

Persistent Agent Loops Become the Workflow

Coding Agents Alpha Tracker

2026-05-20

Persistent Agent Loops Become the Workflow

By Coding Agents Alpha Tracker • May 20, 2026

The highest-signal shift today is operational: practitioners are moving from one-shot prompting to scheduled loops, test-heavy execution, and runtime audits. This brief covers swyx's AI SDLC, Boris Cherny's /loop playbook, LangSmith Engine, Antigravity's new agent stack, DeepAgents updates, and Cursor's Jira handoff.

TOP SIGNAL

- The strongest signal today is operational, not model-level: top users are turning coding agents into a standing layer of engineering work. Boris Cherny said Anthropic has reached a point where code is no longer written manually and that he keeps 5-10 active sessions with ~200 agents running plus thousands overnight via /loop and server-side routines, while Mike Krieger said internal code generation is approaching 100% and engineers are increasingly managing agents instead of hand-writing code; swyx's 4-step AI SDLC is the clearest copyable control loop behind that shift—tests first, plan/refactor hot paths, force completion with periodic deploy/test, then spot-check prod and steer bugs [1, 2, 3].

TRY THIS

- **Copy swyx's 4-step AI SDLC into your agent repo.** Keep ~50 tests, tell the agent to add more, then plan refactors, force completion, and keep spot-checking after deploy [3].

whenever you do browser e2e tests, use computer vision to visually spot check design and ux issues as well on mobile/desktop/ipad/ultrawide resolutions [3]

/plan break up & edit hot paths so you isolate files for easier editing and reading. add proper logging and error boundaries/handling

while you do it. what else should we refactor for maintainability/performance/ai editing? [3]

you can break backward compatibility. first map out all the remaining work, then proceed on this next slice, do not stop until all work is done, periodically stop to commit, deploy and test but do not stop until all work is done [3]

- **Use recurring agent loops for chores you already do every day.** Boris Cherny's `/loop` schedules repeated tasks through Chrome at whatever cadence you want—every minute, every 5 minutes, daily—and he uses it for PR babysitting, auto-rebase, fixing CI, cleaning up flaky tests, and aggregating feedback every 30 minutes [1]. If you want the work to continue after the laptop closes, move the same pattern into server-side routines [1].
- **Close the runtime feedback loop instead of pasting errors back into chat.** Google's new Chrome DevTools for Agents lets the agent run Lighthouse itself, read the report, propose a fix, and rerun audits; the AgentIQ browsing category checks WebMCP registrations, form metadata, the LLMs Txt file, and accessibility signals agents depend on [4, 5]. Practical pattern: give the agent direct runtime observation instead of using a human as the clipboard [4, 5].
- **For CI/autofix workflows, force the model to emit exactly the artifact your pipeline needs.** Kevin Hou's demo starts with a broken CI stack trace and a simple "fix this" request to generate a remediation bash command, then fine-tunes Gemma 4 via LoRA on prompt→command examples so the output is just the command instead of explanatory text; he uses voice input, approves the implementation plan, kicks training to a GPU VM via CLI, sanity-checks logs, and deploys the fine-tuned model [5, 4]. If chatty outputs are breaking automation, this is the practical fix: train or constrain for machine-consumable responses [5, 4].

