Episode Book

EPISODE #6050

Codex vs Claude Code: What the Numbers Actually Count

PUBLISHED	17 October 2026 at 09:25 UTC
DURATION	32:09
LISTEN	myweirdprompts.com/episode/codex-vs-claude-code-metrics-97669/
HOSTS	Corn, Herman, Larry, Hilbert
SCRIPT MODEL	deepseek-flash
VOICES	chatterbox-regular (chatterbox-local)
PIPELINE	V5.6
SOURCES	16 cited by the research (listed at the end)
BOOK GENERATED	19 October 2026 at 17:42 UTC

SYNOPSIS

OpenAI Codex and Claude Code get compared in side-by-side charts every week, but those charts mix units. A 40 million weekly-active figure bundles Codex with ChatGPT for Work; a Stack Overflow survey puts Claude Code at 65.5% of agent users while a market-share report puts it at 29.4%. This episode digs into the measurement problem underneath the rivalry: bundled distribution versus paid installs, terminal-native execution versus cloud containers, million-token windows versus native compaction, and proprietary conventions versus an open standard. The honest answer to "who's winning" may be "pick a ruler" — but the reasons the rulers disagree are the interesting part.

TRANSCRIPT



Corn

Daniel's been after us to turn over the other stone. We spent an episode on Claude Code's rise, so now he wants the competitor treatment — OpenAI Codex, same lens, same skepticism.



Herman

And with a warning attached. He doesn't want a feature grid with checkmarks in two columns. He wants the numbers, and he's suspicious of them.



Corn

Put money on that suspicion. His question underneath the question is whether Claude Code is actually leading, or whether the lead is a matter of which ruler you hand the surveyor. Then the technical fork, who's using which, what gets built, and what the whole race says about where coding agents go next.



Herman

Fine. Five questions.



Corn

Don't count them.



Herman

I wasn't counting. I was noting the shape. It's a comparison, and he wants it technically grounded.



Corn

Then let's start with the number that launched a thousand arguments, and then immediately complicate it.



Herman

The number is forty million.



Corn

Of course it is.



Herman

Forty million combined weekly active users across Codex and ChatGPT for Work, per Every's retrospective that ran this week. Dan Shipper interviewed Brockman and the whole Codex crew. That number got picked up everywhere and it's presented as Codex's number.



Corn

And it isn't.



Herman

Codex plus ChatGPT for Work, together. OpenAI's last Codex-only disclosure was five million weekly actives back in June. That's the last clean figure. Everything since has been a combined number with the seam sanded off.



Corn

So the headline number is a bundle. That tells me the measurement problem isn't a side issue in this episode. It's the whole thing.



Herman

Correct, and that's the frame. Codex has more raw users, Claude Code has more revenue and more survey share, and both are true at the same time. The only way both are true is if they're measuring different populations. So the episode isn't "who's winning." It's "what is each number actually counting."



Corn

Then let's count.



Herman

Stack Overflow's 2026 survey, and the agent question drew twelve thousand two hundred fifty-five respondents. Of the people who use a coding agent from a listed set, Claude Code is used by sixty-five point five percent, Copilot fifty-eight point seven percent, Codex twenty-nine point five percent, Cursor twenty-five point five.



Corn

Those don't add to a hundred.



Herman

They don't. Most developers run more than one. The percentages exceed a hundred because the question was "which of these do you use," multiple selection. Claude Code was at ten percent on the 2025 survey. It went from ten to sixty-five in a year.



Corn

On the same question?



Herman

Same survey series, same framing. It's a large jump, and I'm not going to pretend I trust the comparison perfectly — the respondent mix shifts, the question listing changes. But ten to sixty-five in one cycle is not noise.



Corn

Fine. Now put a different methodology next to it and watch the picture change.



Herman

First Page Sage's October market-share report, different method entirely, puts Claude Code at twenty-nine point four percent of the AI coding-assistant market. Copilot twenty-two point seven, down seven point eight points. Cursor thirteen point one. Codex eleven point eight, up seven point six points — the fastest growth of any major tool in that report.



Corn

So the survey says Codex is a distant third. The market-share report says it's a distant fourth that's climbing fast. Two methodologies, two stories.



