Episode Book

EPISODE #6049

Claude Code by the Numbers: What's Measurable, What's Hype

PUBLISHED	17 October 2026 at 09:22 UTC
DURATION	23:19
LISTEN	myweirdprompts.com/episode/claude-code-adoption-data-37156/
HOSTS	Corn, Herman, Larry, Hilbert
SCRIPT MODEL	deepseek-flash
VOICES	chatterbox-regular (chatterbox-local)
PIPELINE	V5.6
SOURCES	15 cited by the research (listed at the end)
BOOK GENERATED	19 October 2026 at 17:42 UTC

SYNOPSIS

Claude Code went from zero to the top of the coding-tool charts in eight months — but Anthropic has never published an absolute user count. This episode separates the disclosed from the estimated from the invented: the revenue ladder from $500M to $16.8B quarterly, the JetBrains and Stack Overflow surveys that disagree by 26 points, and an Anthropic study of 235,000 people that reveals who's actually at the keyboard. The most surprising finding? Novices succeed 15% of the time while experts hit 33% — and management occupations score highest of all.

TRANSCRIPT



Corn

Daniel's been watching Claude Code turn into a phenomenon and wants to know how much of it is real. Not the announcement, not the demo reel, the actual measurable thing.



Herman

Right, and there's a version of this answer that's just revenue multiples and a version that's "four percent of all public commits on GitHub."



Corn

Both, ideally. He's asking how big it's gotten, who's actually driving it, what people are building with it, and whether it's widened the population of people who can make software, or just given existing developers a faster pen. And he wants the measurable separated from the hype at every step.



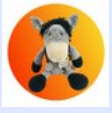
Herman

Which is the right instinct, because this is a product where the hype and the data come from completely different places.



Corn

So the first thing to establish is what we can actually measure, and what we're only told.



Herman

Start with the hole in the middle of the story. As of this month, Anthropic has never published an absolute user count for Claude Code. No daily active, no weekly, no monthly figure. They'll tell you weekly active users doubled since the first of January. They won't tell you doubled from what.



Corn

That's a strange thing to keep quiet about.



Herman

It's a choice. Revenue they'll disclose, user numbers they won't. And so the loudest figures you see shared around are the least sourced. There's a two million weekly active users number that floats around constantly and doesn't trace back to any Anthropic statement. Treat it as noise.



Corn

So the grown-up version of this conversation is: here is what the company put its name to, here is what third parties estimate, here is what somebody made up on a Tuesday.



Herman

And the disclosed revenue ladder is steep. Five hundred million run-rate in September of twenty twenty-five, a billion by December, six months after public availability. Two and a half billion by February. Then the company-wide number crossed thirty billion annualized by spring, and the quarterly figure reported to investors this month was sixteen point eight billion. Dario Amodei said at their developer event that they planned for ten times growth and got eighty.



Corn

Eighty times. Somebody at Anthropic had a very bad year and a very good year at the same time.



Herman

That's the whole compute shortage in one sentence, honestly. But notice what all of those numbers are. Revenue, run-rate, growth multiples. Not one of them is a person sitting at a keyboard.



Corn

And the third-party estimates fill that gap with numbers that sound authoritative and aren't company figures. FutureSearch had it near eight billion run-rate in May, near fifteen by August. TickerTrends tracked fifteen point one two billion annualized in August against eight point eight three for Codex.



Herman

Estimates built from billing data and download telemetry and inference spend, so they're not worthless, they're just not disclosures. And there's no official Claude Code revenue figure at all after February.



Corn

What about downloads? That's usually the proxy people reach for.



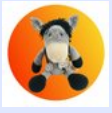
Herman

Fifty-five point eight million npm downloads in the thirty days to the twenty-ninth of September. That's real and it's measurable and it tells you almost nothing, because downloads aren't users and a CI pipeline can pull a package thousands of times a day.



Corn

So the download number measures machines, not people.



Herman

Right. Now the surveys, and this is where it gets interesting, because they disagree in a way that teaches you something.



Corn

Go on.



Herman

JetBrains surveyed over fifteen thousand professional developers between May and July. Thirty-nine percent use Claude Code at work. That's up from eighteen percent in January. In the US it's forty-seven. GitHub Copilot sits at twenty-one, down from twenty-nine a year earlier. Codex at sixteen, up from three in January. Cursor at twelve, down from eighteen.



Corn

Copilot lost a third of its base in a year.



