

Claude Routines Go Async While Codex Tactics and Composer 2.5 Raise the Bar

Coding Agents Alpha Tracker

2026-05-25

Claude Routines Go Async While Codex Tactics and Composer 2.5 Raise the Bar

By Coding Agents Alpha Tracker • May 25, 2026

The sharpest shift today is from direct prompting to supervising background agent loops. Inside: the best copyable workflows, shipping features, and model comparisons from Claude Code, Codex, Cursor, Cloudflare, Rails/OpenCode, and more.

TOP SIGNAL

- **Async orchestration is turning into the default dev loop.** Boris Cherny showed Claude Code routines picking up GitHub issues overnight, launching work on local or cloud compute, and CI Autofix babysitting PRs through review comments, security issues, flaky CI, and merge conflicts; the cloud desktop app is built to manage many parallel sessions, not one chat at a time [1]. bcherny's practical enabler is auto mode (**Shift+Tab**): remove permission prompts, let one session run, and start another in parallel while it executes [2].

TRY THIS

- **Run a weekly self-improvement pass on Codex.** Use Greg Brockman's structure: scan the last 30 days across recent sessions, memories, Chronicle, and existing automations; shortlist only workflows that happened at least twice, have stable inputs/outputs, and are not already covered; then create the *smallest* artifact that fits — **Skill**, **Custom subagent**, **Automation**, or **Skip** [3, 4]. The key constraint is to force the shortlist first and only create high-confidence missing items [3].
- **Turn issues into overnight PRs.** 1) Add a routine that watches GitHub issues. 2) Trigger Claude Code sessions on a schedule, webhook,

or API call. 3) Run them locally or on remote cloud compute. 4) Use auto mode (**Shift+Tab**) so sessions do not stall on permission prompts. 5) In the morning, triage the cloud app's buckets: running, needs input, merged/closed [1, 2]. Then let CI Autofix babysit the PR to green by handling review comments, security issues, retries, and rebases [1].

- **If you're building agent tooling, expose fewer tools and more execution power.** Sunil Pai's Cloudflare pattern is to expose only **search** and **execute**, have the model submit JavaScript, run it inside an isolate, type-check it, and block outgoing traffic by default unless you explicitly allow APIs [5]. Keep the harness separate from the execution environment so you can swap infra later without rewriting the agent contract [5].
- **For big agent-driven refactors, optimize the terrain.** Kristofer Lund's Rails test got a simple CRM with migrations, validations, auth, and backend structure from one prompt in about 15 minutes; DHH's operating advice is to lean on a linter and test suite, keep prompts token-efficient, and not assume static types are the main gating factor for agent performance [6, 7, 8].

WHAT SHIPPED

- **Claude Code stack:** auto mode is now available on the Pro plan, Sonnet 4.6 is supported alongside Opus 4.7, and Boris says routines, cloud desktop app updates, and CI Autofix are all available today [9, 1]. Firsthand adoption signal: Boris says a lot of his own code is now written by routines rather than direct prompting [1].
- **Cursor Composer 2.5:** Cursor's new distilled coding model is based on Kimi K2.5 and only available inside Cursor/ACP CLI/SDK; pricing is \$0.50/M input and \$2.50/M output tokens, with a reported 63% on Cursor Bench versus GPT-5.5 at 64% and Opus 4.7 at 65% at much higher cost [10]. Theo's workflow read: strong for fast interactive back-and-forth in a real IDE on large repos, less ideal for huge parallel swarms, and still no public API [10].
- **Codex/GPT momentum keeps getting corroborated.** Theo says Codex is about 10x better than it was in December and ahead on end-to-end features like computer use, goals, and remote control; separately, Adam says his 20-person team overwhelmingly uses GPT models, mainly because they do better on broader project scoping, while Claude remains stronger on UI-heavy work [11, 12].
- **OpenCode/Rails comparison worth watching.** Kristofer Lund says OpenCode built a simple Rails CRM from one prompt in 15 minutes with migrations, validations, auth, and backend structure; his takeaway is that older, proven stacks like Ruby/Rails and Linux give agents better footing, and DHH adds that linters plus tests matter more than types for many

refactors [6, 7].

GO DEEPER

- **01:09-02:08** — **Claude adds refunds, catches its own race condition, and verifies in-browser.** Best short demo of self-verification attached to real product work: idempotency, multi-currency handling, audit logging, then a browser check that catches a missing success toast before the task closes [1].



Code w/ Claude 2026 San Francisco *Boris Cherny (1:08)*

- **03:08-04:03** — **Boris Cherny on routines as higher-order prompts.** Watch this for the cleanest explanation of the overnight issue-to-PR loop and why reusable automations are starting to matter more than one-off prompts [1].



Code w/ Claude 2026 San Francisco *Boris Cherny (3:07)*

- **02:35-03:26** — **Sunil Pai on collapsing 2,600 APIs into search + execute.** This is the best short clip today for anyone building MCP servers or internal agent platforms: fewer tools, more sandboxed code execution, fewer LLM round-trips [5].



Why you should build Science Fiction — Sunil Pai, Cloudflare (2:35)

- **Repo worth skimming:** Armin Ronacher’s `go-to-bed.ts` is a tiny example of adding behavioral guardrails to coding agents [13].
- **Artifact worth stealing from:** Simon Willison’s Mad House write-up and Claude share show a clean reconstruction pattern: attach the primary-source PDF, ask for a constrained artifact format, and explicitly tell the model to include attribution and links [14].

Editorial take: today’s durable edge is orchestration, not theatrics — remove permission friction, shrink tool surfaces, and let agents finish verifiable work in the background while you manage the exceptions. [2, 5, 1]

Sources

1. Code w/ Claude 2026 San Francisco — Boris Cherny
2. X post by @bcherny
3. X post by @reach_vb
4. X post by @gdb
5. Why you should build Science Fiction — Sunil Pai, Cloudflare
6. X post by @kristoferlund
7. X post by @dhh
8. X post by @dhh

9. X post by @ClaudeDevs
10. Cursor just crushed Claude Code
11. X post by @theo
12. Recovering from AI Psychosis | TheStandup
13. X post by @mitsuhiko
14. Mad House — Usborne Creepy Computer Games