Pi's maintainers: trust agents where there's an oracle, review core code yourself

Coding Agents Alpha Tracker

2026-11-17

Pi's maintainers: trust agents where there's an oracle, review core code yourself

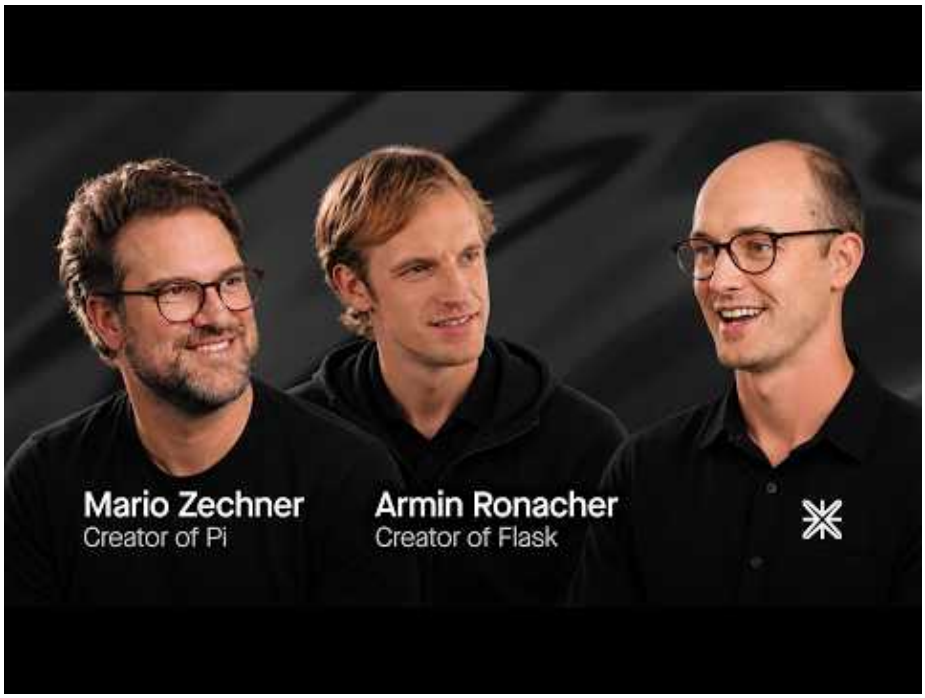
By Coding Agents Alpha Tracker • November 17, 2026

The creators of the Pi coding agent describe how they triage issues, scale code review and use agents for design. Also: Simon Willison's external-sandbox pattern, LangSmith rating coding agents by merged PRs, and pushback on macOS 27.3's new agent warnings.

How the Pi maintainers decide what to trust

On a livestream with Armin Ronacher, Mario (one of Pi's creators) gave the clearest map this period of where agents can be trusted. He says the jump from 4.8 to 5.5 models didn't make them better at system design, architecture or API design. What did improve is orchestrating subagents and "anything that is verifiable through tests or some other oracle" [1]. Ronacher agrees that since Opus 5.5 you can leave some tasks unsupervised for longer. His example is matching an implementation to a reference test suite, and he says this "doesn't generalize" [1].

Issue triage bot. Pi gets 40–50 new issues a day. A bot built from a simple prompt checks whether the bug is real, tries to reproduce it reliably, proposes a fix and assigns a priority. The team tuned it for one to two weeks on old data. When a P0 issue comes with a test that fails on the current code and a fix that makes the test pass, Mario can usually sign off and merge. For feature work involving him as a user, he calls the results "hilarious" [1].



LIVE: Writing code that doesn't suck with the creators of Pi (16:44)

Match review depth to how critical the code is. Mario still reads the full diff for anything touching tools, the agent loop or core primitives. For non-critical code, such as the HTML session export, he cares only that the result is right. He credits reading the fixes with keeping him familiar with the codebase after 12 months of agent-written fixes [1].

