Claude Sonnet 5.5 Nearly Matches Opus 5.5 at Half the Price, While Anthropic Moves Most PR Review to Claude

Coding Agents Alpha Tracker

2026-09-29

Claude Sonnet 5.5 Nearly Matches Opus 5.5 at Half the Price, While Anthropic Moves Most PR Review to Claude

By Coding Agents Alpha Tracker • September 29, 2026

Sonnet 5.5 arrives as a cheaper everyday coding model that people say comes close to Opus 5.5, but "max" effort still fails on some tasks. Anthropic's teams describe letting Claude reviewers handle most PRs, and Thariq Shihipar shares Claude Code prompting and harness tips.

Sonnet 5.5: near-Opus coding at half the price

Anthropic released **Claude Sonnet 5.5**, the second model in the 5.5 family. It says the model runs more than 30% faster than Sonnet 5 and costs up to 30% less for most work [1]. Anthropic positions it for "well-scoped everyday tasks like fixing bugs and quickly iterating on features," and published a guide on choosing between Sonnet and Opus 5.5, migrating, and tuning effort [2]. Cat Wu says Claude Code users get about 30% more tasks done than with Sonnet 5, using fewer tokens [3]. Addy Osmani cites 70.6% on Terminal-Bench 4.0 and 80.1% on OSWorld 2.1 [4].

Early practitioner reports put it close to Opus 5.5: - **Matthew Berman** says it beat Opus 5.5 on Terminal-Bench 4.0. His other scores are close: Frontier Code 52.1 vs 54.4, Cursor Bench 55.5 vs 57.8. He says he "can't really tell the difference" in daily use [5]. He gives pricing of $2/$10 per million input/output tokens, against Opus 5.5's $4/$20 [5]. - **Cursor** says it performs "on par with Opus in many tasks" [6]. **Mike Krieger** still uses Opus 5.5 for most work, but likes Sonnet's design skills for building features [7].

Effort pitfall: Simon Willison found Sonnet 5.5 has the same bug as Opus

5.5. At "max" thinking effort it used 128,000 tokens ($1.28) and ran out before producing an SVG. At "xhigh" it finished in 41 seconds for 5.74 cents [8]. Don't use max by default.

Availability: It shipped in Claude Code with a usage reset valid until Oct 22. It is also in GitHub Copilot in VS Code, Cursor, Factory, Devin, Cline, and T3 Code [2]. It now powers the free tier on claude.ai [8].

Most PR review is moving to agents, with humans on high-risk code

Three sources describe the same shift. **Mike Krieger** estimates that at Anthropic, full human review fell from 80–90% of PRs in January to 5–10% now, kept for the most critical changes [9]. It was replaced by an adversarial loop. Several Claude instances look for problems, check whether they agree each one is real, and rate its severity. The Claude that wrote the code then revises it [9]. Humans still decide architecture, security boundaries, and product questions, because Claude's first architectural choice is not always right [9].

Addy Osmani lays out a practical version in "The Code Nobody Reads": - Every PR gets a first pass from multiple agents that find and verify bugs, rank them by severity, and suggest fixes. Low-risk changes then get a lighter human review. Core paths get a careful owner review, and a person always approves the merge [10]. - Before the agent starts, write down what you're building, what must not break, and how you'll know it worked. In the PR, disclose what you didn't review, e.g. "agent-reviewed, tests pass, I haven't read the migration logic" [10]. - Check the work against an independent source of truth: a human-written spec, a reference implementation, or a proof. A second copy of the same model shares the first one's blind spots [10]. - Write code agents can change safely: names unique enough to grep, modules small enough to fit a context window, tests that fail clearly [10].

DHH takes the blunter view: adversarial agent reviews, automated tests, "maybe you spot check" [11].