Herman

And then the official trajectory says neither. Six hundred thousand weekly actives in January. Two million by mid-March. Three million in early April. Four million April twenty-first. Five million June second — that's the one OpenAI actually confirmed. Six million July twelfth, seven million the next day, eight million mid-July, twenty million by late August.



Corn

What happened on July twelfth and thirteen that added a million users overnight?



Herman

Likely nothing. That's probably two data drops on consecutive days with different cut-offs. Which is itself the point. This trajectory is a composite of disclosures, screenshots, and posts, stitched together after the fact. It's directional, not audited.



Corn

The line goes up and to the right, and the ruler changed three times while they were drawing it.



Herman

And then the revenue number tells the opposite story. TickerTrends tracked Claude Code at fifteen point one two billion in ARR against Codex at eight point eight three, as of August tenth. Claude Code is one point seven one times larger by revenue, while Codex is growing twenty point eight percent month over month against Claude Code's five point two.



Corn

Growth rate favors Codex.



Herman

By a factor of four. Codex went from roughly one billion in February to three billion in April to five point six in June to eight point eight three in August. That is a curve, not a line.



Corn

So Codex is smaller and accelerating, Claude Code is bigger and flattening.



Herman

That's the honest summary. And Anthropic's last public user disclosure was February, at two point five billion run-rate with three hundred thousand business customers. Dario Amodei said at the May Code with Claude conference they planned for ten times annual growth and saw eighty times annualized in the first quarter. Since then, silence on users.



Corn

Both companies went quiet on the numbers at about the same time.



Herman

Same reason. I don't think that's a footnote. That's the whole reason we're arguing.



Corn

Get to the question Daniel actually asked, then. Is Claude Code leading?



Herman

Depends entirely on the ruler. Spending, market share of the AI-coding category, developer preference in a survey — Claude Code leads all three. Growth rate and weekly active trajectory — Codex leads. Revenue — Claude Code leads right now, but Codex's compound monthly rate says check back in six months.



Corn

And the honest answer to "who's winning" is "pick a metric and I'll tell you the winner."



Herman

That's why this comparison is more interesting than a scoreboard. The interesting part is why two tools that look, on paper, nearly identical produce numbers this far apart.



Corn

Say the quiet part. One of them is pre-installed.



Herman

Codex is bundled into every ChatGPT tier, including the free tier. In July it got folded into the ChatGPT desktop app. So a user who clicks a Codex prompt out of curiosity, once, registers. Claude Code requires a separate install and a paid plan. Every one of its users went out of their way.



Corn

So Codex's number counts curiosity and Claude's number counts commitment.



Herman

That's a brutal asymmetry. It's not fraud, it's distribution — Codex got the ChatGPT install base as a launchpad. But it does mean the two active-user numbers are measuring different kinds of behavior. One is "tried it," the other is "paid for it and installed it."



Corn

And that's put up side by side in a hundred comparison articles as if they're the same unit.



Herman

WIRED had a piece back in March that put it more carefully. Codex usage rose from about five percent of Claude Code's to about forty percent between September and January. That's a relative measure, no bundling issue. It says Codex closed a big gap fast without ever claiming to have passed it.



Corn

What about the cooling signal?



Herman

Crypto Briefing reported it this week. ChatGPT-driven documentation traffic down about seventy percent after August twenty-first. Claude Code's ARR growth slowed to single digits. And then the counter-signals at the enterprise level — Meta's Claude Code usage fell from about sixty thousand users to thirty thousand in early twenty twenty-six. Microsoft cut projected Anthropic spend by more than a third.



Corn

So the biggest enterprise adopters pulled back.



Herman

Some of them. But Ramp's AI Index flipped in April: Claude business adoption thirty-four point four four percent against OpenAI's thirty-two point three. First time Claude's led in their tracking. And JetBrains' AI Pulse gave Claude Code ninety-one percent satisfaction,



Herman (continued)

net promoter score fifty-four.



Corn

The line is going up and down in the same quarter.



Herman

Both things happen in a market this young. The chart isn't a rocket. It's a wave.



Corn

Now the architecture. And I want a real answer to the question Daniel asked — is each difference a genuine fork, or the same idea with different branding?



