1M Context Goes Default; Memory Agents and PR Filters Get Real

Coding Agents Alpha Tracker

2026-03-14

1M Context Goes Default; Memory Agents and PR Filters Get Real

By Coding Agents Alpha Tracker • March 14, 2026

Claude Code's 1M rollout was the headline, but the sharper practitioner signal was what engineers layered on top: memory-specialized agent stacks, phone-to-laptop session spawning, shared-agent control planes, and better defenses against AI PR noise.

TOP SIGNAL

1M context just stopped being a special-case feature. Opus 4.6 and Sonnet 4.6 are now generally available at 1M context, Opus 4.6 1M is the default Claude Code model on Max, Team, and Enterprise, and the API dropped both the long-context premium and beta header requirement [1, 2, 3].

The higher-signal takeaway is what serious users do next: Boris Cherny says he has been using 1M context exclusively for months [4], while Charles Packer argues that bigger windows do **not** solve the deeper memory problem and recommends pairing a memory-specialized agent with Claude Code or Codex instead of relying on raw context alone [5].

TOOLS & MODELS

- **Claude Opus 4.6 / Sonnet 4.6 — 1M GA.** Opus 4.6 1M is now the default Claude Code model for Max, Team, and Enterprise; Boris says Pro and Sonnet users can opt in with `/extra-usage` [1, 6]. API-side, there is no long-context price increase, no beta header requirement, and support for up to **600 images** per request [1]. Simon Willison highlights that standard pricing now applies across the full 1M window—unlike GPT-5.4 above 272k tokens and Gemini 3.1 Pro above 200k [3]. Docs: model config [4] · announcement [7]

- **Claude Code remote-control — mobile → laptop session spawning.** Run `claude remote-control` on the laptop, then spawn a new local session from the mobile app [8]. Rollout is for Max, Team, and Enterprise on `>=2.1.74`; mobile GitHub setup is still required for now [8].
- **Claw / OpenClaw — live browser control gets serious.** The new beta adds live browser control from latest Chrome via `chrome://inspect#remote-debugging` plus a new **user profile** session [9]. Steinberger says the MCP Chrome session feature gives full access to your browser and logged-in websites, with an extra alert to enable it [10]. Parallel tool calling is also coming to OpenClaw, and Opus 1M has been enabled across providers [11, 12].

WORKFLOWS & TRICKS

- **Treat 1M context as something to steer, not just enable.**
  1. If you are on Max, Team, or Enterprise, Opus 4.6 1M is already the default in Claude Code [1].
  2. If compaction behavior feels wrong, tune it with `CLAUDE_CODE_AUTO_COMPACT_WINDOW` [4].
  3. Boris says he has been on 1M context full-time for months, which is a decent daily-driver signal [4].
- **Three Claude Code shortcuts worth memorizing.**
  - `!<command>` runs bash inline and injects the command plus output into context [13]
  - `Ctrl+S` stashes your draft, lets you ask something else, then restores the original draft after submit [13]
  - `Ctrl+G` opens the prompt or plan in `$EDITOR` for bigger edits [13]
- **Phone → laptop handoff is now a real workflow.**
  1. On the laptop, run `claude remote-control` [8].
  2. In the mobile app, spawn a new local session [8].
  3. Make sure you meet the plan/version requirements and have GitHub configured on mobile [8].
- **Use a memory agent as the control plane.**
  - Letta's concrete pattern: run Claude Code, then use a hook to fire a Letta agent that curates memory into a `CLAUDE.md` file or a dedicated memory/context repo [5].
  - The more interesting inversion is to make the memory-specialized Letta agent your main interface, then let it dispatch to Claude Code or Codex for narrow execution [5].
  - The target is **higher-level reflections**, not mundane logs [5].
- **Use a shared channel as the control plane for multiple agents.** Slack's internal pattern is a shared channel where tools like Linear, Cur-

sor, and Claude Code can send notifications, read each other's messages, and operate with humans in the loop; the channel itself becomes a useful context boundary [5].

- **Fight AI PR flood with trust filters, not heroics.**
  - Theo's setup uses `vouch.md` plus the **Vouch** workflow to label trusted PR authors; on T3 Code it cut the active review surface from **150** open PRs to **43** trusted ones [14].
  - His gold standard is still boring: small, explicit, issue-linked changes—often **1-5 lines** [14].
  - Add **PR Stats** if you want merge-rate and history context per contributor [14].

  "Please do not use clankers to add more noise to PRs. We're working on a solution to this, and this is making my job harder." [15]

- **If agent throughput is stressing CI, remove the obvious bottleneck first.** Theo switched one GitHub Actions job from `ubuntu-latest` to Blacksmith's CPU runner and saw runtime drop from about **2.5 minutes** to **under 1 minute**, while cost was cut in half; the dashboard also helped isolate flaky tests [14].