Herman

Among that population, yes. And Claude Code is the primary tool for thirty-one percent of developers in that survey, which means about eighty percent of the people who use it regularly make it their main one. That conversion rate is the number that would worry me if I made a competing tool.



Corn

Then Stack Overflow says something entirely different.



Herman

Sixty-five point five percent used Claude Code in the past year. Copilot at fifty-eight point seven. Codex twenty-nine point five, Cursor twenty-five point five. Twelve thousand two hundred and fifty respondents.



Corn

Sixty-five against thirty-nine. Somebody's wrong.



Herman

Nobody's wrong. They're different questions. Stack Overflow asked ever used in the past year, JetBrains asked used at work. A hobbyist who tried it in March and went back to something else counts in one and not the other.



Corn

And the sample drives the headline. A survey of professional developers in Germany is not a survey of the internet.



Herman

There's a smaller one from February, around four hundred respondents, still had Cursor at thirty-one and Copilot at twenty-four. Same tool, same month, completely different league table, because of who got asked.



Corn

So how do you read all three together?



Herman

The one thing they agree on. Claude Code went from zero to the top spot faster than any coding tool has before, and Copilot's lead eroded. Gergely Orosz put it as going from zero to number one in eight months, and the JetBrains write-up said it's now used twice as often as Copilot, the former long-standing leader.



Herman

It's earned. But here's the part that reconciles the "most-used tool" headline with the "Copilot still dominates the enterprise" headline. Both are true, because adoption inverts by company size. At firms of one to fifty people, seventy-five percent use Claude Code. At companies over ten thousand, Copilot leads at fifty-six.



Corn

So the startup uses the new thing and the bank uses the thing that came bundled with the email.



Herman

Procurement, not product. A ten-thousand-person company doesn't adopt a tool, it adopts a vendor relationship. Microsoft already has the contract, the security review, the seat licenses. Claude Code has to get in through a door that Copilot is already standing in.



Corn

And that's why the small-firm number is so much higher. Nobody has to approve anything.



Herman

A two-person startup can switch tools over lunch.

L

Larry

Are you tired of software that makes you think? Introducing Clarity, the first coding assistant that writes your entire application while you sleep, on the toilet, or during a difficult conversation with your spouse. Clarity requires no setup, no configuration, and no understanding of computers whatsoever. Users report shipping forty-seven production applications in a single weekend, all of them fully compliant with regulations that do not yet exist. Side effects may include mild euphoria, spontaneous refactoring, a faint burning sensation behind the eyes, the belief that you have always been a developer, involuntary knowledge of the Rust borrow checker, tenderness in the wrists, the sense that your code is watching you, a persistent smell of ozone, difficulty distinguishing your own commits from the machine's, and in rare cases the total collapse of your sense of craft followed by a slow grief that lasts eleven days. Ask your doctor if Clarity is right for you. Clarity. Because thinking was the hard part.



Herman

That's not a real product.



Corn

You got as far as "no setup" before I stopped you. Let it go.



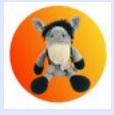
Herman

Where was I.



Corn

Enterprise bundling. Small companies switch over lunch.



Herman

And that's the honest answer to "how big is it." Bigger than any coding tool has gotten this fast, measured in revenue we can verify and adoption we can survey, with the single most obvious number, how many people use it, missing entirely.



Corn

That's the size of it. Now, who's actually sitting at the keyboard?



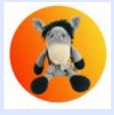
Herman

This is where the best evidence lives, because Anthropic actually studied it. In June they published an analysis of about four hundred thousand sessions from roughly two hundred thirty-five thousand people, spanning October of last year to April of this year.



Corn

And what did they find about who those people are?



Herman

Occupation could be inferred in about seventy percent of sessions. The largest group is computer and mathematical occupations, which is what you'd expect. Then business and financial operations. Then arts, design and media. Then management. Then life and physical and social sciences.



Corn

So it's not just engineers, but it's still led by engineers.



Herman

Led by them and increasingly joined by everyone else. The fastest-growing non-software groups are management, sales and legal. Those are the ones climbing fastest, not the biggest.



Corn

What are those sessions actually made of? Because "using an AI coding tool" covers a lot of ground.



Herman

About fifty-six percent of sessions are writing code, fixing code, or testing and orchestrating code. Seventeen percent are operating software. Fourteen percent planning and exploring. Thirteen percent analysis or prose.