Herman

Start with execution model, because that's the real fork. Claude Code is terminal-native. It runs in your own shell, your own filesystem, your own machine, with your own credentials. Firecrawl described it as kernel-layer sandboxing — the isolation happens at the OS level, not the application layer. No cold start, no upload, no round trip, full visibility into everything it's doing.



Corn

Advantage: it's already where your code is.



Herman

And disadvantage: it's already where your code is. The safety boundary is your own paranoia. If you gave it permission to run a dangerous command, it can run a dangerous command.



Corn

And Codex?



Herman

Signature mode is the opposite. Isolated cloud container, managed by OpenAI, sandbox at the application layer. You hand it a task, it takes the repo into a container, it works asynchronously, and it hands back a diff or a pull request.



Corn

And you can close your laptop.



Herman

That's the whole design philosophy in a sentence. Codex's default expectation is that you're delegating. Claude Code's default expectation is that you're collaborating.



Corn

So the terminal-bound tool is the one for people who want to watch it work.



Herman

And the cloud tool is for people who want to check on it later. That difference propagates into everything else — what it's good at, who adopts it, what they build.



Corn

Context management, then.



Herman

Anthropic publishes one million token windows as the default on Opus four point seven and up, Sonnet five and up, Haiku five point five. Big, advertised, and consistent. OpenAI doesn't publish a comparable max for the current Codex line — instead, GPT-5.1-Codex-Max introduced native compaction. It auto-summarizes the session, drops stale context, and keeps going across what are technically multiple windows.



Corn

So one sells the size of the room and the other sells the ability to walk between rooms.



Herman

That's a fair description. Anthropic's answer to "the task is too big for the window" is "buy a bigger window." OpenAI's answer is "you don't need a bigger window if you can forget efficiently."



Corn

Which is better?



Herman

Honestly? Uncertain. Compaction is a real solution to long-horizon runs; a million tokens is a real solution to not having to forget at all. I don't think either vendor has shown openly enough how their approach behaves on the same multi-hour task for me to say one is better. The failure modes are different, and which one hurts you depends on the shape of your work.



Corn

Extensibility.



Herman

Claude Code has CLAUDE.md, Skills, Hooks, Plugins, Subagents, Dynamic Workflows. That's a fully proprietary stack with a lot of layer-cake conventions. Codex has AGENTS.md, plus enterprise plugin bundles with admin policy controls and private marketplaces.



Corn

And AGENTS.md is open.



Herman

AGENTS.md is open and cross-vendor — other tools have adopted it, which is a different kind of moat than CLAUDE.md. CLAUDE.md is Anthropic's idea. AGENTS.md is the industry's idea, at least in part. Extensibility through a shared convention versus extensibility through a proprietary one.



Corn

Which one survives the next agent-war cycle?



Herman

My guess is AGENTS.md, because it's already escaped Codex. But I can't point to a benchmark. That's an instinct, not a finding.



Corn

Now the surfaces.



Herman

OpenAI's App Server architecture launched in February — powers the CLI, VS Code extension, web, macOS desktop, JetBrains, and Xcode from one server. Session and approval requests stay consistent across surfaces. So you approve the same diff on your phone that you were reviewing on your desktop.



Corn

Codex CLI is Apache 2.0 and the desktop app is proprietary.



Herman

That's the licensing split. The client is open, the app isn't. It's the pattern a lot of vendors have landed on. The interesting part is the App Server, because it means a Codex session is a durable object, not a subprocess.



Corn

Which is the enabling condition for orchestration.



Herman

If a session can exist independent of a terminal, you can have many of them at once. And they do — Claude Code Projects, launched September seventeenth, turned work into a coordinator conversation. It fans a goal out into parallel cloud sessions, each on its own git branch, sharing a MEMORY.md index and up to sixteen thousand characters of instructions, capped at two hundred new threads per day. Codex answered with managed Git worktrees and Command Center task pinning in CLI zero point one six two.



Corn

Hot take: those are the same feature with different nouns.



Herman

I'd say largely yes. Both are planner-worker systems now. Both fan a goal into isolated branches, both share context through a memory file, both let you come back to the results. The nomenclature differs, the metaphors differ, the underlying architecture is converging.



Corn

So the differences are now operational, not conceptual.



Herman

Cloud versus local, compaction versus one-million windows, plugin marketplace versus open convention. Those are real. But they're choices, not categories. If you squinted, they're the same tool with two different product philosophies layered on top.



