Validation Becomes the Bottleneck for Coding Agents

Coding Agents Alpha Tracker

2026-06-25

Validation Becomes the Bottleneck for Coding Agents

By Coding Agents Alpha Tracker • June 25, 2026

Today's brief is about the new control plane for coding agents: isolated sandboxes, critique/review loops, contextual policies, and the most relevant releases from Crabbox, Databricks, Cursor, and Sourcegraph.

TOP SIGNAL

The big shift in today's sources: teams already deep into coding agents are no longer asking whether the agent can write code — they're redesigning the validation layer around it. Boris Cherny says Claude Code has written 100% of the Quad Code codebase for 6+ months and that he runs hundreds to thousands of agents overnight [1], while Jason Zhou says that after his team adopted autonomous loops and 10+ concurrent sessions, the bottleneck moved from code generation to safe review and merge [2]. That is why the interesting work today is isolated sandboxes, review councils, narrow deploy primitives, and contextual policies — not just better prompting [2, 3, 4, 5].

> "The AI step of writing the code is moving the bottleneck to other parts of the SDLC." [3]

TRY THIS

- **Give every agent its own disposable test box.** Jason Zhou and AI Jason's Crabbox recipe is explicit: build a Docker image with the repo's tools, write `crabbox.yaml` with a fast provider like Daytona plus snapshot/sync excludes/env vars, add a one-command `setup.sh`, then run `crabbox warmup` → `crabbox run ...` → `crabbox stopbox` [2]. Add Playwright CLI and artifact commands (`artifacts collect`, `artifacts videos`, `artifacts publish`) so the PR comes back with

screenshots/video instead of a trust-me summary [2]. If you want a starting point, copy the open-sourced skill in AI-Builder-Club/skills [6, 2].

- **Split generation from critique.** Boris Cherny's literal pattern is to have one agent build, then ask Claude to `double check the result` and `open the app and test it by itself` while 15+ other agents run in parallel [1]. Shopify describes the same structure at org scale: parallel subtasks for production, then sequential critique loops with high-reasoning models; they also restrict engineering to the biggest models because human time is worth more than model cost [3]. Keep the human on the hook at the end — the AI can write the code, but your name still goes on the PR [3].
- **For production-touching agents, expose primitives — not raw power.** PlanetScale's demo is the right template: let the agent query platform recommendations, make changes on a branch, open a deploy request, watch live impact, and roll back fast if the change misbehaves [4]. Databricks applies the same idea one layer up: track session state (for example risky package installs or large confidential-doc reads), map low-level tool events to high-level policies, and cap a sub-agent to `$5` unless it asks for more [5]. Narrow interfaces + stateful controls are the pattern.
- **Delete prompt no-ops from your skills.** Matt Pocock's test, amplified by Kent C. Dodds, is brutally simple: remove a line from the skill, rerun, and if the output does not change, the line was a no-op [7, 8]. This trims token waste and makes skills easier to evaluate and maintain [7].