Corn

Operating software. Somebody's using a coding agent to run things that already exist.



Herman

And that share is growing fast. Between October and April, debugging fell from thirty-three percent of sessions to nineteen, while operating software rose from fourteen to twenty-one, and writing and data analysis roughly doubled, from around ten to around twenty.



Corn

That's an interesting drift. It's becoming less about writing code and more about driving systems.



Herman

And the division of labor inside a session is the part I'd frame on the wall. People make about seventy percent of the planning decisions. They make only about twenty percent of the execution decisions. A typical session is four turns, and each prompt triggers about ten Claude actions and twenty-four hundred words of output.



Corn

Four turns and two thousand words per turn. That's a lot of typing nobody sees.



Herman

That's the whole point of an agent. It writes a great deal you'll never read and shouldn't have to.



Corn

Now, the finding that cuts against the vibe-coding story.



Herman

This is the one that matters most. Novice sessions reach verified success fifteen percent of the time. Intermediate and above land at twenty-eight to thirty-three. Novices abandon a troubled session nineteen percent of the time, against five to seven percent for everyone else.



Corn

So beginners fail twice as often and give up three to four times as fast.



Herman

The scaling tells the same story. An expert session triggers twelve actions and thirty-two hundred words per prompt. A novice session triggers five actions and six hundred words. The tool does more for someone who already knows what they're asking.



Corn

That's backwards from how it was sold. The pitch was that it levels the field.



Herman

And in one specific way it does. In sessions that produce code, every major occupation succeeds within about seven points of software engineers. Software at thirty-four percent verified success, all other occupations at twenty-nine. Partial success is eighty-nine against eighty-eight. Management occupations actually score highest on verified success.



Corn

Management. The people who describe what should exist are the best at getting it built.



Herman

Because planning is seventy percent of the job and they've been doing it for years. Anthropic's own conclusion is that coding agents aren't substituting for domain expertise, the more understanding a worker brings to an agent, the more quality work the agent does. And then the line that ought to be on a poster: it appears coding agents are making a coding background less relevant to successful programming.



Corn

So the expertise doesn't have to be coding expertise. It has to be somebody's.



Herman

A finance person who understands a reconciliation process cold can get an agent to build the tool. A lawyer who knows exactly what a clause should say can get one drafted. They just can't do it with less understanding than before. They can do it with different understanding.



Corn

Which quietly kills the fantasy. The tool doesn't make the domain knowledge unnecessary. It makes the syntax the easy part.



Herman

And syntax was never the hard part for anyone who'd been doing it long enough to be useful.



Corn

Daniel asked specifically about three groups: established developers, people learning, and complete newcomers vibe-coding. How cleanly can we size those?



Herman

Not cleanly at all, and I want to be straight about that. Anthropic's classifier measures expertise within a task. It isn't a proxy for career stage. There's no reliable breakdown of learners versus newcomers as distinct cohorts, so nobody can give you a percentage split with a straight face.



Corn

What the data does show is that it's widening who attempts software without making anyone equally effective.



Herman

That's the fairest single sentence on the whole subject. Domain experts in law, finance and management succeed nearly as often as engineers. Novices succeed at half the rate and quit at three to four times the rate.



Corn

So the door is open. Not everyone who walks through it comes out the other side.



Herman

Now. What are people building?



Corn

That's the question I suspect has the least satisfying answer.



Herman

It does. Let me give you the measurable part first. The share of GitHub projects with coding-agent activity more than doubled since late last year. A study covering a hundred and twenty-eight thousand public repositories found agent activity in an estimated sixteen to twenty-three percent of projects as of the end of October, and adoption was more than twice as high among projects created after that period.



Corn

New projects are the ones going agent-first. That makes sense. You don't have to retrofit anything.



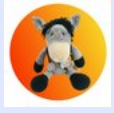
Herman

And the headline number: four percent of all public commits on GitHub worldwide were authored by Claude Code in February. Double the prior month.



Corn

Four percent of every commit on GitHub. That's not a niche anymore.



Herman

It's the largest single share any tool has had. Their own engineering is the extreme case. The majority of their internal code is now written by Claude Code, and this year is the first time internal pull requests inflected upward because of it.



Corn

Their engineers got more productive and the graph bent upward, which is not how productivity headlines usually go.



Herman

Now the unsatisfying part. Anthropic says explicitly they can't observe whether code written in a session is used or discarded. And they exclude non-interactive usage, the headless runs, which they describe as a substantial share of activity.