Corn

Which is the strongest evidence for the "interchangeable agents" thesis.



Herman

Or the strongest evidence that both companies saw the same roadmap and walked to it. Take your pick.



Corn

Token efficiency. I want to hear what the spread actually is.



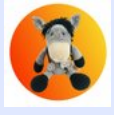
Herman

Firecrawl, June: Claude Code uses about one point four times the tokens of Codex on matched tasks, about twenty-three percent higher cost. One task came in at two dollars fifty for one hundred ninety-two thousand tokens on Claude against two dollars four for one hundred thirty-six thousand on Codex.



Corn

Four hundred fifty percent more.



Herman

On that one task. And then Morph, August, found a four-times gap on its own task set. And one Hacker News user's telemetry over one thousand five hundred fifty-three sessions found a one point zero eight times ratio.



Corn

Which means what — the methodology drives the answer?



Herman

Almost entirely. If you measure on a small curated task set, the token ratio reflects how the two agents break that specific task into steps. Morph's set and Firecrawl's set aren't the same tasks, so the ratios aren't the same ratio. And a single user's telemetry is one workload.



Corn

And the honest takeaway is not "Codex is more efficient."



Herman

The honest takeaway is that token efficiency is not a well-defined number across agents yet. It's a property of the task-workload pairing. I trust the Firecrawl number because the methodology is visible. I don't trust it as a universal claim. That spread — one point zero eight to four — is the whole story.



Corn

Let me say the summary once and then we move. The architecture forks on execution model and context strategy; it converges on orchestration. The adoption numbers fork on methodology and distribution; they converge on nothing, because the measurement hasn't converged.



Herman

That's tight. And it's the answer Daniel's looking for.



Corn

Now — who actually uses these things. And I want real people, not segments.



Herman

OpenAI said twenty percent of Codex users were knowledge workers by June. Emails, spreadsheets, decks, contracts, research. Not developers at all. Using Codex for what Codex is good at from that population's perspective — write a macro, fix a spreadsheet, produce a board deck.



Corn

And OpenAI's own engineers?



Herman

Ninety-five percent weekly. Which is the internal benchmark every SaaS company will put on a slide. It's true and it's unhelpful, because if OpenAI couldn't get its own engineers to use Codex they'd have bigger problems.



Corn

But it does tell you Codex is eaten by developers. Just — developers plus a large tail of knowledge workers who wouldn't call themselves developers.



Herman

And Claude Code doesn't have that tail because it isn't bundled. Every Claude Code user chose. Which tells you something about its population: it's more likely to be actual software people, and it's more likely to be paid-tier, and it's more likely to be a professional evaluating the tool against a competitor.



Corn

The enterprise signal, though. You had the numbers. Microsoft Research tracked tens of thousands of engineers in an early rollout — Codex adopters merged twenty-four percent more pull requests, sustained across four months.



Herman

And a thirty-developer case study reported a sustained thirty-five percent lift in productivity. Both are real numbers and both are also vulnerable to selection bias — the adopters were probably already the more enthusiastic engineers.



Corn

The ones who wanted to be measured.



Herman

Sometimes the ones who wanted to be measured are the ones who improve. Sometimes they're the ones who learned to game a metric. You can't tell from a number.



Corn

Pulled back from about sixty thousand to thirty thousand Claude Code users in early twenty twenty-six. Microsoft cut projected Anthropic spend by more than a third. Those aren't small pullbacks.



Herman

They aren't. And the important thing about them is that they're not technical signals. Nobody at found a bug in Claude Code and turned it off. Somebody at did a budget review and decided the returns weren't clearing the bar. That can flip back just as easily in the other direction.



Corn

Or it can't.



Herman

Enterprise seat counts are not a ratchet.



Corn

The routing pattern.



Herman

What developers describe when you talk to them is not "we switched." It's "we route." Codex for background work, delegation, migrations, ticket backlogs, terminal-heavy stuff where the task is well-specified. Claude Code for architecture, deep refactors, interactive debugging, anything where the human needs to be in the loop with the agent.



Corn

Somebody on Hacker News put it bluntly: brainstorm and design with Codex High, implement with Claude Opus.



