

Multi-Agent Coding Becomes the Default Workflow

Coding Agents Alpha Tracker

2026-05-21

Multi-Agent Coding Becomes the Default Workflow

By Coding Agents Alpha Tracker • May 21, 2026

The clearest signal today is operational: top practitioners are moving from one-agent chat tabs to multi-agent control surfaces with evals, observability, and safe delegation. Inside: copyable ADK and MCP workflows, Cursor and Claude Code updates, benchmark context on Composer 2.5 vs Gemini 3.5 Flash, and three clips worth your time.

TOP SIGNAL

- **Multi-agent is becoming the default posture.** Boris Cherny says most Claude Code users now run many instances at once and he personally runs ~5 locally plus hundreds or thousands in parallel overnight; Google says Anti Gravity is shifting away from the IDE toward a UI for managing multiple agents, and Railway is designing for thousands of coordinated agents with explicit human intervention points. Anthropic says code written per engineer rose about 250% after Claude Code, which makes this look like an operational shift, not a demo trick. [1, 2, 3]

TRY THIS

- **Use a manager-agent pattern.** Boris Cherny says he no longer writes code directly; he prompts one Claude that prompts other Claudes. Copy the structure, not the scale: give one lead agent the spec, let it coordinate other agent instances, start with a handful of parallel tasks locally, then expand to bigger overnight batches only after the review loop feels safe. [1]
- **Use a zero-to-one build prompt, then gate on evals.** In Kevin Hou's demo, the prompt was `build me a daily news bot using`

adk ... I want RSS feeds ... summarize the five stories ...
I should be able to deploy this and fetch the latest stories.
The agent produced an implementation plan you can comment on, created the scaffolding and task list, then kept running evals/tests while fixing its own mistakes; steal this pattern for small internal tools instead of starting from folders and boilerplate. [2]

- **Debug prod with MCPs and one plain-English prompt.** In Google's demo, once the cloud MCPs were configured, the prompt `find out what's wrong with my DynoQuest app` was enough for the agent to inspect the right services and logs. The trick is boring but powerful: wire the MCPs first, then keep the prompt high-level. [2]
- **Compress long-horizon evals before changing prompts.** Palash Shah's process for a 30+ minute agent run is: extract the reasoning from traces, identify the cause of the behavior, recreate only that minimal situation, then iterate on a much smaller eval. This is the cleanest prompt-debug loop in today's sources. [4, 5]

WHAT SHIPPED

- **Cursor 3.5 automations.** Multi-repo automations can now work across codebases to execute, test, and verify tasks; you can also create repo-less automations for jobs like a daily Slack digest. Automations now live in the Agents Window, new automation runs are 50% off for 7 days, and templates are live at cursor.com/marketplace; download is at cursor.com/download. [6, 7, 8, 9]
- **Claude Code Auto Mode.** Anthropic replaced per-tool permission prompts with layered safety checks where a second Claude evaluates tool use, backed by thousands of safety benchmarks; Boris Cherny says this reduced approval fatigue and was safer than the old prompt-by-prompt flow. [1]
- **Cursor Composer 2.5: fresh benchmark context.** Artificial Analysis puts it at 62 on the Coding Agent Index, third behind Claude Opus 4.7 and GPT-5.5; standard is \$0.07/task, Fast is \$0.44/task and averages 6.7 minutes per task, with availability limited to Cursor IDE and CLI. [10]
- **Gemini 3.5 Flash: strong on Google's coding evals, weak on CursorBench.** Google says Flash improved materially on terminal benchmarks, SWE-bench-style coding, MCP calling, and tool use for agentic coding; Theo points to cursor.com/evals and argues Flash 3.5 scored below Composer 2 there while costing 4x more. Treat this as a benchmark split-screen, not a settled verdict. [2, 11, 12]
- **Antigravity tooling is consolidating.** Google says the old Apache-licensed Gemini CLI will stop working with subscription plans on June 18 and be replaced by Antigravity CLI; Simon Willison notes the broader

suite now spans a desktop app, CLI, IDE, and an open-source Python SDK wrapper. Repos: [google-antigravity/antigravity-sdk-python](#), [google-antigravity/antigravity-cli](#), [google-gemini/gemini-cli](#). [13]

- **CodeMinder is one to watch.** Sundar Pichai says Google’s internal security teams already use agentic workflows to detect vulnerabilities and patch them, and the internal CodeMinder system being externalized can identify issues, generate patches, test them, verify them, and deploy fixes. [14]

GO DEEPER

- **2:24:06-2:28:43** — **Kevin Hou’s zero-to-one ADK build.** Best concrete demo of the day if you want to watch an agent go from a natural-language request to plan, scaffolding, fixes, evals, and a finished artifact.



[2]

From the I/O main stage to the terminal (144:05)

- **1:04:53-1:07:02** — **MCP-powered prod debugging.** Good watch if your agents still depend on humans to paste logs around: the key prompt is almost trivial once the MCP wiring exists. [2]



From the I/O main stage to the terminal (64:53)

- **50:07-51:57 — Railway’s self-deploy loop.** Advanced, but worth the two minutes: put the platform CLI inside a process already running on the platform, let the agent provision what it needs, deploy itself, and throw



away bad copies. [3]
The Agent-Native Cloud: 3M Users, 100K Signups/Wk, Data Centers, &

Death PRs — Jake Cooper, Railway (50:07)

- **Repos worth reading.** `google-antigravity/antigravity-sdk-python` is the cleanest entry point if you want to inspect how Google exposes Anti-gravity programmatically; diffing `google-gemini/gemini-cli` against `google-antigravity/antigravity-cli` is the fastest way to understand the tooling transition Simon flagged. [13]

Editorial take: the model is eating the scaffolding, but the durable edge is ops—evals, observability, and safe multi-agent control. [15, 2, 3]

Sources

1. Claude Code Head Boris Cherny: Insane Growth, Tokenmaxxing, AI Agents' Next Frontier
2. From the I/O main stage to the terminal
3. The Agent-Native Cloud: 3M Users, 100K Signups/Wk, Data Centers, & Death PRs — Jake Cooper, Railway
4. X post by @palashshah
5. X post by @LangChain
6. X post by @cursor_ai
7. X post by @cursor_ai
8. X post by @cursor_ai
9. X post by @cursor_ai
10. X post by @ArtificialAnlys
11. X post by @theo
12. X post by @mntruell
13. Google I/O, Gemini Spark, Antigravity
14. Google CEO: Agents, Open Source, Race to AGI, Cybersecurity, Chips, China
15. The Model Eats the Scaffolding: DeepMind's Logan Kilpatrick & Tulsee Doshi on 3.5 Flash, Omni & More