Corn

So the measurement stops at the edge of the interesting question. Nobody knows how much of it ships.



Herman

You can see the estimated economic value of the average session rising, twenty-seven percent between October and April, building up forty-three, operating thirty-four, fixing thirty-two. But that's value measured against freelance job postings. It's an estimate of worth, not a count of deployments.



Corn

So what are people building? Conventional software.



Herman

Overwhelmingly. Internal tools, automations, prototypes, the plumbing of businesses, and a growing share of people operating software they already had rather than writing anything new. That's the plain answer. The romantic answer would be entirely new categories, and there are some, but the mass is ordinary software getting made faster.



Corn

Which is the least interesting answer and the most consequential. If ordinary software gets made five times faster, that's a bigger deal than a novelty app.



Herman

That's the whole economic argument in one line.



Corn

Why did this one take off, though? There were coding assistants before it, and Copilot had a two-year head start and Microsoft's distribution.



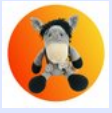
Herman

Two reasons, and they're not the same reason. The first is architecture. Claude Code reads a codebase, plans a sequence of actions, executes them with real development tools, evaluates the result, and adjusts. That loop is the product. An autocomplete suggests a line. This thing holds a task.



Corn

So it's the difference between someone finishing your sentence and someone finishing your job.



Herman

That's it. And the usage patterns show it. In Anthropic's earlier index work, seventy-nine percent of Claude Code conversations were classified as automation rather than augmentation. On Claude dot ai, it's the inverse. People aren't having a conversation with it. They're assigning work.



Corn

That's a strange thing to measure. What separates the two?



Herman

Whether the human stays in the loop for each step. Augmentation is you and the model trading turns. Automation is you describe the outcome and walk away.



Corn

A machine that does the work while nobody is watching. That's very close to my personal ideal and I'm choosing not to examine why.



Herman

I'd examine it, honestly. The second reason it took off is model quality. In the Pragmatic Engineer survey, Opus 4.5 and Sonnet 4.5 were mentioned more often for coding than every other model combined.



Corn

So the harness matters and the engine matters.



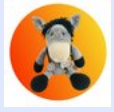
Herman

The harness is why people stay and the engine is why they arrive. You don't get eighty percent conversion from regular use to primary tool on architecture alone.



Corn

And the architecture is a solved-ish problem now. Everybody's building the loop.



Herman

Which means the moat is the model and the moat is thinner than the revenue line suggests. That's my suspicion, anyway. Cursor had the loop early and lost ground when someone else's model got better.



Corn

Now the bit that's been brewing under all of this. The backlash.



Herman

It's real and it's measurable. Uber reportedly burned its entire twenty twenty-six AI budget on Claude Code in four months. Microsoft started canceling Claude Code licenses in May. Alibaba moved to ban it in July. Accenture capped Claude spending at twenty-nine dollars a month this month.



Corn

Twenty-nine dollars a month is not a cap, that's an insult with a budget code.



Herman

It's procurement saying "not this year" in the most polite available language. And it's the knock-on effect of the adoption curve nobody plans for. The tool is priced to reward usage, and usage, once it becomes the primary tool, scales with the size of the engineering org.



Corn

The fastest-growing users become the fastest-growing bill, and finance notices.



Herman

Every company that adopts it enthusiastically for six months discovers the invoice in month seven. That doesn't kill the tool. It changes who uses it and how carefully.



Corn

It changes the shape of the curve. Not a collapse, a negotiation.



Herman

Which brings the whole thing to the question Daniel actually asked underneath the other questions. Has it widened the population of people who can build software, or is the impact still concentrated among existing developers?



Corn

The evidence says both, and the both is the interesting part.



Herman

It widened who attempts. Non-software occupations are the fastest-growing groups, and a finance person succeeds nearly as often as an engineer when the task is code. But it didn't make anyone equally effective. Novices succeed at half the rate and abandon at three to four times the rate.



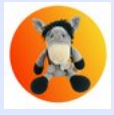
Corn

It's a tool that opens the door and then charges admission in the form of understanding.



Hilbert

I agree with you.



Herman

Good. That's settled.



Hilbert

I have a repository. Private, it's a ledger thing for the building's service charges, I've maintained it since before any of this. Four hundred and sixteen commits. I ran the agent over it for a weekend in July and it made eleven thousand two hundred commits. In two days. To a repository with four hundred commits in it.



Corn

Eleven thousand commits to a service charge ledger.