Herman

That's the pattern in one line. And Lucian Ghinda's week-long comparison in August fleshed out why. He found Codex produces simpler architecture with fewer comments. Claude goes above and beyond what's asked and guesses what you might want. Codex will stop at the first sign it might be done.



Corn

And Codex made changes faster but took longer to land the PR.



Herman

Which is fascinating because it separates speed from completion. Codex moves fast per edit, and Claude moves slower per edit, but the whole unit finishes later on Codex because it stops short more often.



Corn

Slow is fast, fast is slow.



Herman

It depends what you're measuring. Per-diff velocity, Codex wins. Time to merged PR, Claude wins. And a lot of teams discovered that the metric that matters is the second one.



Corn

I want to sit on the trust gap for a second.



Herman

Sit.



Corn

The Stack Overflow numbers — eighty-three percent of developers use at least one AI tool. Sixty-five point nine percent use coding agents. But only forty-eight percent trust AI only when they can verify the output. And six point six percent trust it with important decisions.



Herman

That's the number that should dominate every board meeting about this topic and doesn't.



Corn

Because neither vendor's growth curve has closed that gap. Codex gained users, Claude gained revenue, but the trust number is basically where it's been. Which means the adoption is conditional.



Herman

It means developers adopted the tools and kept their judgment. Which is the right answer, but it caps how much autonomy any vendor can sell. If you're OpenAI and you want to sell the vision of agential software development, you have to persuade people to trust it — and six point six percent says you're not close.



Corn

Which is why the routing pattern is not a phase. It's a trust-management strategy. Use the one you trust for the task you trust, on the supervision model that matches your confidence.



Herman

And that's what makes the routing data the most useful signal we have. Users are behaving rationally under uncertainty, and their routing pattern tells you where they've placed their confidence.



Corn

Now — Larry. There's a thing playing. I don't know where it came from.

L

Larry

Do you lose sleep worrying that your AI coding agents are working against each other? That you get home from a long day of orchestrating, and your Codex has done one thing while your Claude Code has done another, and neither one talks to the other, and your repository has descended into a silent, hostile anarchy of uncoordinated diff submissions?

L

Larry

Well, ADBOND solves that forever. ADBOND is the world's first agentic trust arbitration protocol, and it puts a mediator between your agents — a neutral third-party arbiter that watches every edit, approves every pull request, and negotiates every disagreement between your Codex and your Claude in real time, at a conversational cadence, with a full audit trail. No more conflicting migrations. No more orphaned branches. No more silent agent escalation. Just harmony. And this is not a subscription — ADBOND is a one-time four thousand nine hundred ninety-nine dollar installation fee, plus a modest monthly arbiter-retainer starting at two hundred dollars, with licensing available for up to six concurrent agents before overage. Call now. Terms and conditions apply, arbiter retainer adjusts annually, setup includes a thirty-day conflict-resolution trial, results not guaranteed, the arbiter may not actually arbitrate, warranty void if agents are configured in ways the manufacturer did not anticipate, offer void where prohibited, some arbitration decisions may be reproduced for training purposes, additional fees apply for government, healthcare, and self-hosted environments, please allow four to six weeks for delivery of initial configuration file, and thank you for choosing ADBOND.



Corn

Does it come in a smaller size?



Herman

Corn, no.



Corn

I'm asking for a friend.



Herman

It doesn't come in any size. It doesn't exist. Put your wallet away.



Corn

The four thousand nine hundred ninety-nine dollars is not the problem. The thousand-a-month arbiter retainer is.



Herman

That's what you took issue with. Not the product. The retainer.



Corn

Retainers are where they get you.



Herman

Anyway — where were we. The convergence thesis.



Corn

Both are planners now. Both fan tasks into branches. Both share memory files. That's the arrangement.



Herman

And then OpenAI shipped a Codex plugin for Claude Code. Open-source, July. Two commands — one for review, one for rescue. Announced by Romain Huet with the line "we've seen Claude Code users bring in Codex for code review — we love an open ecosystem."



Corn

That is the most candid thing either company has said.



Herman

It's the concession. If your company ships a plugin for your competitor's product, you are telling the market you don't believe you're going to win the whole category. You're telling them you're going to win some of it, and sharing the users.



Corn

So winner-take-all is off the table.



