Persistent Agents Arrive; Merge Gates Decide What Ships

Coding Agents Alpha Tracker

2026-09-11

Persistent Agents Arrive; Merge Gates Decide What Ships

By Coding Agents Alpha Tracker • September 11, 2026

Cursor Projects and OpenAI's Agents API make persistent, cloud-running coding work concrete; the practical differentiator is now the evaluation and review loop that makes autonomous changes safe to merge.

TOP SIGNAL

Coding agents are becoming durable workers instead of disposable chats. Cursor's Projects beta keeps a coordinator in a single persistent thread, delegates to subagents, and supports scheduled tasks, PR follow-up for CI fixes, Slack bug monitoring, and shared memory/artifacts across the user's device and agents' computers. [1, 2, 3, 4] OpenAI's Agents API is in public beta and brings the Codex harness to developers: one call specifies the task, model, tools, and environment, while the platform supplies managed sandboxing, long-session compaction, tool search/programmatic calls, and parallel subagents. [5]

The practical shift is architectural: make project state—memory, schedules, tools, artifacts, and isolated execution—the durable unit of work instead of rehydrating context in every new chat. [3, 5]

TRY THIS

- **Put background work in one persistent project.** In Cursor Projects, keep the coordinator thread as the source of truth. Give it a scheduled task such as: `Inspect open PRs and recent Slack bug reports; reproduce actionable CI failures; update the shared plan with status, evidence, and next owner.` Leave plans and demos in the project's shared artifacts so the next agent starts with state,

not a blank prompt. Cursor explicitly supports the scheduling, PR/Slack, subagent, memory, and artifact pieces of this loop. [1, 2, 3]

- **Replace manual trace review with an online judge.** Install the current LangSmith skills, ask a coding agent to use the LangSmith CLI to create an evaluator for incoming traces, and define a tight rubric—in the demo, `1` means a frustrated user and `0` means not frustrated, with reasoning attached to the trace. Validate the judge on real runs, inspect the standout cases, then lower sampling after it is trustworthy; the walk-through reviews 13 runs and changes the evaluator to 50%, which the presenter says should halve cost. [6]
- **Turn regression criteria into repo tests.** Run `mda evals init -i` to have Claude Code scaffold Harbor-backed evals, but require a human to approve each generated `task.md`. Keep the environment/job/check structure explicit, then put assertions in `test.sh`/pytest for behaviors such as citing only real documents, admitting when the corpus is insufficient, using the required format, and returning the right status codes. Run the suite nightly in CI and during development; LangSmith can compare changes in models, tools, and tool descriptions over time. [7]
- **Set a higher bar for production agent code than prototypes.** Boris Cherny's split is useful: low-blast-radius throwaway code can be treated as a black box, while production code needs linting, tests, end-to-end tests, daily fuzzing, and automated code/security review; when it misses the bar, raise effort and improve concise repo-specific `CLAUDE.md` instructions and skills. For security work, copy Datasette's separation of duties: one human writes the automated vulnerability test, another implements the fix, and both humans review the result alongside agents running different models. [8, 9]

WHAT SHIPPED

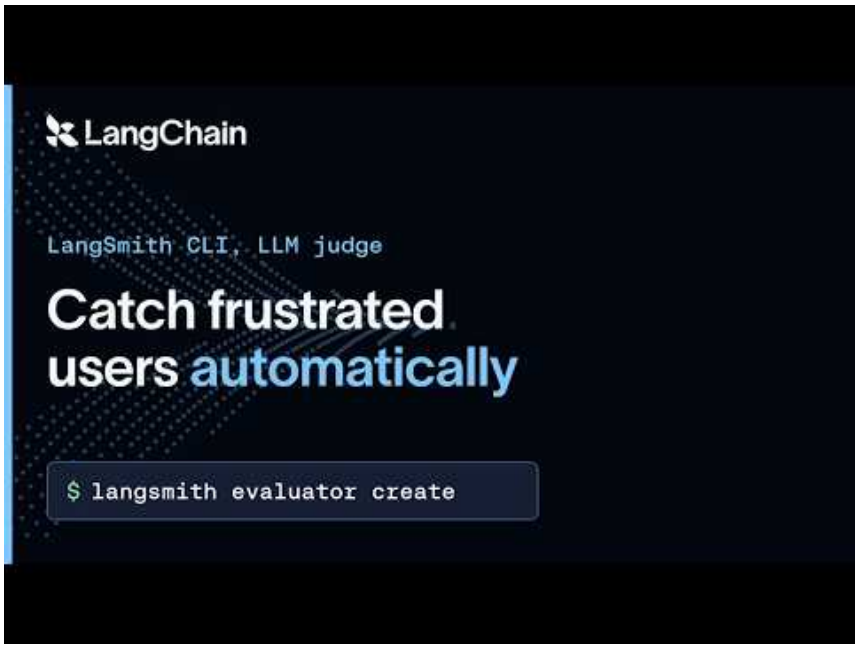
- **OpenAI Agents API — public beta.** Developers can choose an OpenAI-managed sandbox, their own infrastructure, or a sandbox partner; the hosted environment can be configured with files, packages, skills, and plugins. The harness automatically compacts context for long sessions and supports tool search, MCP, custom functions, built-in tools, and programmatic parallel/chained calls. [5]
- **Cursor Projects — beta rollout.** The persistent coordinator can manage subagents, scheduled work, PR/CI follow-up, and Slack bug intake; project agents share memory and generated artifacts that sync across devices. [1, 2, 3, 4]
- **Managed Deep Agents + Harbor.** LangChain's release packages evals in fresh containers and tracks them in LangSmith, reducing test-environment pollution when agents touch files or run external commands.

