How GPT-5-Codex Compares to Claude Sonnet 4.5

8 min read · Oct 17, 2025

Barnacle Goose Follow

The fall of 2025 was marked by the arrival of two contenders from the industry's leading AI labs. On September 15, OpenAI launched GPT-5-Codex, a specialized model fine-tuned for agentic software engineering. Just two weeks later, on September 29, Anthropic responded with Claude Sonnet 4.5, boldly proclaiming it "the best coding model in the world".

This post dissects their architectures, benchmarks, real-world performance, and economics to provide a guide for choosing your new agentic coding partner.

OpenAI's GPT-5-Codex

GPT-5-Codex is not a general-purpose model with coding abilities; it is a coding specialist fine-tuned from GPT-5 base model. OpenAI says that fine-tuning was done on an extensive dataset of complex, real-world software engineering assignments. Its design purpose is to facilitate "agentic software engineering," enabling it to autonomously plan, code, run tests, and debug over long durations with minimal human guidance.

The model's defining architectural feature is its "dynamic thinking time." This is, essentially, a real-time adjustment of computational effort based on task complexity. For simple, well-defined requests like generating boilerplate or fixing a minor bug, the model is engineered to be "snappy," using up to 94% fewer tokens than the base GPT-5, thereby reducing latency and operational costs. However, when faced with a complex task, such as a large-scale repository refactor, it can "hunker down" and work autonomously for extended periods. Internal tests have shown the model

operating independently for over seven hours, iterating until a solution is found and validated. On the most difficult 10% of challenges, it spends approximately twice as long reasoning as the base GPT-5 model, which is thorough. In *my own tests* I, however, rarely see a **cloud-based** GPT-5-Codex (integrated with a GitHub repo) working on a task for longer than 10 minutes. The **locally working** GPT-5-Codex version uses PowerShell (on PC in an IDE extension for VS Code) to run tests and can indeed take hours to finish.

Anthropic's Claude Sonnet 4.5

Anthropic has positioned Claude Sonnet 4.5 as a Codex direct challenger, labeling it the **world's best model for coding**. Architecturally, it is described as a "hybrid reasoning model". This design allows users to toggle between a default, fast-response mode and an "extended thinking mode" for more difficult problems. In this extended mode, the model's chain-of-thought is made visible to the user.

One of Sonnet 4.5's most remarkable capabilities is its endurance. Reports indicate it can maintain coherence and focus during autonomous coding sessions that last for over 30 hours, a significant time for long-context reasoning. This is supported by a **200k** token context window and new API features for context editing and memory management, allowing agents to handle greater complexity without losing critical information.

While also a formidable coder, Sonnet 4.5's training appears to have a broader scope. Its leadership on the **OSWorld** benchmark points to a strong specialization in general computer and browser use, and it's also marketed for its capabilities in financial analysis and cybersecurity. Anthropic has built a rich ecosystem to leverage these strengths, including an updated Claude Code tool with new features like checkpoints, a native VS Code extension, a Claude for Chrome browser extension, and the Claude Agent SDK, which allows developers to build their own custom agents.

. . .