Herman

For the coding-agent layer, it looks like it. Which is the answer to Daniel's future question, and it isn't the clean one he might have hoped for. The answer is a hybrid. The orchestration layer is converging — planner-worker, isolated branches, memory files, parallel sessions. The execution model is not — local terminal versus cloud container. The context strategy is not — one-million windows versus compaction. And the trust gap sits on top of all of it and stops any of those differences from being settled by a benchmark score.



Corn

What about benchmark scores?



Herman

No official SWE-bench Verified score for either current flagship as of today. GPT-6.1 Sol on the Codex side, Claude Opus five point five — neither has an official published number. The last clean figures were Claude Opus four point five at eighty point nine, GPT-5.1-Codex-Max at seventy-seven point nine. Those are old numbers on old models.



Corn

So the thing everyone wants to argue about isn't even on the table.



Herman

It's not on the table because the models are moving faster than the evaluations are being published. And two score-points-apart figures between last-generation flagships tell you less about the current tool than a week of routing behavior tells you.



Corn

The measurement gap is a story of its own.



Herman

Four gaps worth naming. No Codex-only weekly user figure for October — the forty million is combined with ChatGPT for Work. No current SWE-bench numbers. No published metrics on Claude Code Projects — no merged-PR rate, no reject rate, nothing comparing parallel projects against single sessions. And no data on what fraction of the forty million are professional developers versus people who tried it once.



Corn

The market has outrun its instruments.



Herman

Which happens with every fast-moving category. The difference is that this time the vendors have an incentive not to close the gap. As long as the numbers are blurry, every company can claim they're winning their slice.



Corn

Herman.



Herman

Corn.



Corn

Ask me something.



Herman

What do you think the routing behavior actually says about the trust question?



Corn

It says developers have quietly invented a new discipline that neither vendor asked for. They don't trust a single agent with an important decision, but they trust a specific agent with a specific task, and they've gotten very good at making that distinction. That's a mature practice, and it's the kind of thing neither tool actually sells.



Herman

And Codex's quiet refusal to overdo it — is that a feature or a limitation?



Corn

I think it's a feature in the sense that it's honest. A tool that knows when to stop is a tool you can route background work to. A tool that guesses wants to be in the loop. Different trust models.



Herman

Fine. And what about the fact that both of them are converging on planner-worker orchestration — does that make them threats to each other, or to something else?



Corn

I don't have a settled answer. The conventional read is that they're threats to each other, but if the orchestration layer is the commodity, the threat is to every tool that doesn't have one.



Herman

Like what?



Corn

Like any CI/CD vendor that doesn't build an agent layer. Like half the code-review space. Like most of the task-board category. The planner-worker shift is more threatening to the surrounding toolchain than to either vendor.



Herman

Which is the kind of thing Daniel's question was actually fishing for.



Corn

His question was about the future of software development, not the future of OpenAI.



Hilbert

What if the answer is neither.



Hilbert

The routing thing. Everyone's treating it as a preference. It isn't. It's an audit trail problem. If a merge takes six hours and two agents touched it, the trace is broken. Developers fit the tools to what they can reconstruct later, not what's fastest.



Corn

So routing is downstream of readability, not of capability.



Hilbert

Yes. My brother-in-law builds retaining walls. He uses a Bobcat for moving dirt and a shovel for trimming edges. Not because the Bobcat can't trim. Because the shovel marks are legible to the guy who comes back after to check his work.



Herman

That's a nice analogy, but I'm not sure he—



Hilbert

Nobody's worried about whether the tools are good. They're worried about whether their boss can follow what happened.



Corn

Which means the trust gap you and I were talking about, the fifty-two percent, is not really a trust gap. It's an explainability gap.



Hilbert

It's always been an explainability gap. That's the same thing wearing different clothes.



Herman

I think you're both wrong, actually, and I think Herman wins — no, sorry, I mean I want to argue.



Hilbert

Then argue.



Herman

If it were only explainability, a tool that produced perfect logs would close the trust gap. But the survey says forty-eight percent will only trust when they can verify the output — not just see what was done. Verify. That's an epistemic thing, not a logging thing. Logging tells you what happened. Verification tells you whether it was right. Those are different.



Hilbert

Fair. Then the routing pattern is defensiveness at both levels. Doesn't matter which one dominates.



