Loop Engineering, Record & Replay, and New Automation Primitives

Coding Agents Alpha Tracker

2026-06-19

Loop Engineering, Record & Replay, and New Automation Primitives

By Coding Agents Alpha Tracker • June 19, 2026

The strongest coding-agent signal today is the shift from manual prompting to durable loops. This brief covers the concrete workflows behind self-driving PRs, shared-state agent harnesses, and the latest releases from Codex, Cursor, Claude Code, LangSmith, and Datasette.

TOP SIGNAL

The clearest shift today is from manual prompting to loop design. Theo showed Codex clearing stale PRs overnight and waking up to four stacked PRs reviewed and merged [1, 2], Jason Zhou described support and SEO loops already running in production on 30-minute and daily cadences [3], and Steve Yegge's write-up of Ezra Savard's Netflix study treats single-agent and multi-agent use as distinct literacy jumps with dedicated training for each [4]. The common pattern across Addy Osmani and Geoffrey Huntley: the advantage is a harness that can sleep, checkpoint state, recycle context, and use a separate evaluator—not a better one-shot prompt [5, 6].

TRY THIS

- **Run a repo-maintainer loop instead of a cleanup sprint.** Steipete's exact pattern is: tell Codex to maintain your repos, wake every 5 minutes, and direct work to threads; back it with an orchestrator plus triage, autoreview, and computer-use skills [7]. Theo's concrete use: let the loop close useless stale PRs, revive the worthwhile ones, then give each revived PR one build thread and one review thread; if you're pushing a big migration, he also bumped Codex subagent parallelism from 3 to 20 and set

a sharply defined goal [1, 8]. Study the exact skill docs here: maintainer-orchestrator and github-project-triage [7].

- **Move PR review handling off your keyboard.** Theo's next step was giving a PR its own worktree on another machine, then telling the agent to watch for comments, address them, and keep going; one run kept working for 6+ hours [2]. After the code lands, have the agent run the dev server, verify behavior, commit, push the PR, fetch review comments itself, and even spin up reviewer threads; his dynamic loop created PRs, re-reviewed each new SHA, merged, and triggered the next PR automatically [2]. Watch token burn on bad branches: Theo saw one feedback loop chew through 3M+ tokens on a small set of comments [2].
- **Turn a good one-off run into a shared-state loop.** Jason Zhou's setup flow is practical: manually run the task once, calibrate the behavior, then ask the agent to create a README contract with the goal, workflow, timeline, and schema before wiring a recurring trigger [3]. Put outputs into shared folders for artifacts, signals, and tasks so other loops can read/write the same state, and add a global `worklog.md` so each agent reads the last 5-10 entries before starting [3]. Triggers can be cron jobs, webhooks, or other agents [3].
- **Split planner / builder / reviewer at both the agent and model layers.** Addy Osmani's minimum bar for long-running agents is true sleep via events, durable checkpoints on every transition, and a separate evaluator because self-review overrates quality [5]. Matthew Berman's concrete implementation is model routing as a skill: plan with Fable, write with Composer, then review with GPT-5.5 [9]. Geoffrey Huntley's simpler orchestrator constraint is also worth stealing: allow one task only, recycle the context window after each task, and progress state with git commits plus a todo list [6].

WHAT SHIPPED

- **Codex — Record & Replay.** OpenAI shipped a new primitive for teaching Codex by demonstration: record a recurring task once, stop recording when you want, and Codex turns the session into an inspectable, editable skill [10]. Greg Brockman framed it as teaching Codex by demonstration, and Nick Baumann says he's already using it for calendar formatting, PR-to-Slack posting, and onboarding-flow testing [11, 12].
- **Cursor — `/automate` + new triggers.** Cursor added a plain-language `/automate` skill that configures triggers, instructions, and tools for you, plus Slack emoji triggers, GitHub triggers for issues/reviews/workflow runs, and computer use for cloud agents [13, 14, 15]. Changelog: cursor.com/changelog/06-18-26 [15].
- **Claude Code — Artifacts (beta).** Team and Enterprise users can turn a session into an interactive page like a PR walkthrough or living project

dashboard, then share it via private link [16]. Boris Cherny says he's using it for visual explanations of tricky code, system diagrams, animation previews, and shared dashboards; Mike Krieger's tip is to ask Claude to diagram its work as tasks get deeper and more independent; @_catwu says teams are already using it to share architecture changes, analyses, and prototypes [17, 18, 19].