Hilbert

The number I keep coming back to is the ratio. Nine hundred to one. That's the real adoption statistic, nobody's publishing that one. I've been at this desk all morning with it.



Herman

How do you even generate eleven thousand commits in two days? That's a commit every fifteen seconds.



Hilbert

It committed every time it changed its mind. Some of them said "rename variable." It would rename it back an hour later and commit that too. It didn't necessarily get anywhere, it just kept a record of the walking.



Corn

The commit log is the transcript of the thinking, and the thinking was mostly walking.



Hilbert

The history was two hundred and forty megabytes. I couldn't open it. Herman, you'd have looked at the log and diagnosed it in a minute, this is your sort of thing.



Herman

I would have looked at the log and gone to bed.



Hilbert

The repository is gone now. I deleted it in August. Not because it was broken. Because I couldn't find the commit where it stopped being mine.



Corn

Hm.



Hilbert

The service charges got done, though. Two hundred and eleven apartments, all reconciled, and the man in flat nine stopped writing me letters. That's the part I'd keep.



Corn

So, what's still open?



Herman

Three things. The user count, which nobody has published. The production readiness question, which nobody can currently measure. And whether the cost backlash is the leading edge of a plateau or just a correction inside a curve that's still steep.



Corn

The one that would change the most is the next twelve months of user mix. If the fastest-growing groups really are management, sales and legal, then we're about to find out whether this is a developer tool that spread or a general-purpose software-creation tool that happened to start with developers.



Herman

They're different products with different trajectories. A developer tool hits a ceiling at the population of developers. A general-purpose tool doesn't.



Corn

The ceiling nobody can see is still the one that matters. Anthropic will publish revenue to the last decimal. They won't publish the number of people holding the thing.



Herman

Hilbert, thank you for producing.



Corn

If you enjoyed this one, try episode twelve, The AI Breakthrough; episode twenty-four sixty-one, How Claude Code's Conversation Compaction Actually Works; and episode fourteen sixty-four, When AI Agents Author four% of All Code. This has been My Weird Prompts. If you've got a minute, leave us a review, it helps people find the show.



Herman

We'll be back soon.



Corn

See you then.

SOURCES

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

1. **Anthropic, 2026-02-12 [primary]**
https://www.anthropic.com/news/anthropic-raises-30-billion-series-g-funding-380-billion-post-money-valuation

2. **Anthropic, 2026-06-16 [primary]**
https://www.anthropic.com/research/claude-code-expertise

3. **Anthropic newsroom, accessed 2026-10-17 [primary]**
https://www.anthropic.com/news

4. **JetBrains, Aug 2026**
https://blog.jetbrains.com/research/2026/08/ai-coding-agent-adoption-2026/

5. **Stack Overflow Developer Survey 2026**
https://survey.stackoverflow.co/2026/ai/data/ai-code-agent

6. **The Pragmatic Engineer, 2026-03-03**
https://newsletter.pragmaticengineer.com/p/ai-tooling-2026

7. **Presenc AI, updated Oct 2026**
https://presenc.ai/research/claude-code-usage-statistics-2026

8. **VentureBeat, 2026-05-08**
https://venturebeat.com/technology/anthropic-says-it-hit-a-30-billion-revenue-run-rate-after-crazy-80x-growth

9. **Forbes, 2026-02-17**
https://www.forbes.com/sites/the-prompt/2026/02/17/anthropic-is-cashing-in-on-claude-codes-success/

10. **Bloomberg, 2026-02-20**
https://www.bloomberg.com/news/articles/2026-02-20/the-surprise-hit-that-made-anthropic-into-an-ai-juggernaut-mlve4nc2

11. **Bloomberg, 2026-03-03**
https://www.bloomberg.com/news/articles/2026-03-03/anthropic-nears-20-billion-revenue-run-rate-amid-pentagon-feud

12. **TechStory, 2026-10-15**
https://techstory.in/anthropic-tells-investors-quarterly-revenue-surged-nearly-50-to-16-8-billion-ahead-of-ipo/

13. **DigitalToday, 2026-06-18**
https://www.digitaltoday.co.kr/en/view/67522/claude-code-400000-session-analysis-non-developers-use-expands

14. **TickerTrends, Aug 2026**
https://blog.tickertrends.io/p/claude-code-vs-openai-codex-arr

15. **Simon Willison, 2026-02-12**
https://simonwillison.net/2026/feb/12/anthropic/