

DoorDash’s Repo-Context Playbook, Simon’s Release Audit, and GPT-5.6’s Early Edge

Coding Agents Alpha Tracker

2026-07-08

DoorDash’s Repo-Context Playbook, Simon’s Release Audit, and GPT-5.6’s Early Edge

By Coding Agents Alpha Tracker • July 8, 2026

The strongest coding-agent signal today: teams getting real leverage are externalizing judgment into repo-native artifacts—architecture principles, skills, upgrade guides, and audit prompts. This brief covers DoorDash’s concrete operating model, Simon Willison’s release-hardening loop, and early practitioner feedback on GPT-5.6.

TOP SIGNAL

The best teams today are not just picking stronger models; they’re **making engineering judgment legible inside the repo**. At DoorDash, one successful tech lead wrote “50 plus” architectural principles into markdown files so the agent could reference them, and the team turned repetitive mobile setup/test flows into callable skills [1]. Simon Willison hit the same pattern from the release side: he gave Claude Fable 5 the sqlite-utils 4.0 changelog + upgrade guide, had it write scratch scripts to exercise new features, and the run surfaced 4 blockers that became a 16-commit cleanup PR before release [2].

TRY THIS

- **Run a pre-release audit with scratch scripts** (*Simon Willison*). Before tagging a major version, ask the agent to diff from the last stable tag, read the changelog + upgrade guide, and write throwaway scripts that hit every new feature. Simon’s exact prompt:

```
review the changes on main since the last tagged 3.x
release - I am about to ship them as sqlite-utils 4.0, a
stable version that promises no backwards-incompatible
fixes for a very long time.
```

```
review the changelog and upgrade guide, and write
yourself scratch scripts to try out all of the new
features in v4 - save those scripts but don't commit
them [2]
```

Fable 5 wrote 12 scripts, found 4 release blockers plus 10 more issues, and even produced a combined repro script; Simon also had the models draft the upgrade guide and release notes [2].

- **Create repo-native context packs, not Slack lore** (*DoorDash / Boris Cherny*). Put the architecture rules senior engineers keep repeating into markdown files inside the repo; DoorDash says one lead wrote 50+ principles this way so the agent could reference them while coding. Then wrap annoying setup flows into reusable skills: its mobile team built skills for one-click simulator spin-up and targeted workflow tests so the agent can run them in the background instead of re-instrumenting everything manually [1].
- **Use git worktrees to parallelize one repo safely** (*DoorDash / Boris Cherny*). Boris says he usually works from the terminal, opens multiple sessions against the same repo, and uses worktrees to avoid conflicts. Practical pattern: one task per worktree, one agent session per task, then merge or discard independently [1].
- **Split planner and executor to keep costs down** (*@ClaudeDevs + Peter Steinberger*). One pattern in the wild: use Fable 5 as an “advisor” while a cheaper executor model does the actual tool-calling work; the example uses Sonnet 5, so most tokens stay billed at the lower executor rate. Steinberger’s extra twist is to ask Fable to make Codex the workhorse, and start from his codex-first SKILL.md [3, 4].

WHAT SHIPPED

- **sqlite-utils 4.0** — a strong public example of agent-assisted release hardening. Simon Willison says Claude Fable 5 wrote 12 repro scripts, found 4 blockers + 10 additional issues, and the fixes landed in PR #779; the project also shipped a detailed upgrade guide explicitly meant to be feedable to a coding agent [2, 5].
- **GPT-5.6 / gpt-5.6-sol** — Theo’s early practitioner read is strong: it’s becoming his “obvious default” for many tasks, is good at subagent orchestration, can run for long stretches without a /goal, knows iOS dev well, and has far fewer rough edges than 5.5. His comparison is nuanced: not quite as “smart” as Fable, but “incredibly capable”; he separately calls **gpt-5.6-sol** “world leading in computer use” and says Hermes felt like a “night and day difference” once 5.6 came back [6, 7, 8].
- **Kody auth upgrade via Cursor + Fable** — Kent C. Dodds says one prompt to Cursor AI + Fable produced functional 2FA and passkey

support in 2.5 hours. The resulting diff is public in PR #661 [9, 10].

- **Gemini API Managed Agents** — Google added background tasks, remote MCP + function calling, network credential refresh, and free-tier access. Google says the goal is to reduce the cost, friction, and complexity of putting capable agents into production, and says thousands of customers are already using the API [11, 12].

GO DEEPER

- **9:01–11:13** — the “**mindset overhang**” clip. Worth watching if your org is still stuck in tool debates: the argument here is that frontier harnesses are now good enough that the bigger differentiator is how leadership communicates usage and changes workflows. It also contains the strongest claim in the batch: some orgs are close to autonomously shipping almost every PR [13].



Startups shipping at scale with Google DeepMind - Best of Next '26 for Startups (9:01)

- **2:10–2:46** — **Boris Cherny on parallel sessions + worktrees.** Short, practical operator detail: terminal first, multiple concurrent sessions, git worktrees to prevent collisions on the same repo [1].



DoorDash gave every employee Claude Code (2:10)

- **PR #779** — Study this if you want a concrete example of agent-found release blockers turning into a real cleanup sequence, not just a demo repo. Pair it with Simon’s upgrade guide to see how release docs can become machine-readable migration input [2, 5].
- **Fable’s 12-script audit gist** — Probably the most copyable artifact of the day if you want to steal Simon’s release-audit loop directly [2].
- **codex-first SKILL.md** — A concrete artifact for advisor/executor routing. Useful if you want to formalize “expensive model plans, cheaper model executes” as a repeatable skill instead of an unwritten habit [4, 3].

Editorial take: the durable edge today is not raw model IQ; it’s how much of your team’s judgment you can turn into files, skills, and prompts that an agent can actually execute against [1, 2].

Sources

1. DoorDash gave every employee Claude Code
2. sqlite-utils 4.0, now with database schema migrations
3. X post by @ClaudeDevs
4. X post by @steipete
5. X post by @simonw

6. X post by @theo
7. X post by @theo
8. X post by @theo
9. X post by @kentdodds
10. X post by @kentdodds
11. X post by @OfficialLoganK
12. X post by @OfficialLoganK
13. Startups shipping at scale with Google DeepMind - Best of Next '26 for Startups