WHAT SHIPPED

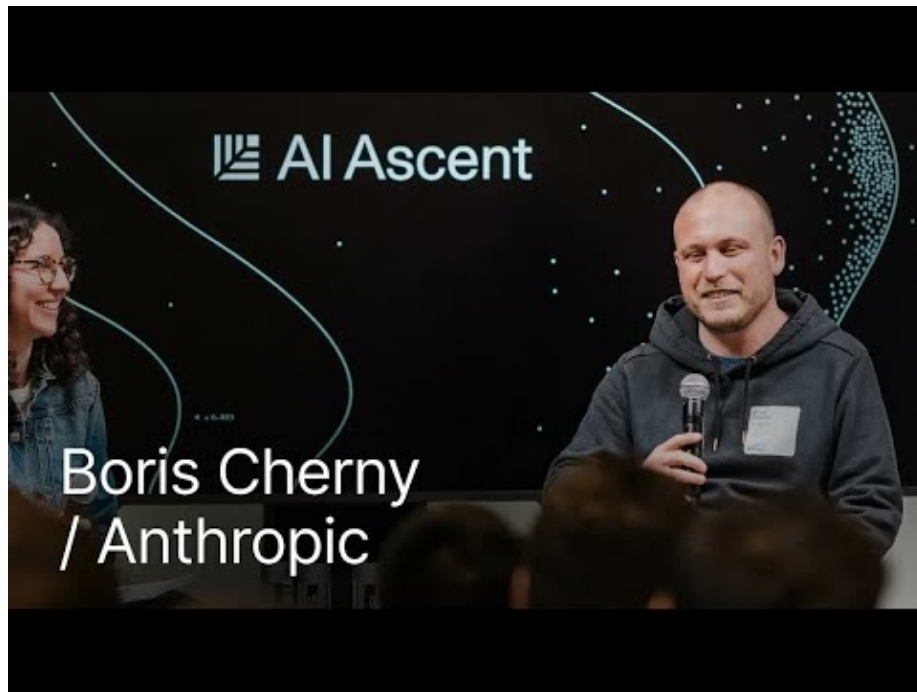
- **LangSmith Engine:** autonomously finds failure patterns in agent traces, clusters them into named issues, drafts code fixes, and proposes eval coverage; it watches tool failures/timeouts, eval failures, anomalies, negative feedback, and unusual behavior, and generates evaluators once fixes are confirmed. Blog: LangSmith Engine [6, 7, 8].
- **Google Antigravity 2.0 + CLI:** new desktop app for multi-agent teams, scheduled tasks, native voice, and one-click integrations, plus a CLI with the same harness/models but a terminal-native UX. Download: antigravity.google / blogs: 2.0 and CLI [9, 10, 11, 12, 13, 14].
- **Gemini API managed agents:** one API call gives you an agent plus a remote Linux sandbox; Google also showed Markdown-defined skills, one-

click AI Studio → Antigravity export, and a Stytc production example that connects to GitHub and emits a design MD file from the codebase [4].

- **Google’s agent-support toolchain got much more concrete:** Modern Web Guidance claims an average **+37 percentage-point** pass-rate lift for guided vs unguided web coding; Chrome DevTools for Agents adds runtime audit/fix/re-audit loops; Android CLI skills + knowledge base cut tokens by about **70%** and complete tasks up to **3x faster** in Google’s internal tests [5, 4].
- **LangChain / LangSmith stack:** DeepAgents code shipped as an open-source example for production coding agents on open models; DeepAgent 0.6 adds a QuickJS-based code interpreter; LangSmith Sandboxes are now GA with persistent/forkable environments and an auth proxy; Context Hub stores agent MD files, skills, and LLM wikis with versioning; LLM Gateway beta adds spend controls plus PII/secrets guardrails; Managed DeepAgents entered private preview [15].
- **Cursor Jira:** assign Cursor to a work item or mention **@Cursor** and it spins up a cloud agent that reads the title, description, comments, and repo settings to produce a merge-ready PR; tasks can include bug fixes, features, tests, or investigation. Changelog: cursor.com/changelog/05-19-26 [16, 17].
- **Early model comparison worth watching:** Gemini Flash 3.5 landed on CursorBench at cursor.com/evals [18], while Theo said its early result scored below Composer 2/2.5 and cost 4x more on that eval [19]. Separately, Antigravity says it is serving Gemini 3.5 Flash **12x faster** in its own product for coding workflows [20].

GO DEEPER

- **07:57-08:14 — Boris Cherny on /loop.** Short, concrete explanation of the scheduling primitive behind his always-on workflow; pair it with the examples of PR babysitting, CI repair, and feedback aggregation and you can copy the pattern immediately [1].



Boris Cherny (Anthropic): Programação Resolvida? O Futuro do Código com IA (7:56)

- **51:14-53:22** — **Chrome DevTools for Agents closes the build → audit → fix loop.** Best clip of the day if you want to see an agent run Lighthouse, inspect its own report, and rerun validation without a human copying errors around [4].