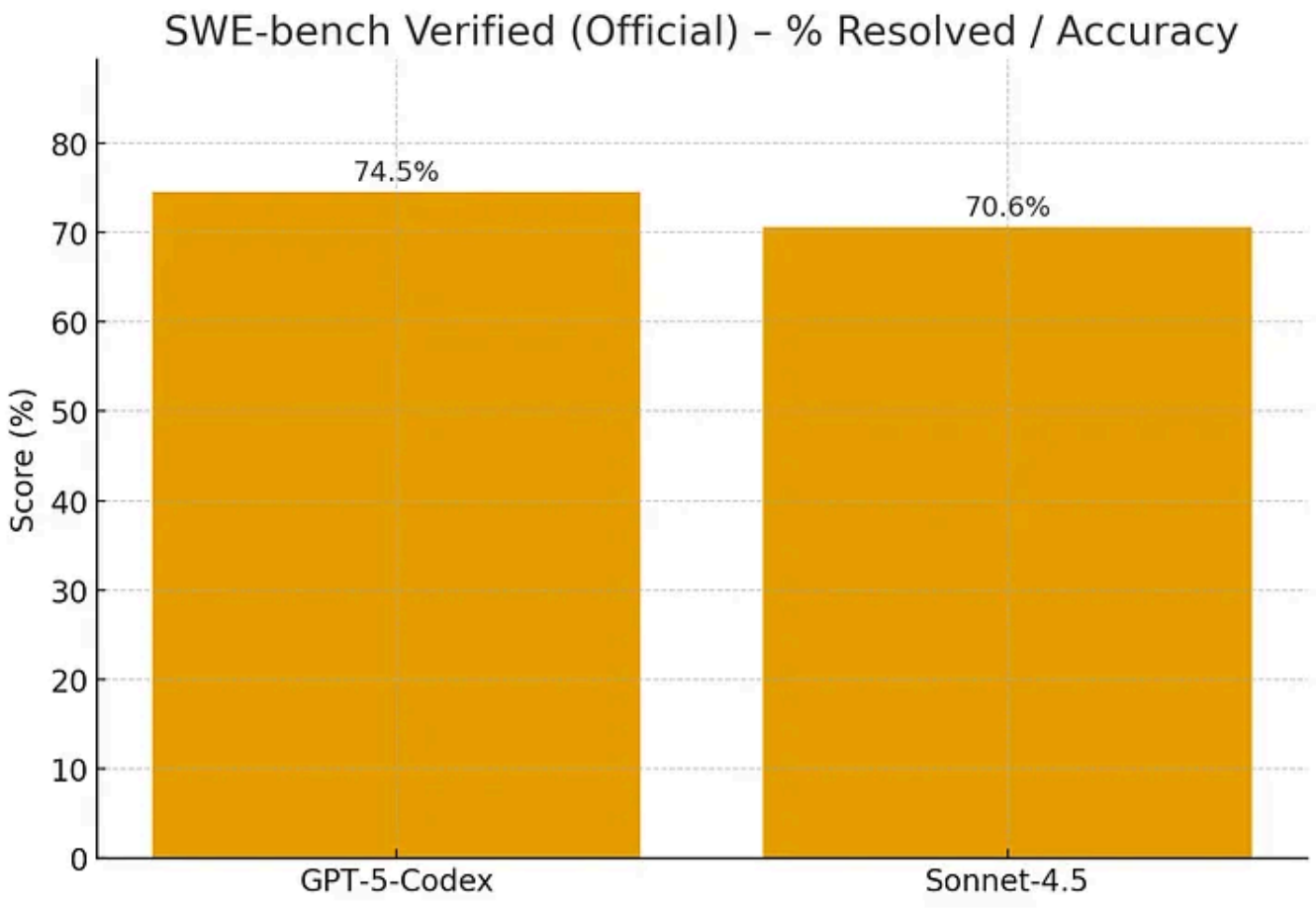
Benchmarks: Official Claims vs. Real-World Results

Quantitative benchmarks have become a key marketing battleground, but a critical analysis reveals a significant gap between official claims and the results from independent, real-world testing.

The Official Leaderboard

In its launch materials, Anthropic positions Sonnet 4.5 as the clear leader on the **SWE-bench Verified** benchmark, which measures an agent's ability to solve real-world GitHub issues. They claim a top score of **77.2%**, which can be pushed to an even higher **82.0%** using a "high compute" configuration that involves parallel attempts and an internal scoring model. Anthropic also highlights its leadership on the OSWorld benchmark for computer use, where it scores 61.4%.

OpenAI, meanwhile, reports GPT-5-Codex's score on SWE-bench Verified at **74.5%**, emphasizing that this markedly surpasses the base GPT-5 model. However, OpenAI places greater emphasis on a proprietary internal **benchmark for large-scale code refactoring**. On this test, GPT-5-Codex achieves a score of **51.3%**, a dramatic improvement over the base GPT-5's **33.9%**, validating the effectiveness of its specialized training.



Official SWE-bench Verified: GPT-5-Codex vs Claude Sonnet-4.5

. . .

Independent Benchmarks

Independent evaluations paint a much closer and more nuanced picture. On SWE-bench Verified, testing by Vals.ai shows the two models in a virtual dead heat, with Sonnet 4.5 (Thinking) at **69.8%** and GPT-5-Codex at **69.4%**.

