Agentic Testing Playbooks and Safer Rails for Coding Agents

Coding Agents Alpha Tracker

2026-05-17

Agentic Testing Playbooks and Safer Rails for Coding Agents

By Coding Agents Alpha Tracker • May 17, 2026

Salvatore Sanfilippo's `testing.md` pattern was the clearest reusable idea today: use an LLM as a QA engineer for nondeterministic systems, then rerun it continuously. Also inside: secret brokering plus gated PR patterns, the OpenClaw vs Hermes local benchmark, and the new tools and projects worth trying.

TOP SIGNAL

- **Agentic testing went from slogan to playbook.** Salvatore Sanfilippo says fixed suites are not enough for systems with timing, tool-calling, or other nondeterministic behavior; his replacement is a `testing.md` file that tells an LLM to act as QA, run tool-calling sessions, hit API endpoints, and simulate user operations. He also has the model generate stress programs and reruns the prompt after major changes so both model sampling and system variability explore new integration states; he says this catches bugs earlier and can run continuously with a second LLM verifying reports before they become issues. [1]

TRY THIS

- **Write `testing.md`, then rerun it after every meaningful change.** Sanfilippo's recipe: (1) create `testing.md`; (2) tell the model it is the QA engineer; (3) have it do tool-calling sessions via Codex, API, or Agent Cloud plus endpoint checks and manual-user-style ops; (4) for stateful systems, ask it to generate Python stress tests that scale from 10 to 50 million entries, add replication consistency checks, save/reload cycles, and mid-operation connection breaks. The point is coverage through variation, not one perfect deterministic script. [1]

- **Broker secrets and gate writes.** Quinnypig's pattern: the platform keeps the secret, the agent only gets a handle, and service calls go through that broker. Pair it with a flow where the agent opens a PR, kicks off an Action, and waits for human or second-agent review instead of mutating prod directly; Kent C. Dodds says this matches his year with Cursor cloud agents, and says Kody already uses the brokering model. [2, 3, 4, 5]
- **Turn repeated work into explicit skills.** Greg Brockman's prompt: `Look through my Chronicle memories and check for workflows that i'm repeating multiple times. Turn them into skills.` The useful twist is recall: Chronicle surfaces the daily tasks you already forgot, so the skills come from real repetition instead of guesswork. [6, 7]
- **Start with the smallest permission set that works.** Armin Ronacher's `pi` examples: `pi -nes -tools bash -append-system-prompt "Use the patch binary to make edits"` for bash-only edits, or `pi -nbt` for no-tools mode. Good pattern when you want predictable edits before giving an agent broader reach. [8, 9]

WHAT SHIPPED

- **Local bake-off: OpenClaw beat Hermes on the same browser+research task.** On a MacBook Pro M5 Max 64GB running a local Qwen 35B model, both agents were asked to scrape GitHub star history for both tools, explain the spike causes, and build a live dashboard; both succeeded, but OpenClaw finished in 12m01s / 203k tokens vs Hermes at 33m01s / 257k. OpenClaw leaned on GitHub API pagination plus star-history JSON; Hermes used parallel GitHub/web/browser calls and switched from Google to DuckDuckGo after rate limits. comparison [10, 11]
- **lossless-claw 0.10.0** adds rotated conversation segments and full-sweep compaction to preserve long chats without churning hot prompt caches. Steipete describes the broader OpenClaw memory model as compacted conversation blocks arranged like a lookup tree for past messages. release thread [12, 13]
- `codex-complexity-optimizer` is a new open-source Codex skill with one-command install: `npx -yes codex-complexity-optimizer`. It scans for loops, repeated lookups, render-heavy code, N+1s, and `O(n²)` / `O(n*m)` patterns, then reports before/after complexity estimates, safe optimization ideas, and required tests; default is report-only mode. Greg Brockman surfaced it as a Codex use case. post [14, 15]
- **Access and billing workarounds are getting more creative.** Zed's agent now accepts ChatGPT subscriptions with the same usage and rate limits as Codex, while a community 200-line bash wrapper keeps `claude -p` running through an already-open Claude Code session so calls land on

the existing subscription instead of the new Agent SDK credit bucket after June 15. Zed Claude wrapper [16, 17, 18]

- **Zero** is a new systems language pitched as easier for agents to use and repair, with explicit capabilities, JSON diagnostics, and typed safe fixes. Armin Ronacher said he has not tried it yet, but that it does several things he recently argued for in a language for agents. announcement [19, 20]
- **Codex shortcuts are now customizable.** Small ship, real daily impact: settings can now be tuned around your workflow instead of forcing default keybindings. update [21]

GO DEEPER

- **06:52-09:22 — Salvatore on the `army of virtual users` idea.** Best clip of the day if you build against tool-calling APIs or other messy systems: why agentic testing fits better than rigid suites, and how a second LLM can filter false positives before escalation. [1]

Un nuovo tipo di test potrebbe presto emergere (6:52)

- **04:39-05:14 — the minimal `testing.md` setup.** Short and immediately reusable: create a markdown playbook, tell the model it is the QA engineer, then have it exercise tool calls, API endpoints, and manual-user-

style flows. [1]
Un nuovo tipo di test potrebbe presto emergere (4:39)

- **Project to study — pi.dev.** Worth digging into if you want tighter control over agent authority. The interesting part is the ability to keep the agent in bash-only mode or drop to no-tools mode entirely. [8, 9]
- **Bug-finding workflow to study — Clawpatch.ai.** Steipete's recommendation is direct: run it on one of your repos and let Codex surface bugs you did not know you had. Worth a closer look if bug discovery is what you want from Codex right now. [22]

Editorial take: the durable edge today is not more autonomy by itself; it is stronger test surfaces and harder rails around the agent you already have. [1, 2, 3]

Sources

1. Un nuovo tipo di test potrebbe presto emergere
2. X post by @QuinnyPig
3. X post by @QuinnyPig
4. X post by @kentcdodds
5. X post by @kentcdodds
6. X post by @kr0der
7. X post by @gdb
8. X post by @mitsuhiko
9. X post by @badlogicgames

10. X post by @atomicbot_ai
11. X post by @steipete
12. X post by @jlehman_
13. X post by @steipete
14. X post by @Kappaemme1926
15. X post by @gdb
16. X post by @zeddotdev
17. X post by @aibuilderclub_
18. X post by @jasonzhou1993
19. X post by @ctatedev
20. X post by @mitsuhiko
21. X post by @OpenAIDevs
22. X post by @steipete