Episode Book

EPISODE #5859

Slow Labs, Fast Ships: What Release Cadence Actually Tells Us

PUBLISHED	11 October 2026 at 10:50 UTC
DURATION	28:34
LISTEN	myweirdprompts.com/episode/ai-release-cadence-analysis-40162/
HOSTS	Corn, Herman, Hilbert
SCRIPT MODEL	deepseek-flash
VOICES	chatterbox-regular (chatterbox-local)
PIPELINE	V5.3
SOURCES	13 cited by the research (listed at the end)
BOOK GENERATED	19 October 2026 at 17:40 UTC

SYNOPSIS

Everyone assumes the careful labs release slowly and the hype-driven ones ship constantly. The data says the opposite. Drawing on Suprmind's AI Models Index and ARK Investment Management's frontier-gap analysis, this episode digs into real release medians across Anthropic, OpenAI, Google, DeepSeek, Mistral, xAI, and Meta — and finds a lab most associated with pacing is actually the fastest shipper in the industry. Then it zooms out to the decades-old rolling-release debate in software: is a slower cadence a mark of engineering maturity, or just a romantic reading of slower development? The answer turns out to be messier — and more interesting — than either side expects.

TRANSCRIPT



Corn

Daniel has a confession to make, and it's the kind of confession that makes him sound like a man who reads release notes for fun.



Herman

Which he does.



Corn

He says he's a fan of AI labs that release new models relatively infrequently. He'd rather have a slower, more predictable cycle where each new model is a substantial improvement than a constant drip of incremental updates. His reasons are decent ones: a slower cadence feels less hype-driven and more like rigorous internal validation, it saves him a mountain of downstream work, because every new model means re-evaluating, re-integrating, and deciding whether to migrate, and in an industry moving at an absurd pace there's something appealing about a stable technological foundation.



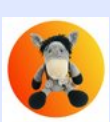
Herman

And he wants two things out of it.



Corn

Two things. First, which labs actually fit which philosophy. Are there companies deliberately favoring fewer, more substantial releases, and others chasing rapid iteration, and can we tell the difference from their release histories rather than their marketing? And is there any actual evidence that a slower cadence correlates with better testing, reliability, or quality? Second, he wants us to zoom out, because this tension predates generative AI by decades. Debian Stable, Ubuntu LTS, Arch, the whole rolling-release argument. He wants to know whether releasing less frequently is a mark of engineering maturity, or a romantic reading of what might just be slower development.



Herman

It's a good instinct, and I think the data is going to complicate it in a way he won't expect.



Corn

Then let's complicate it.



Herman

So here's the thing. Until fairly recently, the "slow lab versus fast lab" debate was vibes. You'd read a blog post, you'd look at a release chart, you'd form an impression. That's changed. There are now two independent datasets tracking frontier release cadence with actual medians and actual dates, and they broadly agree on the shape of the thing.



Corn

Name them.



Herman

Suprmind's AI Models Index, which updates every two weeks, and ARK Investment Management's frontier-gap analysis, which pulls from Artificial Analysis. Suprmind's latest, from the fourth of October, says the industry now ships a premium model every four and a half days. In 2023 it was every seven and a half.



Corn

Four and a half days. That's not a release cycle, that's a heartbeat.



Herman

And the second part of the episode, the history, is where it gets interesting, because the software industry fought this exact fight in the late nineties and settled it in a way that cuts against Daniel's intuition. So we've got two threads: what the cadence data actually shows about AI labs, and what fifty years of release engineering can teach us about whether slow is mature or just slow.



Corn

And the finding you're building toward.



Herman

The lab most associated with pacing is the fastest shipper in the industry.



Corn

That's either a paradox or a marketing department.



Herman

Let's do the data first.



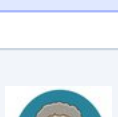
Corn

Let's do the data.



Herman

Suprmind breaks it down lab by lab, and the headline is Anthropic. Median gap between premium releases: one hundred and thirty-three days in 2023, down to twenty-two and a half days in 2026, across ten releases. That's the fastest-shipping lab in the industry, by a wide margin.



Corn

That is not the lab I would have guessed. That is the exact opposite of the lab I would have guessed.



Herman

OpenAI is the steadiest. Fifty-two to fifty-seven days every year since 2025. Almost metronomic. Google and DeepSeek actually slowed down. Google's median gap grew from forty-three and a half days in 2025 to seventy-nine and a half in 2026. DeepSeek went from sixty-three to a hundred and twenty-seven and a half. And Mistral is the slowest of the majors at a hundred and forty-seven days.



Corn

So the two labs with the reputation for restraint are the ones moving fastest, and the ones with the reputation for shipping constantly have actually slowed.



Herman

Which is why ARK's numbers are worth putting next to Suprmind's, because they measure slightly different things. ARK looked at five US labs, and the median frontier gap fell from thirty-seven and a half days in 2023 to eleven days in 2026 so far. OpenAI fastest at forty-nine days in their cut, Anthropic at seventy-one and a half, Google at ninety-three, xAI around a hundred and thirty-five and a half. And Meta is the outlier at three hundred and sixty-eight days.



