

Background Loops, Agent Docs, and Multi-Model Routing

Coding Agents Alpha Tracker

2026-05-05

Background Loops, Agent Docs, and Multi-Model Routing

By Coding Agents Alpha Tracker • May 5, 2026

Today’s best signals are operational, not aspirational: top practitioners are turning coding agents into scheduled background workers. The practical edge is in loops, future-aware prompts, on-distribution stacks, repo-specific docs, and the infrastructure shipping around them.

TOP SIGNAL

Today’s clearest signal: background agents are becoming the real coding workflow. Boris Cherny says he runs dozens of Claude Code `/loops` to babysit PRs, fix flaky CI, and cluster feedback every 30 minutes, with new server-side **routines** keeping jobs alive when the laptop is closed [1]. Alexander Embircos says Codex already supports the same time-based pattern for unresolved discussions, launch bugs, and flaky tests [2]—and Riley Brown’s warning is the useful counter-signal: cronjobs, memory, re-auth, and file-placement reliability are still where agent power users lose time [3].

TRY THIS

- **Steal Boris Cherny’s first three loops.** Set up recurring jobs for PR **babysitting** (auto-rebase + fix CI), **CI health** (catch/fix flaky tests), and **feedback triage** (cluster feedback every 30 minutes). Run them on a cron-style repeat via `/loop`; if your tool has server-side execution, move long runners there so they keep working offline [1].
- **Use future-oriented prompts as lightweight automation.** Embircos says he uses this pattern in Codex all the time and that it’s powerful but non-obvious [2, 4]. > “tomorrow, check in on this discussion and ping me if it isn’t resolved” [2] > > “let me know if this bug isn’t fixed by the

day before launch” [2] > > “bug me if this flaky test doesn’t go green after retry” [2]

- **If you want max agent throughput, bias toward boring, on-distribution tech.** Cherny says Claude Code’s codebase is simple TypeScript + React, originally chosen because that combo was very on distribution for the model; that helped them reach 100% model-written code early [1]. If you’re starting greenfield and expect heavy agent involvement, this is the pragmatic default [1].
- **Write the migration doc before the port.** Bun appears to be exploring a Zig→Rust port with a dedicated docs/PORTING.md aimed at coding agents [5, 6]. Steal the pattern: if agents are handling a big refactor or language move, give them a repo-local playbook first [5].

WHAT SHIPPED

- **Bun showed two strong agent-native signals.** Armin Ronacher reported a bug and says a coding agent fixed it and pushed PR #30257 within five minutes; later, agents were debating on the PR itself [7, 8]. Simon Willison separately spotted Bun’s agent-specific PORTING.md as the project explores a Zig→Rust port [5, 6].
- **Vercel deepsec.** New open-source coding security harness: CLI-first, sandbox-based scaling, pluggable coding agents, large-repo focus, and support for AI Gateway or your own subscription. Vercel says it followed months of internal use and tests on some of the largest open-source codebases; blog: introducing deepsec [9].
- **deepagents-cli + Profiles API.** LangChain is pushing it as a model-agnostic harness for open-weight coding agents. Recent CLI features: /agents, /model, headless --json + --max-turns, --acp, /skill:name, and MCP with OAuth; docs: overview [10, 11].
- **LangSmith Fleet multi-model routing.** Sub-agents can now use different models, with the stated goal of pushing simple work to fast/cheap models and saving stronger models for the hard parts; page: Fleet [12, 13].
- **Gemini API infrastructure updates.** Logan Kilpatrick says webhooks are live for long-running tasks including agents, and the Interactions API now returns more human- and agent-readable error messages [14, 15].
- **Codex adoption looks real; Copilot economics look strained.** @linuz90 called Codex his favorite coding app and says it now handles 90%+ of his work despite earlier terminal/lock-in hesitation [16, 17]. Theo says one Copilot message burned through 60M+ tokens/\$30, and 15 messages totaled \$221 of tokens under a flat-message plan he thinks GitHub cannot sustain [18, 19].

- **Model preference is still split.** Cherny says Claude Code reached 100% agent-written code on a simple TypeScript/React codebase, with each Anthropic release from Opus 4 through 4.7 improving the curve [1]. Theo, by contrast, says Anthropic is still meaningfully worse than OpenAI for most code outside frontend, even though many enterprise developers use Claude/Opus via Bedrock, Cursor, or Copilot in existing cloud setups [20].

GO DEEPER

- **7:35-8:49** — **Boris Cherny's /loop playbook.** Best short walkthrough today of a practical background-agent setup: PR babysitting, CI repair, feedback clustering, and why server-side routines matter once jobs need to survive laptop sleep [1].



Anthropic's Boris Cherny: Why Coding Is Solved, and What Comes Next (7:35)

- **19:50-20:37** — **When the model starts the loop for you.** Cherny says Claude 4.7 increasingly notices time-varying work on its own, starts a loop, and offers 30-minute Slack reports [1].



Anthropic's Boris Cherny: Why Coding Is Solved, and What Comes Next (19:50)

- **Study Bun's live artifacts, not the discourse.** PR #30257 is a report→fix→PR example that landed minutes after a bug report, and Bun's docs/PORTING.md shows what agent-facing migration guidance can look like in a real repo [7, 8, 5].
- **Study Simon Willison's narrow-tool workflow.** His Redis Array Playground and PR #277 show Claude Code for web being used for a focused dev utility around one new Redis feature, not a giant monolith ask. More context: blog post [21, 22].

Editorial take: the edge is shifting from single-shot codegen to reliable background workflows—loops, timers, sub-agent routing, and repo-specific guidance. [1, 2, 12, 5, 3]

Sources

1. Anthropic's Boris Cherny: Why Coding Is Solved, and What Comes Next
2. X post by @embirico
3. X post by @rileybrown
4. X post by @embirico
5. X post by @simonw

6. X post by @LukeParkerDev
7. X post by @mitsuhiko
8. X post by @mitsuhiko
9. X post by @vercel_dev
10. X post by @masondrxy
11. X post by @LangChain
12. X post by @BraceSproul
13. X post by @LangChain
14. X post by @OfficialLoganK
15. X post by @OfficialLoganK
16. X post by @linuz90
17. X post by @gdb
18. X post by @theo
19. X post by @theo
20. Microsoft and OpenAI break up (Amazon is pumped)
21. X post by @simonw
22. Redis Array Playground