Adversarial review across model families. For system design, he gives models from different families the same inputs to review and gets "super different" problems back. They can be overzealous, so he filters the findings with his own judgment [1]. This matches Theo's Council approach from last period. One difference: the hosts push back on running reviewers in a loop "until nothing comes up" [1].

Explore designs in parallel. Instead of writing scratch programs by hand, Mario has agents write about 10 versions of an API or set of types. He then iterates "until it feels simple enough yet capable enough" [1].

Two cautions: - Green CI isn't proof. Agents are still "reward hacking little bastards" that may measure the wrong thing [1]. Mario's example: an overnight goal to cut Pi's startup time reported a 100% improvement because the agent had ripped out all the modules [1]. - Ronacher says agents lean toward defensive complexity and adding features. Extra protections can make code look secure while it misses something important [1].

Domain vocabulary also matters. Using the right terms and saying what order to do things in noticeably changes results [1].

External sandboxes: agent on the Mac, code in a VM

Simon Willison, working with Prime Radiant (the lab behind Superpowers), published an "external sandbox" pattern. Claude Code, Codex or Pi run on your Mac, but the code they write runs in a WebAssembly, QEMU or Apple Container sandbox. Files move in and out through mounted folders, and network traffic goes through an allow-list proxy [2]. Each sandbox server has an optional HTTP/HTTPS allow-list plus credential replacement, so the agent can make authenticated API calls without seeing the secret [3]. That addresses the agent-credentials problem raised last period. The index repo has a protocol spec, a conformance runner, implementations using v86, Apple Container and Quicksand/QEMU, and a Pi extension published to npm [4].

macOS 27.3 adds warnings, practitioners object

MacRumors reports that macOS 27.3 warns against giving AI apps full control of the Mac [5]. Peter Steinberger says people asked for fewer permission alerts so agents could work "without constant babysitting," and got "new scary alerts" instead [6]. DHH says users excited about agents should "get you a new OS" [7]. Separately, Steinberger told another developer to "move to cloud agents. local isn't the future" [8].

LangSmith rates coding agents by merged PRs

LangChain rebuilt LangSmith around an agent record. Each agent has an owner, an immutable ID and a type, and its local, dev, staging and production environments sit under that one ID [9]. The coding-agent type judges agents by whether their code gets merged. Its dashboard shows merged PRs, token cost and type of work, and a GitHub connection tracks downstream outcomes. Because every agent of a type is measured the same way, results are comparable across engineers [9]. You can connect agents that run elsewhere without changing your workflow [9]. It's available in cloud, BYOC and self-hosted versions [9].

Models: knowing when to stop

- **Fable 5.2:** Theo calls it the first model that "consistently ignores unimportant feedback." At least a dozen times, Opus "started running in circles" on a thread, and switching to Fable got a "this is fine just ship it" [10]. That makes Fable one option for breaking review loops.
- **Gemini 4 Argon:** Logan Kilpatrick's go-to test prompt is a Minecraft-style clone with procedural terrain, block placement and survival mechanics, followed by two or three short fixes such as "It's a bit laggy. Fix that"

[11]. He finds its front-end output overcomplicated by default, and asking explicitly for a much simpler UI works well [11]. Because more capable models keep working instead of cutting their losses, he advises scoping work tightly [11]. An Android tour-guide app built in AI Studio took 23 minutes and ran in a real emulator [11].

Tool updates

- **Claude Code desktop** now supports iPhone Duo in the iOS simulator (public beta). Claude can fold and unfold the device to test each layout [12].
- **Antigravity** rebuilt its UI from scratch as a place to orchestrate multiple agents in parallel [13].

Sources

1. LIVE: Writing code that doesn't suck with the creators of Pi
2. X post by @simonw
3. X post by @simonw
4. X post by @simonw
5. X post by @MacRumors
6. X post by @steipete
7. X post by @dhh
8. X post by @steipete
9. Why we rebuilt LangSmith around the agent
10. X post by @theo
11. Google's AI Studio Lead Shows How Gemini 4 Builds Android Apps
12. X post by @ClaudeDevs
13. X post by @antigravity