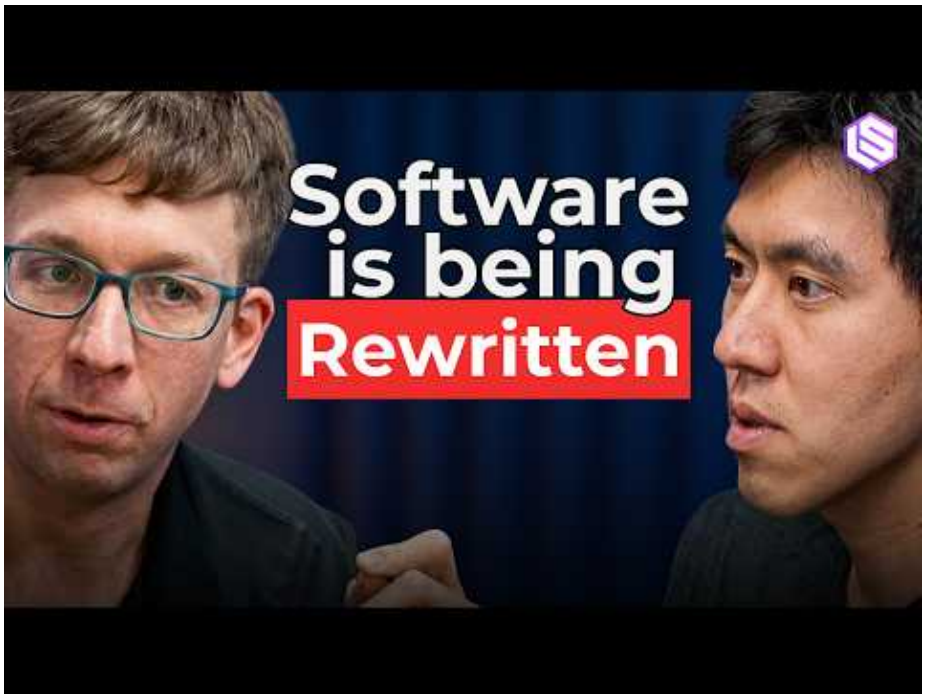
WHAT SHIPPED

- **Crabbox setup skill + deep dive** — Jason Zhou open-sourced a Crabbox setup skill and says the pattern helped his team ship **10x more PRs**; Crabbox gives each local agent a cloud box that syncs uncommitted changes, runs the full stack in isolation, and returns screenshots/videos as proof. Repo: AI-Builder-Club/skills [6].
- **Databricks Omnigen** — Open-sourced meta-harness/common API for sessions, files, streams, tool calls, and cancellation across Claude Code, Codex, Cursor CLI, OpenAI SDK, and more, with collaborative hosting, contextual policies, and spend controls [9, 5]. Databricks says it had around 400 merges soon after release, roughly half from outside the team, with Kubernetes and additional sandbox integrations already landing [5]. Background: Latent Space writeup
- **Cursor Notion** — You can now delegate tasks to Cursor directly from Notion; because it uses the Cursor SDK, the cloud agent runs on the same models, harness, and runtime as Cursor. The user flow is simple: assign `@Cursor` to a spec/task and it opens a PR the team can review [10]. Details: Notion build post [11].

- **GLM 5.2 in Cursor** — Cursor now supports **GLM 5.2**, its first GLM integration; Jediah Katz explicitly asked users to report any strange behavior, and the launch included eval results [12, 13].
- **Sourcegraph Deep Search auto-compaction** — Deep Search now automatically compacts long conversations when a follow-up gets close to the context window limit. Changelog: sourcegraph.com/changelog/deep-search-auto-compaction [14, 15].

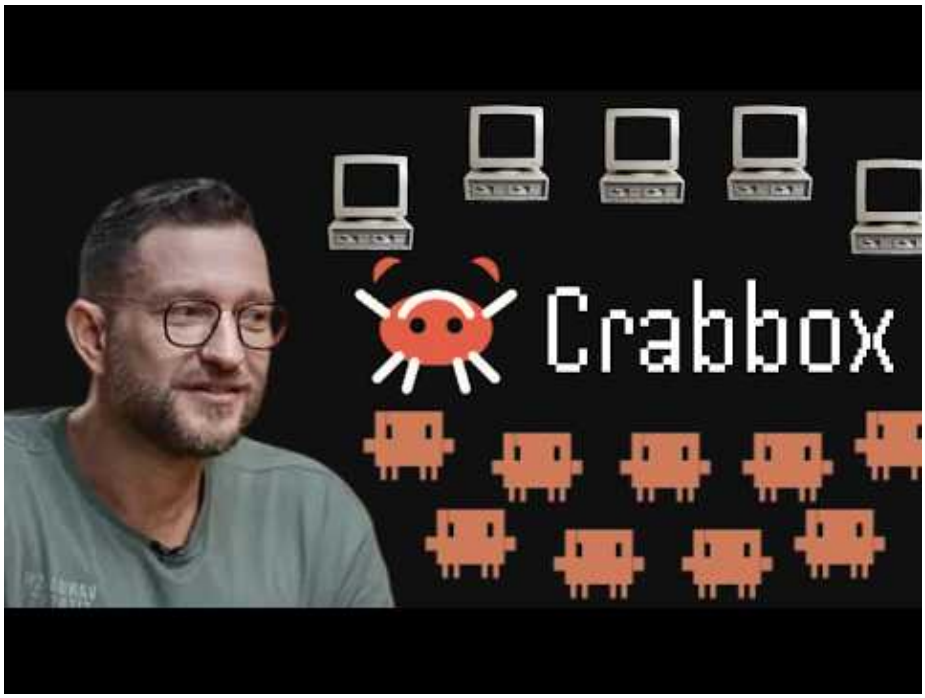
GO DEEPER

- **19:36-22:39 — Matei Zaharia on contextual policies.** Best clip today if you're building internal agent infra: why static allow/deny breaks down, how to track risky actions across a session, and why spend caps should live in the session state [5].



