Cloudflare's AI Maintenance Playbook, Agent Kits, and File-First Context

Coding Agents Alpha Tracker

2026-04-05

Cloudflare's AI Maintenance Playbook, Agent Kits, and File-First Context

By Coding Agents Alpha Tracker • April 5, 2026

Cloudflare shows what disciplined AI-heavy repo maintenance actually looks like, while Karpathy, Simon Willison, OpenAI, Matthew Berman, and Theo add practical patterns for agent memory, packaging, deployment, and secret hygiene.

TOP SIGNAL

Cloudflare's ViteNext is the clearest production-grade AI-coding playbook in public right now. Steve Faulkner says AI bots triage issues, review PRs and security, track upstream Next.js commits, and help maintain the repo, while the team relies on ported tests, weekly slop audits, an internal Engineering Codex, and humans who still review and attest every line — `no vibe coding` [1]. Dane Connell adds that the project already has 50+ committers, where a committer can simply mean someone wrote the plan for an agent to implement [1].

TOOLS & MODELS

- **Journey** — Matthew Berman's new registry for installable agent workflows, packaged as `kits` with skills, tools/code, memories, services, tests, failure examples, and learnings. One example kit, `Code refactoring planner v1`, analyzes a codebase with static complexity metrics and uses Claude to produce a phased refactor plan; install is either agent-first or `npm install -g journey-kits` [2].
- **Journey team mode** — private/org kits, shared contexts/resources, and version pinning look like the interesting part. Journey points agents at

existing systems such as 1Password, Supabase, or Firecrawl without storing credentials itself, and Berman says the product is free for discovering/installing kits right now [2].

- **Codex app server + Vercel plugin** — Greg Brockman is now explicitly positioning the Codex app server as the primitive for building agentic apps, and the kitty litter/Litter demo shows why: sessions, chats, skills, agents, folders, and prompts sync between desktop and phone via exposed endpoints. OpenAI Devs also shipped a Vercel plugin inside the Codex app for project-setup-to-deployment flow [3, 4, 5, 6].
- `scan-for-secrets 0.1` — Simon Willison's new Python CLI scans folders and logs for leaked secrets before you share them, including common escaped/encoded variants. He built it with README-driven development in Claude Code [7, 8].
- **Security review got better with newer frontier models** — Salvatore Sanfilippo relays a Linux kernel hacker's observation that AI security reports went from mostly false positives to mostly valid after Opus 4.5 and GPT 5.2; his practical takeaway is that shipping a serious code patch without an articulated AI review is now a mistake [9].
- **Claude Code policy caveat** — if you're building wrappers or CI around Claude Code / the Agent SDK, Matt Pocock says the current rules are still muddy around CI, distributed sandboxes, and commercial software [10].

WORKFLOWS & TRICKS

- **Cloudflare's reusable AI-maintenance loop**
  1. Port real upstream tests and keep unit, end-to-end, and smoke coverage running against production deployments.
  2. Let AI bots handle issue triage, PR review, security review, and upstream change detection.
  3. Encode house rules in an Engineering Codex the reviewer checks automatically.
  4. Run recurring slop audits and feed PR mistakes back into `agents.md`.
  5. Keep humans on architecture and final line-by-line review [1].
- **File-first memory for agents**
  1. Dump raw source material into `raw/`.
  2. Let the LLM compile a markdown wiki with summaries, backlinks, concepts, and local images.
  3. Start the agent from `index.md` or view the repo in Obsidian.
  4. Ask task-specific questions against the files.
  5. File outputs back into the wiki so every query compounds the knowledge base.

  Karpathy says this worked for roughly 100 articles / 400K words without fancy RAG at that scale, and Farza says the filesystem-native wiki beat

his earlier RAG setup; his `Farzapedia` turned 2,500 diary, Notes, and iMessage entries into 400 linked articles for the agent to crawl [11, 12].

- **Share idea files, not full apps** — Karpathy's next move is to publish abstract `idea files` that other people's agents can build locally and customize. He published one gist for the wiki workflow here: https://gist.github.com/karpathy/442a6bf555914893e9891c11519de94f, and argues that `prompt requests` / prompt libraries are becoming more valuable than code repos [13, 14, 15].
- **Spec first, then agent, then leak check** — Simon's pattern is clean: write the README/spec by hand, hand it to Claude Code with red/green TDD, then run `uvx scan-for-secrets $OPENAI_API_KEY -d logs-to-publish/` before publishing logs or transcripts. He also keeps a `~/.scan-for-secrets.conf.sh` file that prints recurring secrets for bulk scans [8].
- **Use the agent to map vendor APIs before changing your abstraction** — Simon had Claude Code read the Python client libraries for Anthropic, OpenAI, Gemini, and Mistral and generate raw `curl` examples for streaming and non-streaming cases before he changed `llm` to support newer features such as server-side tool execution [16].
- **Micro-context hint** — Armin Ronacher says adding his email addresses to agent setup materially improved how the agent interpreted his own comments, users, and log files [17].