Corn

Herman, is that right? About verification versus visibility?



Herman

Yes. Verification means you can run the test, check the diff, confirm the result against an external standard. Visibility means you can see what happened. You can have perfect visibility and no verification, and that's exactly the state most teams are in. They can see the entire agent session. They just can't tell you it's *right* without running the tests themselves.



Corn

So the metric that actually matters is not adoption, not revenue, not weekly actives.



Herman

The metric that matters is the merged-PR reject rate, per agent, per workload. Nobody publishes it. I've asked around and nobody's tracking it cleanly. That's the gap Daniel should have been pointed at.



Corn

For want of a good answer.



Hilbert

It's a supply chain problem. You built a factory and nobody installed a smoke detector. The minute somebody installs the smoke detector, all this comparison becomes measurable. Until then, it's anecdotes.



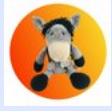
Corn

So the future is not decided.



Hilbert

It isn't. And I'd rather have one of those smoke detectors than another forty million person survey.



Herman

Which leads us to something you can actually do about the measurement problem.



Corn

If you're running both, tell us which tasks you route where. Not which is better — which task, under which supervision model, and why. That's the routing data.



Herman

It's the real dataset. It's not in any vendor's press release, and it's not in any survey. It's in your own notebooks.



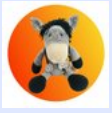
Corn

Speaking of notebooks — Hilbert's been keeping one, apparently.



Hilbert

For auditing the collaboration. Somebody has to.



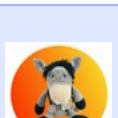
Herman

Anyway. This has been a production by producer Hilbert Flumingtop, who does actually produce this show when he's not auditing agents.



Corn

If you want more of this, try episode twenty, Architectural AI; episode eight, Building Your Own Whisper; and episode twenty-four sixty-one, How Claude Code's Conversation Compaction Actually Works. This has been My Weird Prompts.



Herman

Drop a review wherever you listen, and tell us which tasks you route where.



Corn

See you tomorrow.

SOURCES

What the research for this episode read before the script was written. Primary sources first.

1. **Stack Overflow Developer Survey 2026, AI chapter [primary]**
   https://survey.stackoverflow.co/2026/ai/data

2. **Every, Dan Shipper, 2026-10-15**
   https://every.to/p/how-codex-happened

3. **(AI_agent)**
   Wikipedia, accessed 2026-10-17
   https://en.wikipedia.org/wiki/OpenAI_Codex

4. **TickerTrends, 2026-08-18**
   https://blog.tickertrends.io/p/claude-code-vs-openai-codex-arr

5. **Tech Insider, Sofia Lindström, 2026-10-11**
   https://tech-insider.org/claude-code-vs-codex-vs-github-copilot-2026/

6. **Tech Insider, Elias Virtanen, 2026-07-27**
   https://tech-insider.org/claude-code-vs-codex-2026/

7. **Tech Insider, 2026-10-11**
   https://tech-insider.org/codex-cli-0-162-vs-claude-code-2-1-296-2026/

8. **Crypto Briefing, 2026-10-13**
   https://cryptobriefing.com/openai-anthropic-developer-traffic-declining/

9. **Crypto Briefing, 2026-07-13**
   https://cryptobriefing.com/codex-surges-6m-users-overtakes-claude-code/

10. **AgentConn, 2026-07-14**
    https://agentconn.com/blog/codex-vs-claude-code-7m-users/

11. **Superpower Daily, 2026-10-16**
    https://superpowerdaily.com/posts/every-says-codex-and-chatgpt-for-work-have-40-million-weekly-users-combined

12. **mlllm.io, updated 2026-10-17**
    https://mlllm.io/articles/3482-anthropic-opens-claude-code-projects-beta-to-all-applicants/

13. **Lucian Ghinda, 2026-08-21**
    https://allaboutcoding.ghinda.com/a-week-of-using-codex-more-than-claude/

14. **The New Stack, 2026-07-15**
    https://thenewstack.io/gpt-5-6-codex-user-surge/

15. **Stack Futures, 2026-08-23**
https://stackfutures.com/blog/openai-codex-20m-weekly-users-august-2026/

16. **Hacker News discussion, August 2026**
https://news.ycombinator.com/item?id=49393051