

Review Gates, Intent Files, and Cursor's Agent-Native Git

Coding Agents Alpha Tracker

2026-06-17

Review Gates, Intent Files, and Cursor's Agent-Native Git

By Coding Agents Alpha Tracker • June 17, 2026

Review guardrails dominated today: Addy Osmani's data-rich case for risk-tiered agent PR review was the clearest signal. Also worth your time: Kent C. Dodds's INTENT.md pattern, Cursor's Origin launch, LangSmith Sandboxes, and a few concrete Cursor workflows.

TOP SIGNAL

- Addy Osmani's **Agentic Code Review** is the clearest practical read of the day: GitClear data says daily AI users generate about **4x** the code for only about **12%** more delivered value, while incidents-to-PR rose **242.7%**, per-developer defects went from **9% to 54%**, and median review duration rose **441.5%** [1]. His answer is not to back away from agents, but to review differently: batch-triage PRs with Claude Code or Codex, use heterogeneous reviewers, tier by blast radius, and keep the merge decision human-owned [1].

“Treat CI as the wall that does not move.” [1]

TRY THIS

- **Turn review into a gated pipeline, not a vibe check.** Addy Osmani's playbook is straightforward: 1) point Claude Code or Codex at a batch of PRs and bucket them into **safe to merge**, **needs work**, and **high-risk**; 2) run **two different** AI reviewers on risky diffs; 3) tier depth by blast radius; 4) refuse review without an intent statement, test output, and a small diff; 5) read rewritten tests first, keep deterministic CI strict, and let a human own merge [1].

- **Shift human effort into the plan, then automate the line-by-line gate.** Kun Chen’s solo workflow: write a detailed plan up front, run **20-30 agents in parallel** for hours, stay on escalation for stuck agents, and gate merges through an automated **No Mistakes** review step. The transferable pattern is simple: human-owned intent before execution, automated verification after execution [1].
- **Add an `INTENT.md` contract to long-lived packages.** Kent C. Dodds has Kody create and maintain an `INTENT.md` file describing package goals, then compare every proposed change against it. If the goal itself changed, the agent should only update `INTENT.md` when the user explicitly wants that change [2].
- **Build internal tools your agent can operate, not just generate.** Riley Brown’s Cursor demo prompt was essentially `make a to do app for me as a creator with a full database... be able to write to this database... make it look like a simple version of Notion, but dark mode`. He used Convex for the DB, let the agent add tasks by natural language, and deployed it with `@vercel` `put this on the internet` [3].

WHAT SHIPPED

- **Cursor Origin** — Cursor is launching code storage and git hosting so teams and agents can host, review, and collaborate on code; swyx/Tomas Reimers highlighted agent-specific features: scalability for agent workloads, API/MCP extensibility, built-in merge conflict resolution, and CI/CD failure resolution. Available this fall; waitlist [4, 5, 6].
- **Cursor at Compile** — Michael Truell said **>95%** of Cursor users now use it primarily as an agent, and agent requests are used about **5x** more than assistive features. He also described Cursor 3 capabilities around gesture-based design edits, recursive sub-agents, days-long remote project handoffs, and broader SDK/CLI/plugin extensibility [7].
- **AI reviewer comparison got sharper** — CodeRabbit topped the Martian benchmark on F1; Greptile was cited at about **82%** bug-catch versus CodeRabbit’s **44%** in one benchmark; Anthropic said its internal Code Review had **<1%** incorrect findings and raised substantive reviews from **16% to 54%**. The operational takeaway from Addy’s roundup: reviewer diversity matters, because in one 146-PR test **93.4%** of flagged locations were unique to a single tool [1].
- **LangSmith Sandboxes** — LangChain positioned this as the right layer when an agent needs to *do* something: verify generated code runs before responding, operate on real files, persist state across tool calls, scale bursty parallel evals/RL, or safely handle user input that may be executed. Blog [8, 9].

- **GLM 5.2 in Cursor via OpenRouter** — Riley Brown shared the exact setup: paste an OpenRouter key into Cursor's OpenAI API override, set the base URL to `https://openrouter.ai/api/v1`, then add custom model `z-ai/glm-5.2`. Context from Kalo: people he trusts were reporting strong results from GLM 5.2 [10, 11].

GO DEEPER

- **12:00-14:30** — **Riley Brown's agent-writable internal app demo.** Good clip if you want a concrete pattern instead of a slogan: prompt the app into existence, attach a database, let the agent write into it, then verify the state persists [3].



SpaceX Just Bought Cursor for \$60B. It's About to Take OVER. (12:00)

- **16:39-18:52** — **Codex/Claude -> Cursor skills handoff.** Watch this if tool-switching friction is your blocker: Riley exports skills and memory into a Codex Import folder with a README and Needed Keys, then asks Cursor to import it globally [3].



SpaceX Just Bought Cursor for \$60B. It's About to Take OVER. (16:39)

- **8:13-9:11** — **Michael Truell on the next agent handoff shape.** Short but high-signal: the target state is not three local agents for 30 minutes, but handing out whole projects and getting back completed, tested work days later [7].



Cursor's first Compile user conference: Michael Truell's opening keynote (8:13)

- **Repo/file to study:** `llama.cpp's .pi/gg/SYSTEM.md`. Georgi Gerganov's local setup is intentionally tiny—`pi -nc --offline` plus a short system prompt. Start with the `SYSTEM.md` and the `ggml-org Assisted-by` commit trail if you want a minimal maintainer-grade local-agent workflow [12].

Editorial take: more code is already cheap; the leverage has moved to intent control, review gates, and agent-native infrastructure. [1, 2, 4]

Sources

1. Agentic Code Review
2. X post by @kentedodds
3. SpaceX Just Bought Cursor for \$60B. It's About to Take OVER.
4. X post by @cursor_ai
5. X post by @swyx
6. X post by @swyx
7. Cursor's first Compile user conference: Michael Truell's opening keynote
8. X post by @LangChain
9. X post by @LangChain
10. X post by @rileybrown
11. X post by @kalomaze

12. Quoting Georgi Gerganov