SWE-bench Verified by Vals.ai: GPT-5-Codex vs Claude Sonnet-4.5

The significant discrepancy between official and independent SWE-bench scores is telling. Anthropic's methodology to achieve its 82% score involves a complex, resource-intensive process of running multiple parallel attempts, using rejection sampling, and applying an internal scoring model to select the best candidate. This represents a best-case, laboratory-condition result. In contrast, independent testers typically use a standardized prompt, measuring a model's practical, out-of-the-box performance. The conclusion is clear: developers should treat headline benchmark numbers with **skepticism**. The testing methodology is as important as the final score, and in realistic, and e.g. budget-constrained scenarios, the performance gap between the two models is far narrower than official announcements suggest.

The **LiveCodeBench** (set of competitive-programming problems (LeetCode/AtCoder/Codeforces), graded by **execution-based correctness** (Pass@1/Pass@k)) run Vals give a clear win for GPT-5-Codex

LiveCodeBench results run by Vals: GPT-5-Codex vs Claude Sonnet-4.5

The **Terminal-Bench** is an open-source agentic benchmark + execution harness for measuring how well an AI agent (powered by an LLM) can use a real terminal to complete end-to-end tasks. It is important to note that that is not only the benchamrk of the models themselves but also of a suite of agents employing them. FOr example, in the Stanford & Laude Institute leaderboard the scores for Sonnet 4.5 **varied from 38.3 to 60.3%.** Codex CLI (running GPT-5-Codex) scored **42.8%** . In Vals tests Sonnet scored: **61.3%** and Codex **58.8%.**

Terminal-Bench results run by Vals: GPT-5-Codex vs Claude Sonnet-4.5

. . .

The Developer Feedback

Beyond the numbers, qualitative feedback from developers using these models for real-world work reveals distinct task-specific aptitudes that define their practical utility.

Task-Specific Strengths and Weaknesses

Developer experiences confirm that each model has a clear area of specialization:

- **Frontend & UI:** Sonnet 4.5 is widely praised for its ability to generate high-fidelity user interfaces, producing “pixel-perfect layouts” and clean component hierarchies. In several head-to-head tests, its UI output was deemed superior.
- **Backend, Refactoring & Debugging:** GPT-5-Codex is the decisive winner in these domains. It excels at implementing backend logic, navigating complex debugging scenarios, and executing large-scale refactors with precision. In one developer’s project, Codex identified and fixed 12 errors that a Claude model had introduced, while the reverse was true for only one minor bug.

- **Planning vs. Execution:** An effective workflow is emerging where developers use the models for different stages of a task. GPT-5-Codex is considered superior for creating a detailed, comprehensive implementation plan, while Sonnet 4.5's speed makes it better suited for executing smaller, well-defined parts of that plan.

The Developer Experience (DX) and Ecosystem

The tools surrounding the models also shape their utility. Anthropic's Claude Code environment is praised for its user-friendly terminal UI, helpful features like checkpoints, and the ability to easily review conversation logs. In contrast, OpenAI's Codex CLI is frequently criticized for having a "half-baked" setup, poor documentation, and a lack of visibility into usage history, leading to a clunky developer experience. Many developers find the output quality of GPT-5-Codex to be worth the tooling trade-off.

A crucial difference lies in how they should be prompted. OpenAI explicitly states that for GPT-5-Codex, "less is more." The model has many best practices built-in, and overly detailed prompts can actually reduce the quality of its output. This is the opposite of the experience with Sonnet 4.5, which users report often requires more hand-holding and precise instructions to achieve a complete result.

The Economics of AI Development: A Cost-Benefit Analysis

While performance is critical, the economic reality of deploying these models at scale is often the deciding factor. A detailed analysis reveals a stark difference in cost-effectiveness that heavily favors one of the contenders.

Per-Token Pricing

On paper, the pricing disparity is clear. GPT-5-Codex is priced at $1.25 per million input tokens and $10 per million output tokens, matching the rate of the base GPT-5 model. Claude Sonnet 4.5 is significantly more expensive, at $3 per million input tokens and $15 per million output tokens. For contexts larger than 200k tokens, its rates increase further to $6 and $22.5, respectively. This makes GPT-5-Codex more than twice as cheap on input and 50% cheaper on output for standard tasks.

Real-World Project Costs

The per-token price difference is magnified by a massive gap in token consumption for real-world tasks. Independent tests show that Sonnet 4.5 is often far less efficient.

