Coding-Agent Loops Need Contracts, Not Just Prompts

Coding Agents Alpha Tracker

2026-07-14

Coding-Agent Loops Need Contracts, Not Just Prompts

By Coding Agents Alpha Tracker • July 14, 2026

Today's strongest practical pattern is the durable coding-agent loop: a clear contract, persistent state, cost-aware triggers, and verification that returns feedback to the original agent. Also included: Codex containment tips, new agent-team and collaboration features, and two implementation walkthroughs worth watching.

TOP SIGNAL

Build agent loops as durable operating systems, not repeated prompts. Jason Zhou's team uses a single Markdown document to hold a loop's goal, autonomy boundaries, SOP, durable state, and append-only run log; after a month automating much of SuperDesign, they report that some loops worked and some did not. [1, 2] Pair that contract with the right trigger: their cost-conscious "combo" pattern checks a data source with a script first, then wakes the agent only when there is real work. [1]

TRY THIS

- **Put every recurring agent task behind a `LOOP.md`.** Define: (1) a finish line, (2) what the agent may ship or merge alone versus what must escalate, (3) a repeatable SOP, (4) a compact current-state section for hypotheses/backlog/follow-ups, and (5) an append-only log. For scheduled work, run a cheap script first; if it finds no changes, exit without invoking the model. Jason Zhou's team uses this structure for loops such as daily documentation-drift checks and automated React issue cleanup. [1]

- **Return review findings to the author-agent, then require evidence.** Peter Steinberger's PR/commit skill invokes an available review CLI, but sends the findings back to the original coding session—the one that understands the task's constraints—rather than asking a fresh reviewer to blindly patch them. Have that session record accepted or rejected review decisions in the PR description, and define repository invariants in `agent.md` so reviewers do not "fix" intentional behavior. For UI or setup-sensitive changes, give the verifier a fresh-machine box with screenshots and click capability. [3]
- **Contain Codex before it contains your rate limit.** Theo's current baseline is **High** reasoning, with **Fast mode** and **Ultra** turned off; he characterizes Ultra as an instruction to use more subagents rather than a reasoning level. [4, 5] Add this guardrail to global `agents.md`:

  `Only use subagents if the user explicitly requests them` [4]

Then control autonomy in the task prompt: require a plan and feedback checkpoint, or let the agent build, test, open a PR, handle the first review round, and stop there. Theo's point: explicit stop conditions are a cleaner control surface than repeatedly changing the harness. [4]

WHAT SHIPPED

- **Antigravity Agent Teams:** run `/teamwork-preview` to create specialized subagents that coordinate in the background to plan, build, and verify complex engineering work in parallel. [6]
- **Claude Code Artifacts:** artifacts now support public sharing, multiplayer editing, and creation with Claude Tag. [7]
- **Loopany platform:** SuperDesign open-sourced its internal loop-management tool, which centralizes contracts, state, logs, triggers, and periodic "evolve" runs. Its templates include documentation maintenance, React Doctor, and tech-debt cleanup. Repo [2, 1]
- **Deep Agents Code on NVIDIA NemoClaw:** LangChain says a single command deploys Deep Agents Code as a governed blueprint using Nemotron 3 Ultra, while retaining control of source, model, and audit trail. Details [8]
- **Field signal, not a benchmark:** Simon Willison's `sqlite-utils` 4.0rc2 was "mostly written by Claude Fable" for about $149.25. [9]

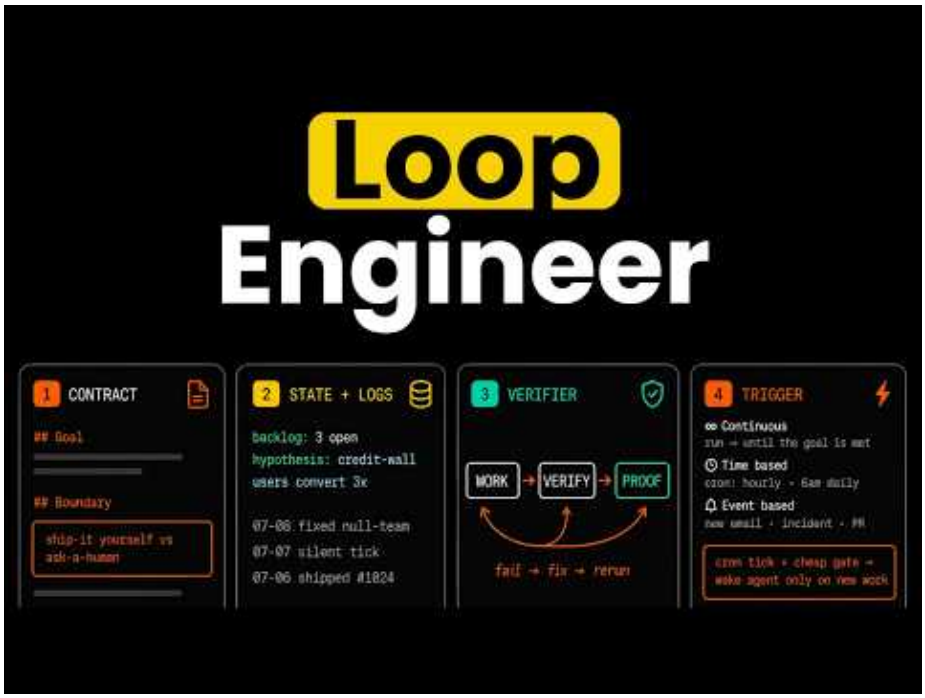
GO DEEPER

- **5:05–7:21 — Peter Steinberger on context-aware auto-review.** Watch the implementation detail that matters: run a separate review, but

feed the result back to the agent that authored the change so it can judge feedback against the real constraints.

[ ] OpenClaw Peter Steinberger AI (5:05)

- **7:56–9:34 — Jason Zhou on the three-role loop.** A concise production pattern: an orchestrator plans, isolated worktree executors run in parallel, and a verifier attaches test evidence for a human to inspect.



What I learnt after running loops for 1 month??? (7:56)

- **Study the Loopany repo.** Its useful contribution is not another agent wrapper—it makes the contract, trigger, state, logs, and iterative improvement cycle first-class artifacts. Open the repo [2, 1]

Editorial take: agent autonomy is earned through explicit boundaries, durable state, and verifiable evidence—not by indiscriminately adding more agents. [1]

Sources

1. What I learnt after running loops for 1 month???
2. X post by @jasonzhou1993
3. [ ] OpenClaw Peter Steinberger AI
4. This is absolute chaos…
5. I can't believe they released this
6. X post by @antigravity
7. X post by @ClaudeDevs
8. X post by @LangChain
9. datasette code-frequency chart on GitHub