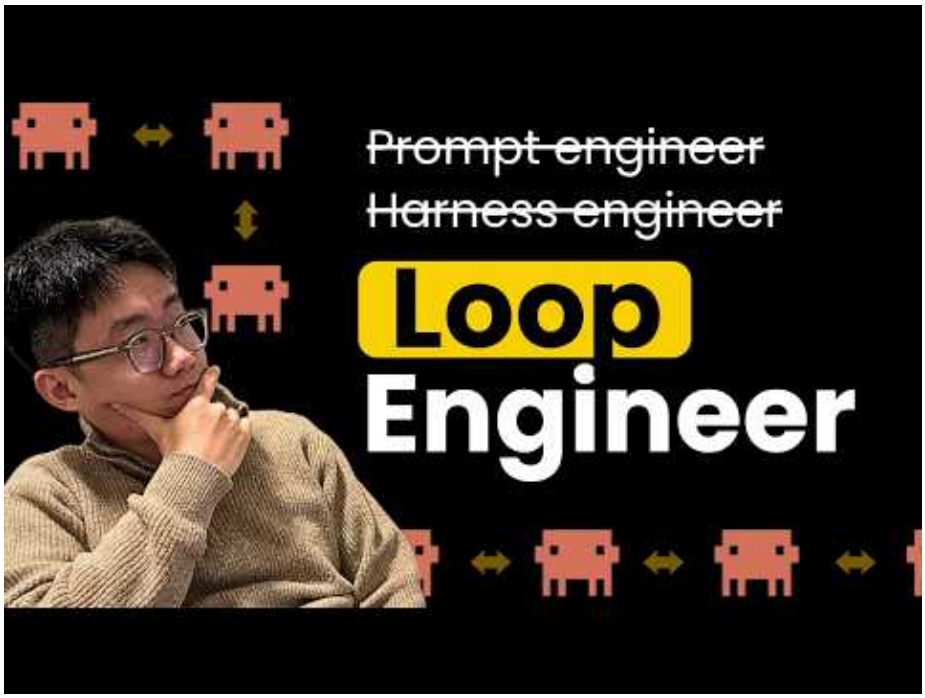
- **LangSmith — LLM Gateway.** LangChain launched a gateway positioned as a budget guardrail against agents burning through large LLM bills overnight [20]. Link: Introducing LLM Gateway [20]. Timely context: Theo said his Codex loops drove more than $20,000 in inference over 48 hours [21].
- **Datasette Agent / Datasette Apps.** Simon Willison's latest write-up shows a coding-agent workflow that's unusually clean: describe an app in chat, let the agent call `describe_table`, then `app_create`, and generate a single-file HTML app against a constrained API [22]. His build stack is also a useful comparison point: Claude Opus 4.6 for the first plugin, Codex Desktop + GPT-5.5 for planning, and Claude Fable 5 for security review—which caught a real CSP privilege-escalation issue [22].
- **GLM-5.2.** Simon notes the 753B MoE model has a 1M context window, open weights under MIT, ranks #2 on the Code Arena WebDev leaderboard behind only Claude Fable 5, and is listed on OpenRouter around $1.40 / $4.40 per million tokens input/output [22]. In his testing it did especially well on animated SVG output, though one more complex illustration regressed versus GLM-5.1 [22].

GO DEEPER

- **12:28-13:26 — Theo on loops that create more loops.** Short demo of the agentic endgame: one thread makes the PR, another reviews each new SHA, fixes get re-reviewed, then the PR merges and the next one starts [2].

I guess we're writing loops now? (12:28)

- **18:24-19:29 — AI Jason on the handoff from manual run to production loop.** He shows the exact move most people skip: test the workflow once, then make the agent write a README contract and wire the recurring trigger around it [3].



After spent 30+ hrs building loops… (18:24)

- **1:03-3:17 — Addy Osmani on why long-running agents fail.** Compact explanation of the three requirements: event-driven sleep, durable checkpoints, and a separate evaluator instead of self-grading [5].
- **1:33-2:29 — Geoffrey Huntley on Ralph loops.** Good antidote to the `while true` meme: single-task constraint, context recycling, and state progression via git commit + todo list [6].
- **Read Steve Yegge's Netflix training note:** The Flat Curve Society. Useful if you're rolling agents out to a team: 0M / 4M / 12-15M qualified-day token cohorts, team-based training, and the shift from raw spend metrics to waste reduction and pocket evals [4].
- **Study the exact skills behind the maintainer loop:** maintainer-orchestrator and github-project-triage. These are the concrete skill docs steipete says he combines with triage, autoreview, and computer use so work can land autonomously [7].
- **Study Datasette Agent + the Datasette Apps article.** It's a strong example of an agent with explicit tools, constrained APIs, and a copyable prompt template that other models can reuse [22].

Editorial take: the winners are starting to look less like prompt whisperers and more like workflow engineers with budgets, checkpoints, and reusable state [3, 5, 20].

Sources

1. X post by @theo
2. I guess we're writing loops now?
3. After spent 30+ hrs building loops…
4. The Flat Curve Society
5. 3 patterns to build long-running AI agents
6. Dark Factories, Cargo Cult AI, and Drunk Agents with @GeoffreyHuntley
7. X post by @steipete
8. X post by @theo
9. I figured out the best way to vibe code
10. X post by @OpenAIDevs
11. X post by @gdb
12. X post by @nickbaumann_
13. X post by @cursor_ai
14. X post by @cursor_ai
15. X post by @cursor_ai
16. X post by @claudeai
17. X post by @bcherny
18. X post by @mikeyk
19. X post by @_catwu
20. X post by @LangChain
21. X post by @theo
22. Datasette Apps: Host custom HTML applications inside Datasette