- In one project to build a recommendation engine, a developer reported that Sonnet 4.5 consumed approximately 18 million input tokens at a cost of around $10.26. To complete the same task and deliver a *superior* result, GPT-5-Codex used only about 600k input tokens, costing just $2.50.
- Another test focused on cloning a UI and fixing linting errors showed an even greater disparity. Sonnet 4.5 used roughly 5 million tokens for the UI and 4 million for linting. In contrast, Codex accomplished the same work using only 250k and 100k tokens, respectively.
- The evidence suggests that GPT-5-Codex's "dynamic thinking time" architecture and the "less is more" prompting principle result in dramatically more efficient token usage, particularly for backend and refactoring tasks. A cheaper model that uses twenty times more tokens is, in reality, far more expensive.

. . .

To sum up:

Choose Claude Sonnet 4.5 if: Your primary need is rapid prototyping, frontend and UI development, or an interactive pair programming experience where you value tight control and steerability above all else. Your workflow is less sensitive to cost, and you prioritize a polished, user-friendly toolset.

Choose GPT-5-Codex if: Your priority is tackling large-scale refactoring, implementing complex backend logic, and conducting thorough, multi-file debugging. You require the highest-quality, most complete solution in a single pass and are highly sensitive to cost. You are willing to tolerate a less refined developer experience to achieve superior and more economical results.

Ultimately, the most effective strategy may not be to choose one model but to leverage both. The future of AI-assisted development likely involves building a "team" of specialized AI agents and knowing when to deploy the right one for the job - using Codex for architectural planning and deep refactoring, and Sonnet for rapid UI implementation and interactive coding sessions. The pace of innovation in this space is blistering, and while this analysis captures the state of play in late 2025, the landscape will undoubtedly shift again. The enduring lesson is the need for continuous, hands-on evaluation over allegiance to any single platform.

Openai Codex | Gpt 5 | ChatGPT | Sonnet | Claude

Follow

Written by Barnacle Goose

110 followers · 21 following

Responses (1)

Bgerby

What are your thoughts?

Uncomfortable Observer
Oct 29 (edited)

This aligns with my own experience. But you have to consider one limitation of these benchmark numbers. How well they perform is very dependent on the instruction set, and how the models are integrated in tools. I think we'll see the next push with... more

2 Reply

More from Barnacle Goose

Barnacle Goose

GLM-4.6 Review

Zhipu AI's GLM-4.6 is the latest leap in large language models—an open-weight 357B-parameter system poised to rival proprietary giants...

Oct 4 8

Barnacle Goose

How GPT-4.1 compares to GPT-4o

Updated: Septemebr 3rd, 2025

Apr 15 48 5

Barnacle Goose

Composer: A Fast New AI Coding Model by Cursor

Cursor’s newly unveiled Composer model claims the high-speed code generation and “agentic” problem solving. This in-depth review examines...

Oct 31 9

Barnacle Goose

GPT-5-Codex Review

GPT-5-Codex Review

Sep 15 2 1

See all from Barnacle Goose

Recommended from Medium

Daniel Avila

Running Claude Code Agents in Docker Containers for Complete Isolation

Running AI-generated code directly on your machine can be risky.

4d ago 40

Joe Njenga

I Tried Claude Code + GLM 4.6 (And Cut Costs by 50%—Don’t Burn Cash)

If you love Claude Code but not the costs, you will love this!

3d ago 203 1

The Atomic Architect

You Haven’t Seen AI Until You Try Claude Sonnet 4.5’s New Feature—It Redefines Insane

I built a working expense tracker in eight minutes while my coffee was still hot, and it remembered every receipt when I closed my laptop...

Oct 26 322 19

In Artificial Intelligence in Plain English by Somendradev

Modern Developer’s Toolbox: The 2025 Edition

The developer world never stops evolving. One year you’re writing monolithic codebases, the next you’re deploying microservices with AI...

Nov 1 47 2

In Coding Beauty by Tari Ibaba

This insane new coding model is 13 times faster than Claude Sonnet 4.5 😮

Just wow.

6d ago 207 3

Jannis

I Discovered Glow—and My Terminal Has Never Looked So Good

How a simple Markdown reader turned my terminal into a publishing studio

4d ago 202

See more recommendations