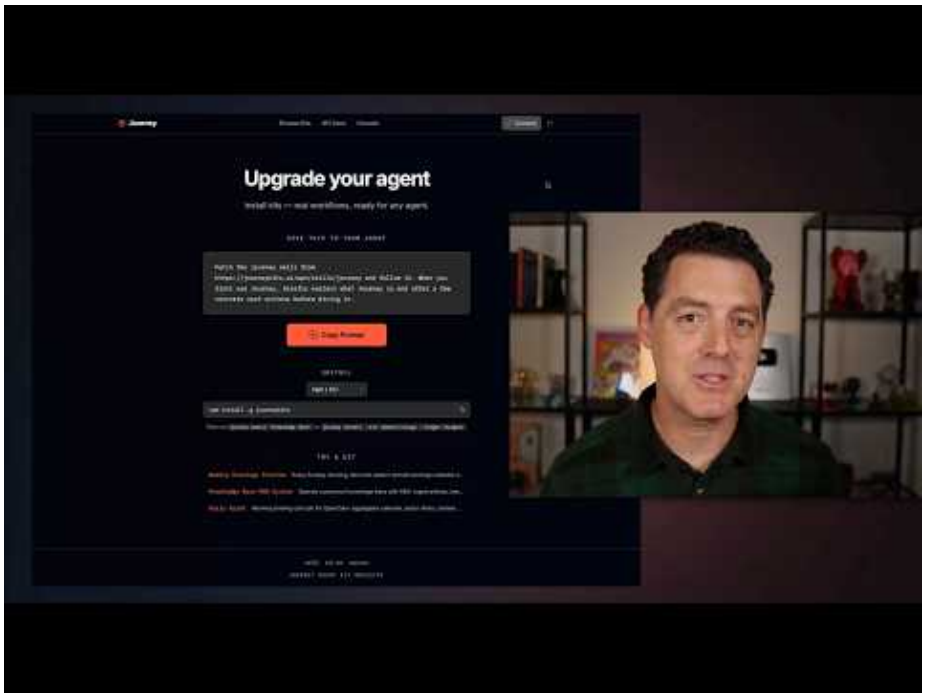
PEOPLE TO WATCH

- **Steve Faulkner + Dane Connell** — rare public operators showing what an AI-heavy open-source repo looks like when you keep tests, policy, and human review intact [1].
- **Andrej Karpathy** — still ahead on durable context design; the interesting new twist is shifting from sharing apps to sharing idea files and prompt libraries [18, 13, 15].
- **Simon Willison** — shipped a genuinely useful utility today and paired it with reproducible build patterns instead of vague prompting advice [7, 8, 16].
- **Matthew Berman** — worth watching because Journey is not just a demo; he's dogfooding it for a team knowledge-base workflow with shared resources and private org kits [2].
- **Theo** — loud, but useful on harness UX/performance and open-source accountability; his T3 Code numbers make the point concrete [19, 20].

WATCH & LISTEN

- **Matthew Berman — Journey org kits and shared contexts (11:56-16:50)** — Best clip today if you want to package an agent

workflow once and reuse it across a team without sharing one agent or leaking credentials. He walks through private kits, 1Password-based resource bindings, audit logs, and keeping agents pinned to the right versions [2].



I built something…. (11:55)

- **Theo — why T3 Code is Electron and open source (21:31-24:53)** — Worth the time for one concrete takeaway: token-stream UIs are harder to make performant than they look. Theo argues Electron plus a reusable event system beat his native experiments, and open-sourcing the harness let the community fork and pressure-test the design [19].

I'm serious. (21:30)

PROJECTS & REPOS

- **T3 Code** — open-source Electron agent orchestrator with a custom event system; Theo says it has about 30k users and 1.1k forks, and he explicitly views the forkability as part of the accountability model [19].
- **ViteNext** — Cloudflare's AI-heavy open-source experiment around the Next.js API surface on Vite/Cloudflare; it already has 50+ committers, with many contributions starting as plans for agents to implement [1].
- **scan-for-secrets** — new Python CLI for secret scanning before you publish logs or transcripts. README: scan-for-secrets [21].
- **Codex CLI / app server** — Theo notes the open-source Codex CLI includes the full app server, which is why third-party harnesses can build against the same primitive the official Codex app uses. Brockman and OpenAI Devs are already using that surface for app-building and Vercel deployment flows [19, 3, 5, 6].

Editorial take: the durable edge is moving out of the model and into external artifacts — tests, markdown files, kits, app servers, and review rules that any decent agent can operate against. [1, 2, 18, 8, 3]

Sources

1. Maintaining a codebase with AI | The Standup
2. I built something….
3. X post by @gdb
4. X post by @LLMJunky
5. X post by @OpenAIDevs
6. X post by @gdb
7. X post by @simonw
8. scan-for-secrets 0.1
9. Il coraggio di capire
10. X post by @mattpocockuk
11. X post by @karpathy
12. X post by @FarzaTV
13. X post by @karpathy
14. X post by @karpathy
15. X post by @NirDiamantAI
16. research-llm-apis 2026-04-04
17. X post by @mitsuhiko
18. X post by @karpathy
19. I'm serious.
20. X post by @theo
21. X post by @simonw