

Goal Loops Take Over Coding Agents; Google Ships Managed Agents and Cursor Adds /loop

Coding Agents Alpha Tracker

2026-06-04

Goal Loops Take Over Coding Agents; Google Ships Managed Agents and Cursor Adds /loop

By Coding Agents Alpha Tracker • June 4, 2026

Goal-based coding agents are converging across OpenAI, Google, Cursor, and Microsoft. This brief covers the copyable workflows behind that shift, what shipped today, and the strongest anti-hype lessons from engineers who benchmark agent output against expert baselines.

TOP SIGNAL

The biggest practical shift today: **coding agents are standardizing around goal loops, not chat turns**—Romain Huet shows Codex’s goal flow as one ambitious task plus a verifiable completion condition that can run for hours or days [1], while Google’s managed-agents team says Gemini’s Interactions API had to become agent-first because real agents do tool calls, sub-agents, and continuous steps rather than simple user/model turns [2]. Jediah Katz’s new Cursor /loop skill and Satya Nadella’s note that Copilot now needs UI for **100+ concurrent agent sessions** make the same point from the product side: the valuable skill is increasingly orchestration—wake-up conditions, state, and review—not just faster prompting [3, 4]. Also: don’t confuse autonomy with quality—Mitchell Hashimoto’s loop found a big local win, but his handwritten baseline was still ~75x better [5], and Alexander Embiricos’s **bring the taste** principle is the right operating model for serious work [6].

“AI can write your code... But it cannot care... You have to bring the taste.” [6]

TRY THIS

- **Batch a whole class of work with one hard done-condition (Romain Huet / Codex).** In Codex, type `goal`, then phrase the task as a final state the agent can verify: `pull all of the bugs from the backlog from yesterday's launch and prepare a PR for each of them and make sure all of the tests pass`. OpenAI says goal mode is meant for tasks that run for hours or even days; Huet shows the same pattern on a large migration with `migrate this entire code base to Java 26` plus the requirement that everything keeps going until tests pass [1].
- **Add a wake-up loop for waiting tasks (Jediah Katz / Cursor).** Cursor can now watch terminal output and take action; Katz used that to build a public `/loop` skill. Copyable prompts: `/loop until this PR merges` and `/loop 1h check #infra-logs for anything critical`. Caveats: it does **not** work in Cloud Agents yet, and it will not fire while your computer is sleeping [3, 7].
- **Constrain optimization agents like you would a junior performance engineer (Mitchell Hashimoto).** His RALPH loop was basically `while not done: try again`, but with explicit no-go zones: the agent could not modify input data structures, the public API, or tests [5]. That still produced a large improvement—**88ms -> 1.5-2ms** and **150k allocations -> 500** [5]—but the real move is the second pass: benchmark against an expert baseline before you call the result great, because Hashimoto’s handwritten version was still far better [5].
- **Move agent definition into files, not ephemeral chat state (Google Managed Agents / Anti Gravity).** Google’s current workflow is plain markdown: an `agents.md` file for how the agent should work, plus separate markdown skill files [2]. Pair that with agent-first docs in markdown and the MCP server, then call the agent through the Interactions API—or start in AI Studio and one-click export into Anti Gravity once the project hits real-codebase territory [2, 8].

WHAT SHIPPED

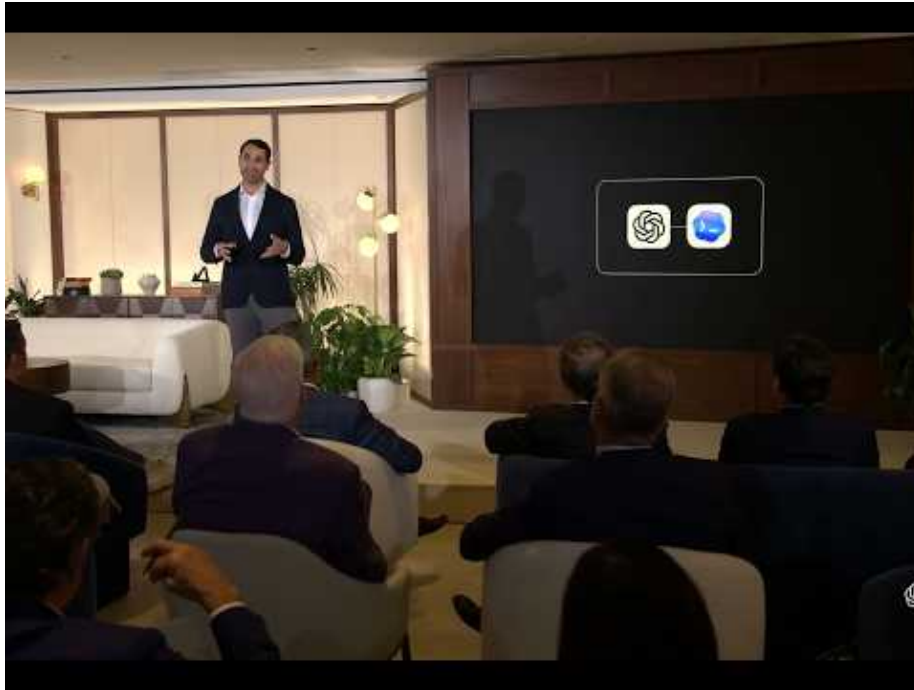
- **Google — Managed Agents in the Gemini API.** One API call spins up an autonomous agent in a remote Linux sandbox that can write code, run Bash, and create files; the launch stack is Gemini 3.5 Flash plus the Anti Gravity agent harness [2].
- **Google — Anti Gravity now spans IDE, CLI, SDK, and API.** Google positions it for agentic engineering on very large codebases with guardrails, not just quick prototypes [8].
- **Google — Interactions API is the unifying layer for models + agents.** The same interface can call models and managed agents; the data