[10, 7]

- **Cognition SWE-2.** Cognition claims the model is on par with recent frontier models on leading evals at up to 70% lower cost. Treat that as a vendor claim to reproduce on your own repository, not a universal ranking. [11]
- **Model routing is separating "deep" from "mergeable."** In Theo's comparison, Fable 5.1 averaged two additional follow-ups from filed PR to merge versus six for Astra on projects serving hundreds of thousands of users. His rule: use Fable for focused fixes, UI changes, features, and performance work; use Astra for harder problems where exhaustive assumption-testing is worth the extra time and tokens, then hand an Astra death loop to Fable to land cleanly. [12]
- **Datasette 1.0a39 and 0.65.4 security releases.** Simon Willison says public Datasette instances—especially those mixing public and private tables—should upgrade; the fixes followed a multi-model audit with Claude Fable 5.1, GPT-5.6, and GPT-6 Astra that found subtle bugs. [9]
- **Credit Genie's OpenWiki loop.** Its coding agents stopped guessing how repositories work and now consult OpenWiki; new repos are onboarded through a stub PR, nightly commits trigger update PRs, and a daily sweep merges pending work and rebuilds the portal. [13, 14]

GO DEEPER

- **Score Every Production Trace with an LLM Judge, from Your Terminal** — The most immediately reusable walkthrough: install agent skills, build a rubric-backed evaluator with the CLI, inspect its reasoning on real traces, and tune sampling after validation. [6]



Score Every Production Trace with an LLM Judge, from Your Terminal (LangSmith CLI) (1:24)

- **Catch Agent Regressions Before You Ship** — Watch the `mda evals init -i` flow, the human approval checkpoint for generated specs, and the nightly LangSmith loop. [7]

Catch Agent Regressions Before You Ship: Evals for Managed Deep Agents

(3:42)

- **Fable Vs Astra Debate Is Over** — Skip to the mergeability comparison. The useful lesson is not a leaderboard winner; it is the routing rule based on rework after PR filing, depth of investigation, and whether the agent is stuck. [12]

Fable Vs Astra Debate Is Over (33:34)

- **Credit Genie's OpenWiki case study** — Study the maintenance loop that turns repository knowledge into an automatically refreshed surface for coding agents instead of a document someone must remember to update. [13, 14]
- **Datasette security-release workflow** — A compact example of combining frontier-model audits with two-human test/fix separation before shipping security patches. [9]

Editorial take: The durable advantage is shifting from the clever prompt to the runtime around it—persistent state, isolated execution, model routing, and tests that make autonomous changes inspectable and mergeable. [5, 7, 9]

Sources

1. X post by @cursor_ai
2. X post by @cursor_ai
3. X post by @cursor_ai
4. X post by @cursor_ai

5. Introducing the Agents API | OpenAI
6. Score Every Production Trace with an LLM Judge, from Your Terminal (LangSmith CLI)
7. Catch Agent Regressions Before You Ship: Evals for Managed Deep Agents
8. X post by @bcherny
9. Datasette 1.0a39 and 0.65.4 security releases
10. X post by @LangChain
11. X post by @cognition
12. Fable Vs Astra Debate Is Over
13. X post by @LangChain
14. X post by @LangChain