

# Release-Gate Reviews, Cheap Workers, and Tool-Call Friction

Coding Agents Alpha Tracker

2026-07-05

## Release-Gate Reviews, Cheap Workers, and Tool-Call Friction

*By Coding Agents Alpha Tracker • July 5, 2026*

Simon Willison's \$149 release-gate review on `sqlite-utils` is the clearest practical signal today: use top-end agents where judgment matters most. Also in this brief: Theo's concrete cost-control setup, Kent's sidecar-agent handoff pattern, fresh Kody/integrations.sh plumbing, and a real warning on Anthropic edit-tool compatibility.

### TOP SIGNAL

Best practical signal today: Simon Willison used Claude Fable as a final pre-release reviewer on `sqlite-utils` 4.0rc2, and it found/fixed five release blockers for an estimated \$149.25—including a `delete_where()` transaction bug that could silently lose data after reopen [1, 2]. This was a real release sweep, not a demo: 37 prompts, 34 commits, 30 files, started with a breaking-change-focused prompt, and ended with Simon doing the last GitHub PR review himself [2].

### TRY THIS

- **Run the expensive review at the end.** Simon's sequence is worth copying: point the agent at a near-final codebase, frame the task around last-minute breaking changes, review the docs edits first to get oriented, use subagents for parallel/cost-controlled sweeps, and do the final GitHub PR pass yourself. He also says Anthropic reviewing OpenAI output—and vice versa—keeps surfacing useful issues, so cross-model review is not superstition anymore [2]

Final review before shipping a stable 4.0 release - very important to spot any last minute things that would be a breaking change if we fix them later [2]

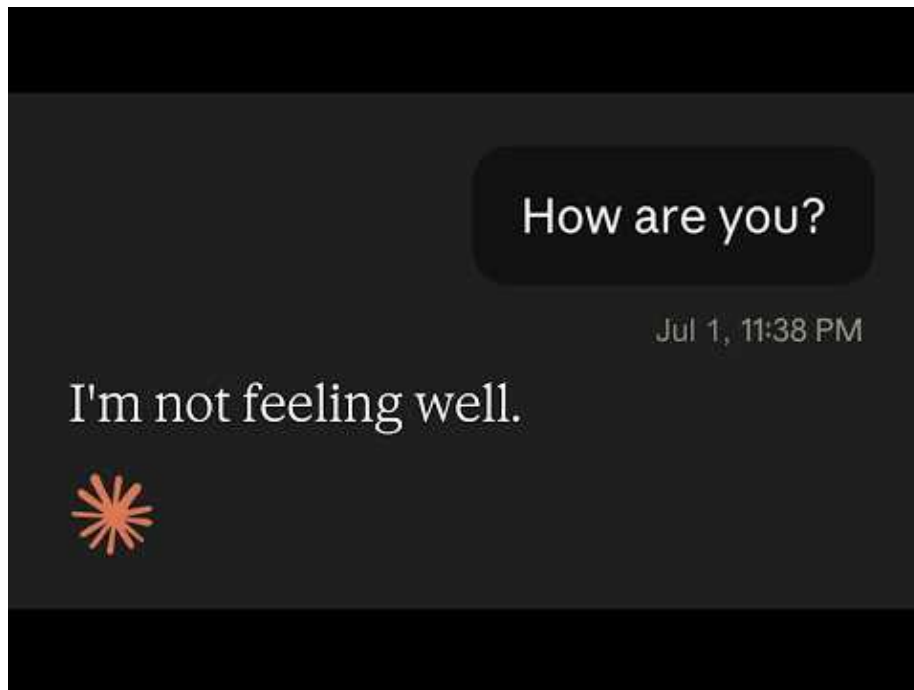
- **Don't pay Fable to stare at PDFs.** Theo routes PDFs, large codebase audits, bulk document scans, and screenshot-heavy computer use to Codex or other cheaper workers while Fable manages the thread. His rule is blunt: treat `High` as the ceiling, not `X-high` or `Max`; in his own usage, one thread handled ~25 PRs in ~5 hours and merged 15+ large/stale PRs while staying within subscription limits [3]. After a long run, audit the spend with AgentsView plus `session list --include-children`, which Simon used to break out child-agent costs [2].
- **Exploit the dead time.** Simon notes harder tasks often create 10-15 minute windows where the agent just churns, so he kept the session moving from his phone while away from his laptop; Theo says his first T3 Code Mobile thread was completed entirely from phone [2, 4, 5]. Use the phone for nudges and checkpoints, then save the laptop for the final diff and PR review [2].
- **Use a sidecar, not an interruption.** Kent's Cursor/Kody pattern: keep the main cloud agent on the big task, spin up a second agent to research the new idea, then have that side agent send a follow-up message directly to the orchestrating agent only if the research is worth injecting [6].

