Ronacher: have the agent write its own test suite before porting a codebase

Coding Agents Alpha Tracker

2026-11-02

Ronacher: have the agent write its own test suite before porting a codebase

By Coding Agents Alpha Tracker • November 2, 2026

Armin Ronacher argues that agent-led reimplementation works best when the agent first writes tests against the original, and points to a readline port as an example. Also covered: Karpathy's HackerTube process, the return of `ultrathink` in Claude Code, and T3 Code's managed policies.

Porting with an agent: have it write the tests first

Armin Ronacher disagrees with the common reply that "the value is in the testsuite." In his view you don't need an existing test suite to rebuild software. You are "probably better off having the agent build its own test suite specifically for porting," partly because "lots of pre-existing tests are bad for hill climbing." His example: a few weeks ago an agent reimplemented Excel's formula engine and built its own test suite from scratch [1].

His public example is **lineread**. It is readline reimplemented against synthetic tests the agent first wrote against the original readline. The steps: 1. Have the agent write tests that pin down how the original behaves. 2. Have it build the new implementation against those tests.

He calls the result "basically a GPL to MIT laundromat" [2]. Take that as a warning that this workflow raises licensing questions, not as a legal opinion.

He also predicts that an Opus 5.5 agent could reimplement all of CPython "probably within two weeks without any human input, against just the tests." He thinks it might reach full ABI compatibility, drop the GIL, add a JIT, and beat CPython on every measurable metric [3]. This is a prediction, not a result. His advice for anyone who tries: "make it look at the JVM rather than cpython for design decisions" [4].

Karpathy's HackerTube: the human handles only accounts and infrastructure

Karpathy built HackerTube in about two hours with Opus 5.5. Claude filtered 43M Hacker News comments from 20 years down to 6,454 videos that people keep recommending [5]. His default setup is the Claude Code desktop app for Mac with Opus 5.5 on **Medium** effort. He talks only to Claude and has it do everything. He did three things by hand in the browser: - created a Google Cloud API key for YouTube metadata; - started a VPS; - bought the domain.

Then he gave Claude the server's IP and told it to "update, secure, harden, deploy and optimize the code," and it did all of it [5]. He published the comment-analysis data as one zip file. The idea is that you point your own Claude at it and ask for your own version [5].

Claude Code: `ultrathink` works mid-turn again

`ultrathink` is back in the Claude Code CLI. Anthropic's @dmwlff says it had been turned off for a while because it was "weird and maybe confusing." He mostly runs Opus 5.5 on medium and uses `ultrathink` on turns that need more effort [6]. Addy Osmani's tip: if a task turns out harder than expected, type `ultrathink` *while Claude is working*. Effort goes up for the rest of that turn. This works in the terminal on the latest release [7]. Together with Karpathy's Medium default, this suggests a pattern: run at medium effort and raise it only for the turns that need it.

T3 Code: org policies, and no fixed agent roles

T3 Code now supports **managed policies**, so IT teams can configure how it runs across an organization [8]. Theo says it is used internally at companies "from Microsoft to AWS to Apple" [9]. Smaller additions: custom "snooze" time options [10], and one user reports that using T3 Code on a phone to reach a remote VPS stayed smooth on poor travel internet [11].

On design, Theo says T3 Code is exploring "bot"-like systems, "specifically meta threads for commanding your threads/agents/envs." It will not have defined roles, which he calls "artifacts of a time where models were much dumber" [12]. He says everyone disagreeing so far has been a heavy local-model user [13]. A widely shared interview teaser argues the opposite. It credits a SpaceXAI engineer (formerly at Cursor) with running 70+ single-job bots, each with its own memory. She talks only to a "Chief of Staff" bot that watches Slack and starts work on its own [14]. The post is promotional, so treat the details as unverified.

Short notes

- **uv + Python 3.15:** Simon Willison's rule is "If in doubt, delete the uv.lock file." Git-ignored `uv.lock` files left behind by `uv run pytest` can pin dependencies that don't have 3.15 wheels yet. His example was `watchfiles==1.1.1` failing because PyO3 supports up to 3.14 [15].
- **Trust in agent-built software:** Willison quotes Kyle Kingsbury's Jepsen analysis of ParticleDB 1.1.4: LLMs make the usual signs of a mature system "easier to counterfeit." Kingsbury asks how maintainers can show users their software is correct and documented accurately [16].
- **Antigravity UI sidecars:** Google DeepMind researchers built an interactive 3D biomolecular structure viewer as a sidecar in the Antigravity workspace. It let them inspect structures alongside the agent while searching millions of uncharacterized proteins [17].
- **Production agent controls:** Simon Budziak's walkthrough covers five controls, with code, for agents built on LangGraph and LangSmith. In one, a confidence gate sends about 15% of conversations to a clinician, and clinicians judged 80% of those handoffs appropriate [18].
- **Ask for many options in one image:** For an app icon, Salvatore Sanfilippo asks the image model for one sheet with many different icons in a single pass instead of waiting through one-at-a-time attempts [19]. Once a concept looks promising, he asks for a panel of variations on it [19]. GPT pointed out that the phone in his chosen design was too small at icon proportions. He then generated four close-up variants and picked B [19].



Generare l'icona di un'app per iPhone con l'AI: il mio processo (2:03)

- **Google Carbon:** Dylan Patel apologized to Google, saying "they cooked with Carbon." He calls it spikier across domains and capabilities, but "really good in a few areas" [20].

Sources

1. X post by @mitsuhiko
2. X post by @mitsuhiko
3. X post by @mitsuhiko
4. X post by @mitsuhiko
5. X post by @karpathy
6. X post by @dmwlff
7. X post by @addyosmani
8. X post by @theo
9. X post by @theo
10. X post by @theo
11. X post by @theayush
12. X post by @theo
13. X post by @theo
14. X post by @akarixbt
15. If in doubt, delete the uv.lock file

16. Quoting Kyle Kingsbury
17. X post by @antigravity
18. X post by @LangChain
19. Generare l'icona di un'app per iPhone con l'AI: il mio processo
20. X post by @dylan522p