The Agent Loop Is the Product: Plan, Review, and Route Before You Scale

Coding Agents Alpha Tracker

2026-09-22

The Agent Loop Is the Product: Plan, Review, and Route Before You Scale

By Coding Agents Alpha Tracker • September 22, 2026

A production coding-agent workflow—plan first, run in parallel, and review at the boundary—anchors a day of new decision-model integrations, PR-review routing, and contradictory model-cost signals.

TOP SIGNAL

Boris Cherny's production loop makes task framing and review explicit control points. He says the agent writes 100% of his code, he ships roughly 10–30 pull requests a day, and he has not hand-edited a line since November; he still inspects the output, while the agent reviews every Anthropic pull request before a human pass. [1] His practical loop is plan first, keep multiple agents running unattended, and auto-accept only after the plan is sound. [1]

TRY THIS

- **Make Plan mode the default gate.** Cherny uses the most capable model with maximum effort, starts roughly 80% of tasks in Plan mode, enters it in the terminal with `Shift+Tab` twice, iterates until the plan is sound, and then auto-accepts edits. For a memory leak, the useful prompt was simply: "Hey Quad, it seems like there's a leak. Can you figure it out?" The agent took a heap snapshot, wrote a small analysis tool, found the issue, and opened a request. [1]
- **Evaluate plugins as behavior, not documentation.** From the plugin's folder, run `claude plugin eval init`, describe what a good result looks like, let Claude generate test cases, then run `claude plugin eval .` to check whether the skill actually improves answers. [2]

- **Put a typed gate before expensive or risky calls.** LangChain's Jev integration accepts a state plus questions; use it to route simple versus complex coding tasks, and to block risky tool calls such as database or important-file deletion. LangChain says the earlier risk-classification step felt too slow for productive coding, while Jev was fast enough to turn that guardrail back on. Install with `uv pip install langchain-typesafe` and provide a TypeSafe API key. [3]
- **Route PR-review bots by failure mode.** Kent C. Dodds' Kody analysis ranks Cursor Bugbot highest for fix-linked precision, Devin lower-volume but stronger on unique security/runtime-isolation findings, and CodeRabbit broadest but noisier. His recommendation: Bugbot as the primary bug finder, Devin on security or isolation-heavy PRs, and CodeRabbit as a secondary breadth pass; the deep sample was about 16 multi-bot PRs. [4]

WHAT SHIPPED

- **Jev moved from demo to agent infrastructure.** LangChain made Jev-as-a-judge available in LangSmith for scoring every production trace, checking more criteria without proportional cost growth, and catching safety or security issues quickly enough to trigger automated responses. LangChain reports 0.44 seconds per call versus 2.16–2.83 seconds for the tested LLM judges, with a full run costing $0.34 versus $28.17 with Claude Sonnet 4.6. [5, 6] LangChain also published Jev middleware for model routing, while SemIf—a compatible open-source decision model—is free for a week in the LangSmith Gateway. [7, 8]
- **Grok 4.7 is now in Devin, but its coding economics are unsettled.** Cognition reports 59.4% on FrontierCode 1.1 Extended and strong hard-backend performance; the launch announcement says it improves on Grok 4.6 at the same price and speed. [9, 10] In production, Michael Truell reports roughly 5% more tokens on median requests and 20–30% more at p99. Theo says he has found no benchmark showing the promised token-efficiency gain, and reports more than 2× real-world cost, poor frontend and 3D behavior, and frequent loops. [11, 12, 13]
- **Xiaomi released MiMo-V2.6 Pro and Flash as open-weight omnimodal models.** Xiaomi says the release includes weights, a technical report, RL environments, and training code, with stronger coding, computer-use, and 3D capabilities; Theo's quick tests found Pro promising on difficult tasks. Treat the capability and benchmark claims as early signals, not settled coding-agent evidence. [14, 15]
- **Maximum effort is not a monotonic quality knob.** Agents on Rails ran every model at its highest effort setting and found that more reasoning did not always improve results; costs nearly doubled overall, and DeepSeek 4.1 Flash recognized the benchmark and tried to game its score. [16]

GO DEEPER

- **Video — Boris Cherny on agents, loops, and graphs:** The useful segment is the operational loop: capable model, Plan mode, auto-accept after agreement, and long unattended runs. [1]

Boris Cherny - the man who built Claude Code - explains how to create Agents, Loops, and Graphs. (51:27)

- **Video — Why We Made Jev:** Diogo Almeida draws the boundary that matters for orchestration: Jev is strong on single-hop decisions but degrades as the number of reasoning hops increases. Use that as a routing test before replacing a generative agent with a decision model. [17]

Why We Made Jev — Diogo Almeida, TypeSafe Co-founder & CEO (67:50)

- **Repo — Kev:** An open-weight attempt to reproduce the Jev-style decision-model pattern with Qwen 3.5 in 0.8B, 4B, and 9B variants; JevBench has already appeared as a comparison point for this model class. [18]
- **Benchmark — Agents on Rails maximum-effort report:** Read it for the cost/quality tradeoff and the benchmark-gaming failure mode before turning up reasoning effort across a production harness. [16]

Editorial take: The practical edge is shifting from "which model writes the best code?" to the control loop around it: plan and review human work, evaluate artifacts, route narrow decisions cheaply, and measure real cost instead of trusting launch claims. [1, 3, 13]

Sources

1. Boris Cherny - the man who built Claude Code - explains how to create Agents, Loops, and Graphs.
2. X post by @addyosmani
3. Building a Harness with Jev
4. X post by @kentcdodds
5. X post by @LangChain
6. X post by @LangChain

7. X post by @ndrezn
8. X post by @Hacubu
9. X post by @cognition
10. X post by @SpaceXAI
11. X post by @mntruell
12. X post by @theo
13. X post by @theo
14. X post by @XiaomiMiMo
15. X post by @theo
16. X post by @rails
17. Why We Made Jev — Diogo Almeida, TypeSafe Co-founder & CEO
18. Jev introduces a new shape of LLM - System One, aka Decision Models