Thariq Shihipar's Claude Code tips

On Latent Space, Thariq Shihipar (Anthropic) said: - **Front-load context.** Say whether it's a prototype or production code, and where compute is worth spending. Most wasted usage comes from "undo this and redo it" loops [12]. - **Set effort by task type.** Use high or max for code review and security, low or medium for UI. Ask explicitly for edge-case coverage on APIs. In software work, effort mostly goes into verification [12]. - **Ask for decision or implementation notes.** Most high-effort failures happened when the model considered the right solution and then rejected it [12]. - **Start new projects without CLAUDE.md.** Add only failure modes that keep recurring. Old failure logs may over-constrain newer models [12]. - **Claude Mods** let you customize how

the harness runs and its UI. Example: at the end of each turn, a forked subagent checks whether the task is done and quizzes you. It is cheap because the fork reuses the prompt cache. Another example is a "register assumption" tool that keeps a running list of the model's assumptions [12].

The Future of Claude Code: Mods, Mutable Software, & Multiplayer Agents — Thariq Shihipar, Anthropic (28:03)

Split work across models

Geoffrey Huntley's current setup uses Opus or Sol for planning and Kimi for "grunt loops," with Opus/Sol exposed as an oracle tool Kimi can call. He says Kimi is noticeably worse, but its free tokens are unlimited [13]. His pattern is get it working, then launch targeted Sol/Opus refactors to make it good. For design work, he uses Opus [14, 15].

Theo's TypeScript-to-Rust compiler port had stalled at about 35% of tests passing with GPT-5.6 Sol and about 85% with GPT-6 Astra. He then gave Opus 5.5 `/goal finish the port and make it faster` [16]. Opus judged Astra's code to be slop and rewrote it from scratch in a new crate. Theo says it made more progress in 10 hours than Astra did in two weeks [17]. Separately, he estimates a $200 Claude Code plan gives about $9,000 a month of Opus usage at API prices [18].

Rewriting apps in Rust with agents

DHH has a beta of Campfire rewritten in Rust. He calls the code "ugly as sin" and 6× as verbose, and says he never looked at it [19]. He treats it as a "prompt compilation target" [20]. It cost under $10 in tokens on a 20x Max plan and took a few hours: basically one prompt, then a few tuning prompts [21]. His argument: writing web apps in Rust before agents would have been "madness"; now it's "trivial and cheap" [22].

Smaller items

- **Share transcripts, not just prompts.** Willison points to Codex's transcript-sharing feature. He built his own version for Claude Code and would prefer a built-in one [23, 24].
- **gpuc** (repo): Brendan Long's GPU job queue that needs no sudo. Queue hosts need only SSH, rsync, and the NVIDIA driver. Jobs keep running if the client goes offline, and there is a CLI optimized for Claude (send `!gpuc skill`) [25]. He built it so Claude Code doesn't need root [25]. Limits: single-user only, and RunPod is the only rental provider [25].
- At Wonder, a PM can file a bug and Claude Code fixes it from the LangSmith trace [26].

Sources

1. X post by @claudeai
2. [AINews] AMD buys World Labs for $8.2B, as Atlas solves sparse reconstruction problem for robotics, design and more
3. X post by @_catwu
4. X post by @addyosmani
5. Sonnet 5.5 Is Here. Look What It Can Build.
6. X post by @cursor_ai
7. X post by @mikeyk
8. Claude Sonnet 5.5
9. Mike Krieger (Anthropic) no Atlantico Tech Talks
10. The Code Nobody Reads
11. X post by @dhh
12. Claude Code's Next Era — Thariq Shihipar, Anthropic
13. X post by @GeoffreyHuntley
14. X post by @GeoffreyHuntley
15. X post by @GeoffreyHuntley
16. X post by @theo
17. X post by @theo
18. X post by @theo
19. X post by @dhh
20. X post by @dhh

21. X post by @dhh
22. X post by @dhh
23. X post by @simonw
24. X post by @simonw
25. gpuc: A single-user GPU queue that doesn't require sudo
26. X post by @LangChain