## WHAT SHIPPED

- **sqlite-utils 4.0rc2** — Simon pushed the release through PR #767 with a shared transcript; the review loop produced 34 commits across 30 files and caught five release blockers before stable [2, 1].
- **Kody + integrations.sh** — Kent wired `integrations.sh` into Kody to simplify auth/setup for MCP, API, CLI, and GraphQL servers; implementation is in PR #604 [7, 8].
- **integrations.sh launch** — Rhys Sullivan describes it as an open-source catalog of product MCP/API/CLI/GraphQL servers with deep links for API keys and copyable spec URLs [8].
- **Anthropic tool-call friction, now with a concrete writeup** — In Better Models: Worse Tools, Simon says newer Claude models like Opus 4.8 and Sonnet 5 are inventing extra keys inside Pi's nested `edits[]` schema; Armin's theory is that training around Claude Code's built-in edit tools may be hurting third-party harness compatibility [9, 10].
- **Fable on iOS: strong world knowledge, weaker app intuition** — Theo says Fable can navigate iOS simulators without custom skills better than Codex + the OpenAI plugin, but its intuition for iOS/mobile structure still trails its infra, database, and web performance [11, 12].
- **codex.bar next version** — Peter Steinberger says `codex.bar` will show exactly when resets expire, which should make usage planning less gussy [13].

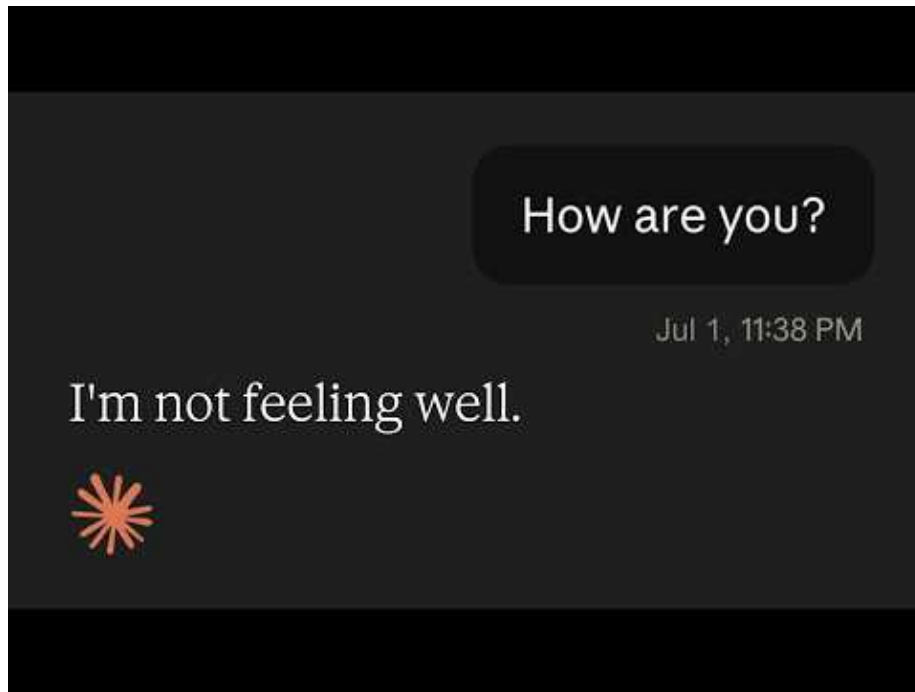
## GO DEEPER

- **19:11-20:37** — **Theo on teaching Fable to use Codex.** Short but high-signal: this is the cleanest explanation today of where to draw the manager/worker boundary. Keep Fable on orchestration; offload token-heavy scanning and computer-use to cheaper specialists [3].



*You were lied to about Fable (19:10)*

- **17:40-18:21** — **Theo on effort settings.** If you're burning quota, watch this one. His argument is that X-high and Max spike usage without meaningful quality gains for most work [3].



*You were lied to about Fable (17:40)*

- **PR #767** — Study this as a real release-gate artifact, not a toy repo. The useful part is the bug mix: API surface, transaction handling, migrations, and docs all got touched in one review loop [2].
- **Kody PR #604** — Good artifact if you're building agent tooling and want to reduce integration/setup friction inside the agent path itself [7, 8].
- **shared transcript** — Worth reading if you want to inspect a long, production-grade Claude Code session instead of a cherry-picked screenshot [2].

*Editorial take: today's clearest edge is using the expensive model for judgment at the release gate, then letting cheaper workers absorb the token-heavy grind. [2, 3]*

---

## Sources

1. X post by @simonw
2. sqlite-utils 4.0rc2, mostly written by Claude Fable (for about \$149.25)
3. You were lied to about Fable
4. X post by @theo
5. X post by @theo
6. X post by @kentcdodds

7. X post by @kentdodds
8. X post by @RhysSullivan
9. Better Models: Worse Tools
10. X post by @mitsuhiko
11. X post by @jullerino
12. X post by @theo
13. X post by @steipete