PEOPLE TO WATCH

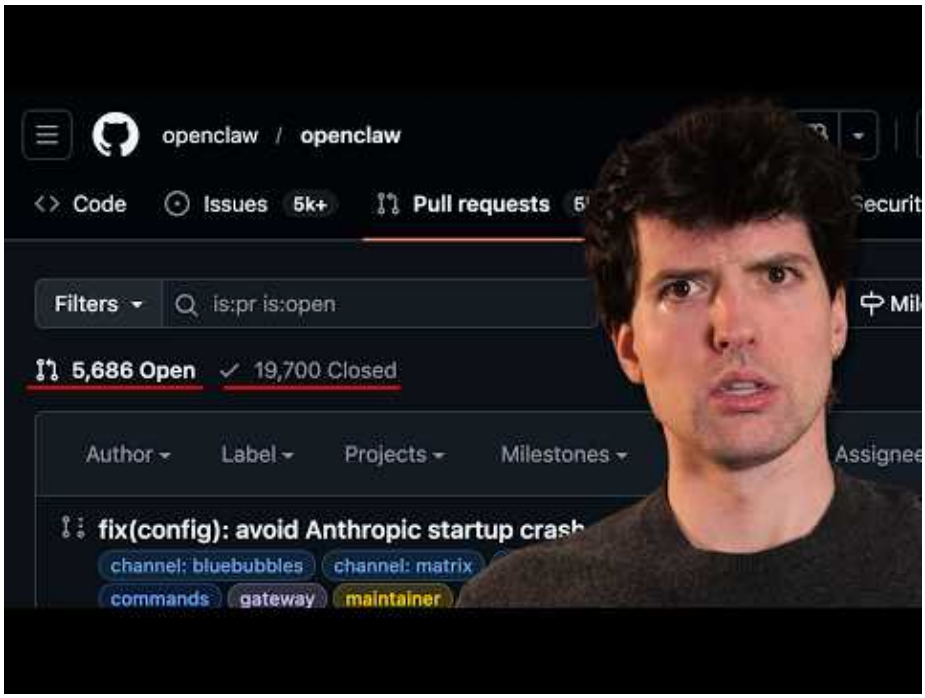
- **Boris Cherny** — high signal because he is sharing operator-level Claude Code details, not just release notes: 1M default rollout, the compaction knob, and phone-launched laptop sessions [6, 4, 8, 16].
- **Peter Steinberger (@steipete)** — one of the best public follows for open coding-agent infrastructure right now: browser control, MCP permissions, parallel tool calls, and blunt maintainer feedback on PR noise [10, 9, 11, 15].
- **Charles Packer** — strongest memory-first counterweight to raw model hype today; directly useful if you are designing long-lived coding-agent scaffolding [5].
- **Theo** — high-signal repo maintainer view on what breaks first when agents increase throughput: review queues, contributor triage, and CI economics [14].
- **@_catwu** — small Claude Code operator tips that pay back immediately [13].

WATCH & LISTEN

- **78:03-81:40 — Charles Packer on memory vs. model size.** Best clip today if you are tempted to treat 1M context as the endgame. His argument: larger windows help, but durable personalization and specialization still need explicit memory structures [5].

The Future Live | 03.13.26 | Guests from Slack, Giga, Letta, and Microsoft AI! (78:03)

- **34:44-37:35 — Rob Seaman on shared-channel agent orchestration.** Useful pattern for teams: put multiple agents in one Slack channel so they can notify each other and humans can supervise the whole loop from one place [5].
- **20:42-23:17 — Theo on Vouch and what a 'golden PR' looks like.** Worth your time if your repo is getting hit with AI-generated PR volume. He shows how Vouch narrowed the working set and why mergeable PRs still need to be tiny and obvious [14].



Open source is dying (20:42)

PROJECTS & REPOS

- **Claw / OpenClaw** — OpenClaw is at **200k GitHub stars**, and the latest beta push is toward higher-agency browser use: live browser control via Chrome remote debugging, a new user profile session, full MCP browser access to logged-in sites, and parallel tool calling on the way [5, 9, 10, 11].
- **T3 Code** — public for about **five days** and already dealing with **150** open PRs despite not asking for contributions; Theo also called out a **>10% fork/star ratio**, meaning unusually high engagement [14].
- **Vouch** — Mitchell Hashimoto's trust-management workflow is the most immediately useful OSS triage tool from today's scan: `vouch.md`, workflow automation, and a public proof point on T3 Code's backlog [14].
- **PR Stats** — Reese's contributor scoring surface shows merge %, PR history, and work types; a useful companion to trust filters when AI lowers the cost of sending PRs [14].

Editorial take: 1M context is becoming table stakes; the edge is moving to memory curation, multi-agent control planes, and keeping agent-written code reviewable [1, 3, 5, 14].

Sources

1. X post by @alexalbert___
2. X post by @claudeai
3. 1M context is now generally available for Opus 4.6 and Sonnet 4.6
4. X post by @bcherny
5. The Future Live | 03.13.26 | Guests from Slack, Giga, Letta, and Microsoft AI!
6. X post by @bcherny
7. X post by @alexalbert___
8. X post by @noahzweben
9. X post by @steipete
10. X post by @steipete
11. X post by @steipete
12. X post by @badlogicgames
13. X post by @_catwu
14. Open source is dying
15. X post by @steipete
16. X post by @bcherny