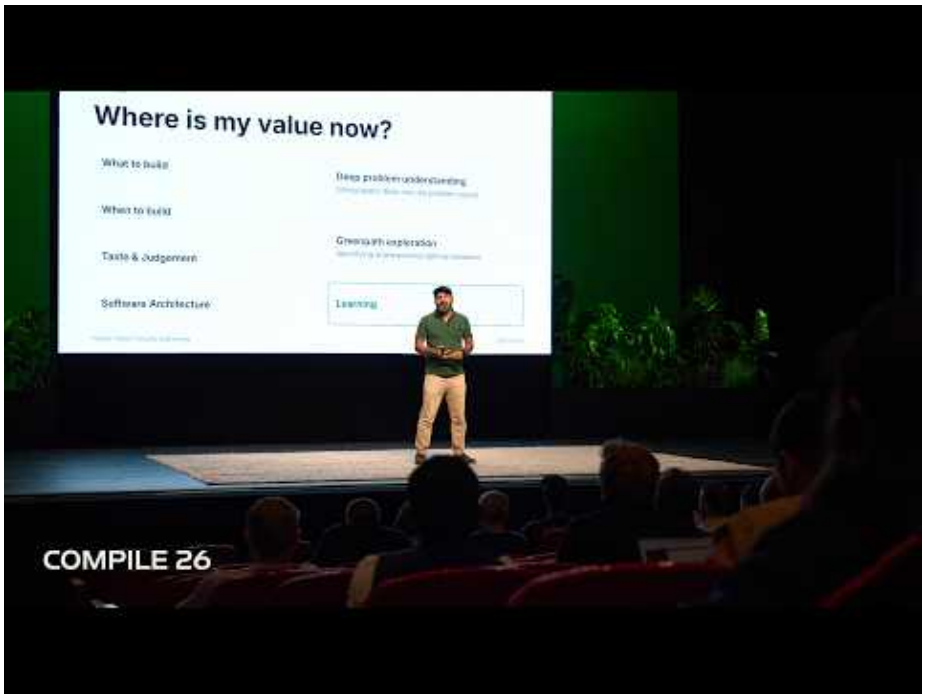
The Agent Cloud: Databricks' Bet on the Future of AI — Matei Zaharia and Reynold Xin (19:36)

- **0:53-1:21 — AI Jason on proof-first PRs.** Short, concrete rationale for making agents return Playwright artifacts so reviewers merge based on evidence, not narration [2].



OpenClaw Creator's new secret project… (0:53)

- **21:57-22:51 — Farhan Thawar on the `Council of LLMs`.** A useful review architecture: have different models judge different aspects of a change before production, but keep a human responsible for the final PR [3].



What Is Your Job Now, Farhan Thawar | Compile 26 (21:56)

- **Study AI-Builder-Club/skills.** It is one of the rare public, copyable setups that turns isolated sandboxes + verification artifacts into a reusable agent skill rather than a one-off demo [6, 2].

Editorial take: the winning teams are standardizing the control plane around agents — isolated runtimes, narrow deploy interfaces, session-aware policies, and explicit human ownership — because code generation is no longer the bottleneck. [2, 3, 5]

Sources

1. Claude Code Creator: "I Run Thousands of AI Agents Every Night"
2. OpenClaw Creator's new secret project…
3. What Is Your Job Now, Farhan Thawar | Compile 26
4. Agents and Infrastructure, Sam Lambert | Compile 26
5. The Agent Cloud: Databricks' Bet on the Future of AI — Matei Zaharia and Reynold Xin
6. X post by @jasonzhou1993
7. X post by @mattpocockuk
8. X post by @kentcdodds
9. Why the Frontier Ecosystem must be Open — Matei Zaharia and Reynold Xin, Databricks

10. X post by @cursor_ai
11. X post by @cursor_ai
12. X post by @leerob
13. X post by @jediahkatz
14. X post by @Sourcegraph
15. X post by @Sourcegraph