model is now agent-first, with tool calls, sub-agents, and continuous step streams instead of turn-based chat [2].

- **OpenAI — Codex / GPT 5.5 goes deeper into full-lifecycle engineering.** Goal mode can run ambitious software tasks for hours or days with verifiable completion conditions [1]; Cisco says Codex is already being used for new code and legacy migrations that used to take months and now take weeks [1]. The same demo showed 6.5-hour full-codebase security scans with inline P0 findings, appshot/computer-use testing that drives the app without taking over the user’s machine, and automatic engineering context pulled from tools like Databricks [1].
- **Cursor — terminal-watching agents are public.** The /loop skill is available now, making scheduled or output-triggered wake-ups a practical local workflow; still no Cloud Agents support, and sleep mode stops it [3, 7].
- **LangChain — LangSmith Sandboxes GA.** LangChain’s framing is exactly the coding-agent requirement: stateful little computers where agents can install packages, edit files, follow long-running threads, resume later, and run untrusted code safely by default [9]. Announcement: langchain.com/blog/langsmith-sandboxes-generally-available [9]
- **Sourcegraph — MCP server for Copilot context.** GitHub Copilot can be connected to the Sourcegraph MCP server to pull context from all repositories, including code that lives in GitLab [10].
- **Microsoft / GitHub Copilot — agent scale is reshaping both UI and pricing.** Satya Nadella says coding usage has grown to the point where the IDE now has to manage **100+ agent sessions**, which is why chat alone no longer works and a canvas UI is needed [4]. He also says Copilot had to move away from pure per-user economics because long-running agent workloads are much more intense than classic code-complete usage [4].
- **Microsoft — the harness is becoming the product.** Nadella describes a GitHub harness that loops models, data, and tools, uses rich context prep plus multimodal tool access for efficiency, and can be tuned with private evals; he says the same harness is used across products and is available in Foundry [4].

GO DEEPER

- **37:04-38:03 — Romain Huet on Codex goal mode.** Best short explainer of the new interaction pattern: ambitious goal, explicit done-condition, then let the agent run [1].



OpenAI Codex will merge into ChatGPT: Denise Dresser, Alex Emibiricos, Romain Huet, Sam Altman (37:04)

- **5:59-6:40 — Google on why agent APIs can't stay turn-based.**
Useful mental model if you're building your own harness: the real unit is a stream of tool, function, and sub-agent steps—not one chat reply [2].

Managed Agents in the Gemini API



 Google AI Studio

Managed Agents in the Gemini API (5:59)

- **7:06-7:37** — **Satya Nadella on the 100 agent sessions problem.** Short clip, big signal: once agents run in parallel, chat-only IDE UX breaks [4].



Satya Nadella on AI: @NoPriorsPodcast x Latent Space Crossover Special at Microsoft Build 2026 (7:06)

- **34:31-35:41** — **Mitchell Hashimoto's optimization loop, with the anti-hype payoff.** Watch this for a concrete example of constraint-driven agent search—and why a huge gain can still be nowhere near the real ceiling [5].



Layoffs are getting Wild (ft. Big A) (34:31)

- **Study:** LangSmith Sandboxes GA is the cleanest short writeup in today's set on the execution model serious coding agents need: stateful environments, resumability, and safe untrusted-code execution. langchain.com/blog/langsmith-sandboxes-generally-available [9]
- **Study:** Sourcegraph's Copilot + MCP demo is worth a quick pass if your pain point is cross-repo context, especially in mixed GitHub/GitLab setups [10].

*Editorial take: the edge is moving from prompt cleverness to operational discipline—clear goal conditions, stateful sandboxes, wake-up loops, and humans who know when **better** still isn't good enough [1, 9, 3, 5].*

Sources

1. OpenAI Codex will merge into ChatGPT: Denise Dresser, Alex Emibiricos, Romain Huet, Sam Altman
2. Managed Agents in the Gemini API
3. X post by @jediahkatz
4. Satya Nadella on AI: @NoPriorsPodcast x Latent Space Crossover Special at Microsoft Build 2026
5. Layoffs are getting Wild (ft. Big A)

6. X post by @jipvanderfelde
7. X post by @jediahkatz
8. How Google wants to turn prompts into companies
9. X post by @LangChain
10. X post by @Sourcegraph