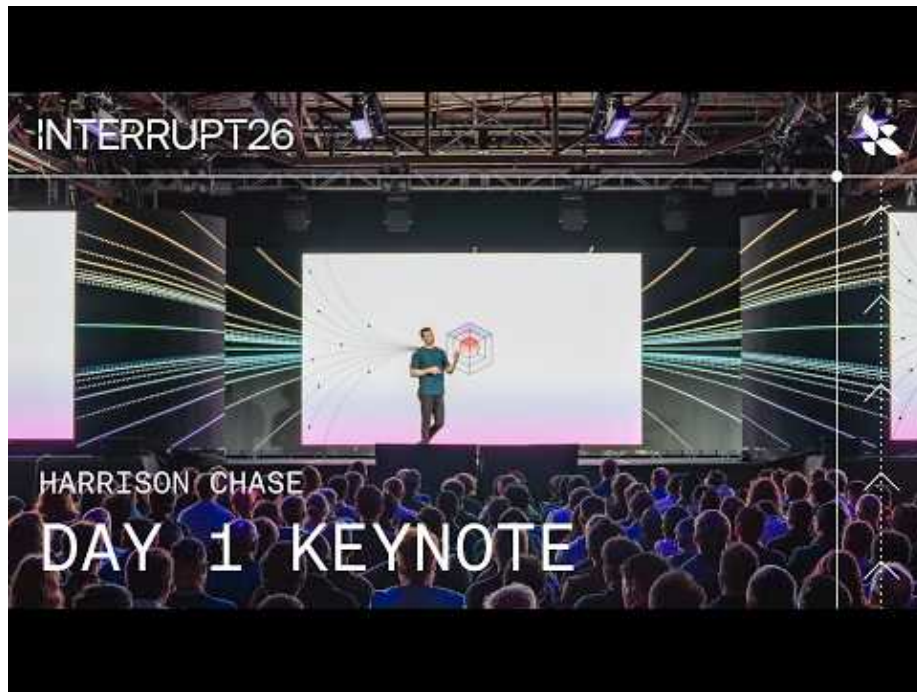
Google I/O

2026 Developer keynote



Developer Keynote (Google I/O '26) (51:13)

- **15:01-15:54 — LangSmith Sandboxes GA.** Harrison Chase lays out the case for persistent, forkable sandboxes with an auth proxy so agents can use API-keyed tools without ever seeing the keys themselves [15].



The Agent Development Lifecycle: Build, Test, Deploy, Monitor / Interrupt 26 (15:00)

- **21:30-22:47** — **Managed DeepAgents architecture.** Good higher-level watch if you care about the full production stack: harness, deployments, sandboxes, Context Hub memory, MCP connections, and UI streaming behind one API [15].



The Agent Development Lifecycle: Build, Test, Deploy, Monitor | Interrupt 26 (21:29)

- **Worth studying:** *DeepAgents code* is LangChain’s open-source example of how to build a production coding agent on top of DeepAgents, especially if you care about open models and execution-environment design [15]. *LangSmith Engine* is worth reading end-to-end if your bottleneck is trace review → fix → eval coverage rather than raw model quality [6, 21].

Editorial take: the edge is moving to control surfaces—tests, scheduled loops, runtime feedback, and human steering [3, 1, 4, 2].

Sources

1. Boris Cherny (Anthropic): Programação Resolvida? O Futuro do Código com IA
2. A barreira entre ter uma ideia e construir um produto acabou
3. X post by @swyx
4. Developer Keynote (Google I/O '26)
5. Developer Keynote (Google I/O '26) - American Sign Language
6. X post by @LangChain
7. X post by @LangChain
8. X post by @LangChain
9. X post by @antigravity

10. X post by @antigravity
11. X post by @antigravity
12. X post by @antigravity
13. X post by @antigravity
14. X post by @antigravity
15. The Agent Development Lifecycle: Build, Test, Deploy, Monitor | Interrupt 26
16. X post by @cursor_ai
17. X post by @cursor_ai
18. X post by @mntruell
19. X post by @theo
20. X post by @antigravity
21. X post by @LangChain