-climb

Theo says agents now use T3 Code’s tools “4x better while using 79% less context” [1]. T3 Code’s tool surface had grown with the built-in browser, mobile device manager, subagents and delegates, so he built an eval from how his agents actually used the tools, including “all the failures and rage moments.” It has over 80 cases, and Opus failed more than half of them at the start [2]. His loop is one you can copy for your own MCP servers or tools:

1. Benchmark a cheaper model (Sonnet 5.5) and keep changing tool descriptions until it gets a perfect score.
2. Add Sol, then Opus, and do the same for each.
3. Merge once error rates improve.
4. In a new thread, tell Opus to cut context use by 50% with no regressions on the eval, then ship [2].

He says this “shape” is close to the one he used for his earlier UI-overhaul hill climb [2]. He also posted that the run cost about \$1,500 in tokens.

The result shipped in **T3 Code 0.1.3**, which brings stable up to date with nightly. In this release, T3’s agent instructions are 60% shorter, MCP tool schemas are a third smaller, and “steer” replaces queueing as the default. Open-

ing a thread loads 85% less data, guest sharing takes two clicks, and T3 Connect switches to LAN when other machines are on the same network [3].

Theo's skills, now public as tstack

Theo is publishing his skills at t3dotgg/tstack [4, 5]. A walkthrough video explains three of them:

- **Descriptions are triggers.** A skill's description should say *when* to use it (for example, "use when the user says Soul Loop"), not what it is [6]. When an agent misbehaves, he asks it why and then patches the skill [6].
- **Review loop with a budget.** Opus writes the code and GPT-6.1 Sol reviews it. The skill says to make minimal changes, watch for scope creep, and push back when a finding is wrong. Three rounds at most: the first is a full review, and later rounds check only changed code and open findings. It can go to five rounds for security, permission or data-loss risk, and stops early if only nits remain. Only one reviewer runs at a time, with no nested subagents [6]. The PR must report the reviewer's verdict, because he noticed Opus "almost wants to hide" disagreement [6].
- **"Full send" and "full stop."** Full send hands the agent everything up to a merged PR. It explicitly grants merge and force-with-lease permission, which he added because OpenAI models hesitate to merge without it [6]. The agent must show proof it observed itself: reproduce the failure, fix it, see the reproduction pass, run the repo's CI checks, and add one focused test [6]. The merge gate requires proof covering every change, green required checks, every bot finding fixed or dismissed in writing, no open human change requests, and a branch that is up to date and mergeable [6]. In a recent window, 27 of 32 full-send threads merged without him looking again [6]. Full stop does everything except merge, for riskier changes [6].



These 3 skills changed my life (17:51)

Theo also tells users not to add multiple accounts directly to T3 Code. He uses a CLIPProxy fork to share 7 Claude and 3 Codex subscriptions across 12 machines [7], and posted a setup gist you can give to an agent [8]. He warns that Anthropic expects Claude subscriptions to be used only with the Claude Code system prompt [6].

Models and tools

- **GPT-6.1 Terra (OpenAI):** pitched as fast computer use at a lower price [9]. OpenAI's Tibo says it gives 2× more usage than 6.1 Sol across API and subscriptions, with better writing and understanding of requests [10]. Greg Brockman says it feels faster at using a computer than he is unassisted [11].
- **Cursor Agent SDK:** define agents as code, with prompts in Markdown and tools in TypeScript, then run them on Cursor's hosted infrastructure or your own. Evals are built in [12].
- **DeepSeek V4.1 Flash in Cursor:** Cursor says every effort level is on the CursorBench cost frontier at \$0.60–\$0.98 per task. Turn it on under Settings > Models [13].
- **The harness changes results:** a posted evaluation found the Pi harness ended Gemini 4 Argon runs on fallbacks in 20% of cases. Rerun in the Antigravity harness, Argon beat Astra and came just behind Opus 5.5

[14]. Check the traces before you trust a model comparison.

- **Theo's picks:** his “realistic pareto line” is GPT-6 Luna, GPT-6.1 Sol and Opus 5.5 [15]. He says Muse finds bugs Opus misses and writes prompts and skills better [16].
- **Claude Code:** `--dangerously-skip-permissions` still blocks things like `rm -rf ~`. A new env var, `CLAUDE_CODE_EVEN_MORE_DANGEROUSLY_SKIP_PERMISSIONS=1`, turns that last block off [17].

How much human in the loop?

Brent Traut argues unit tests give agents little safety because “AI just changes all the unit tests anyway” [18]. When replies said to keep test changes separate from code changes, he questioned whether people really have humans reading or writing every test [19]. He says improving loop automation is “extremely leveraged work” [20]. Theo’s proof requirements and merge gates are one concrete version of that automation.

Addy Osmani’s rule for subagents: keep one only if the work is parallel, needs fresh context, or one agent can’t pick the right tools. Otherwise, move its prompt into a skill, fold it into the main loop, and run evals [21].

Shorter notes

- **Billing watcher:** Anj Nanda suggests an always-on agent, which can be Claude Code, that checks transactions and logs every 30 minutes, fixes problems and messages you. He says it prevented at least three incidents and \$30,000 in losses for about a \$200 subscription [22].
- **Entire:** a new Git forge for agent-human collaboration [23]. Jason Zhou says linking agent sessions to commits is “such an unlock” [24].
- **Managed Deep Agents 0.9 (LangChain):** agents can create schedules, each run can pick its model, skills and tools, and agents react in Slack before replying [25]. The `mda init` / `mda dev` / `mda deploy` CLI scaffolds, runs and ships agents [26].
- **Token costs:** Theo says cached reads and writes make up most of his Opus spend, while plain input and output tokens are under 8%. That makes total token counts a poor estimate of cost [27].
- **Bun AOT:** `--trim` shrinks a macOS `Bun.serve` binary from 89 MB to 25 MB [28]. Peter Steinberger reports 2–3× better performance and less than half the memory [29].

Sources

1. X post by @theo
2. X post by @theo
3. X post by @theo

4. X post by @theo
5. X post by @theo
6. These 3 skills changed my life
7. X post by @theo
8. X post by @theo
9. X post by @OpenAI
10. X post by @thsottiaux
11. X post by @gdb
12. X post by @cursor_ai
13. X post by @cursor_ai
14. X post by @wenhaocha1
15. X post by @theo
16. X post by @theo
17. X post by @The_Whole_Daisy
18. X post by @btraut
19. X post by @btraut
20. X post by @btraut
21. X post by @addyosmani
22. The Most Terrifying Week in AI History...
23. X post by @ashtom
24. X post by @jasonzhou1993
25. X post by @LangChain
26. Inside Managed Deep Agents: mda CLI, Context Hub, and Harbor Evals
| Interrupt London
27. pstack creator spends \$132,000,000/year on ai????
28. X post by @jarredsumner
29. X post by @steipete