Corn

Three hundred and sixty-eight days isn't a cadence. That's a geological era.



Herman

It's worse than it sounds. Shipped zero frontier models in the fourteen months after Llama 4, which came out in April 2025. That's not a slow cadence, that's a gap in the record.



Corn

So when Daniel asks whether there are labs that deliberately favor fewer, more substantial releases, the honest answer is almost no. There is no standalone slow-release lab.



Herman

Every major lab has compressed since 2023. The two closest things to deliberate slowness are, and 's slowness is attributed to a team overhaul, not a philosophy, and Anthropic's held ceiling, which is a different animal entirely. Anthropic has a model it calls Model 2, described in its August risk report as more capable than its current flagship, and it has no plans to release it.



Corn

So Anthropic is holding a model back while shipping faster than anyone else?



Herman

That's the shape of it. And it makes sense once you separate two things people constantly conflate. A release-gating policy is not a release cadence. Anthropic gates which models get out the door. It doesn't slow down the ones that do.



Corn

Say more about the pacing thing, because that's the argument Daniel is implicitly buying.



Herman

Dario Amodei published an essay on the twelfth of September titled "We Must Pace the Frontier." Within a day, Sam Altman and Elon Musk had both endorsed it, which is not a sentence I expected to say. And the essay commits Anthropic to embedded outside evaluators. It does not commit Anthropic to a slower release schedule.



Corn

So the essay is about oversight, not tempo.



Herman

Correct. And then on the twenty-second of September, Anthropic shipped Opus 5.5 at four dollars and twenty dollars per million tokens, down from Opus 5's five and twenty-five. And on Terminal-Bench 4.0, the cheaper Opus 5.5 scored sixty-six point four percent, while



Herman (continued)

the flagship Fable 5.1 scored fifty-five point eight. The cheaper, newer, faster-shipped model beat the more expensive one.



Corn

That's a strange kind of slowing down.



Herman

Isaac Vazquez made the sharp version of this argument on the twenty-seventh of September. He said pacing, as Anthropic practices it, holds the top tier's name and price in place while the capability underneath gets cheaper. Which is a pricing strategy wearing a safety stance.



Corn

That's the line of the episode.



Herman

It's the line of someone else's essay, but I'll take it. And it's worth noting the skeptics. TechCrunch ran a follow-up on the seventeenth of September gathering critics who called the pacing essay regulatory capture. Aidan Gomez at Cohere framed the whole debate as a fight over who writes the guardrails.



Corn

Which is a different argument than whether the pacing is real. The pacing is real at the top of the stack. It's just not what people think it is.



Herman

Right, so let's get to the actual question Daniel asked, because this is where it gets uncomfortable. Is there evidence that slower cadence correlates with better testing, reliability, or model quality?



Corn

And the answer is no.



Herman

The answer is no, and it's worth being precise about why. The available data measures cadence and blind-vote preference. Nobody publishes how much internal validation a model got before release. There is no line item for "hours of red-teaming" next to "median days since last release." So the thing Daniel is asking about is unmeasurable from the outside.



Corn

Which doesn't mean the intuition is wrong. It means it's unverified.



Herman

But here's where it gets interesting, because Suprmind found something that cuts against the simple story. Fast follow-ups disappoint. The median release beat its predecessor by twenty-one and a half LMArena points in 2025, but only nine point three in 2026. Among Western labs, the 2026 median gain is two and a half points. And ten of twenty-nine measured releases went backwards.



Corn

Backwards how? Losing to the model they replaced?



Herman

The 2026 regression rate is sixteen percent, up from eleven in 2025. About one in six releases clearly lost to its own predecessor.



Corn

One in six. That's not noise, that's a pattern.



Herman

And the clearest example is GPT-5.2. Shipped twenty-nine days after GPT-5.1. It wins forty-seven point three percent of blind votes against it. That's a statistically real loss.



Corn

So OpenAI shipped a model that is, on blind preference, slightly worse than the model it replaced, because the gap was twenty-nine days.



Herman

And the gap is the whole story. Point releases shipped within forty-five days gained a median of point one LMArena points. Point releases after longer gaps gained fourteen point nine. Suprmind's own summary is that the time since the last release explains more than the version label.



Corn

The wait explains the result more than the version number does.



Herman

Yes. Which is a good finding, because it means there's a real mechanism here. If you wait, you have more to say. If you don't wait, you ship something that's mostly a re-labelling.



Corn

And the counterweight?



Herman

The counterweight is that capability still rises. On the Artificial Analysis Intelligence Index, none of thirty-nine version pairs went backwards. Point releases gained a median of four index points, new generations five and a half. Suprmind is careful to note that LMArena compresses at the top, so part of the smaller step is a measurement artifact rather than a real slowdown.



Corn

So you have two indexes saying different things. One says releases are getting smaller and sometimes regressing. The other says nothing is regressing at all.



Herman

Which is a good reminder that the benchmark you pick shapes the story you tell. LMArena measures pairwise preference, which compresses once models get good. Artificial Analysis measures capability on a fixed scale, which doesn't.



Corn

And you don't trust either one alone.



Herman

I trust the direction both are pointing at, which is that the marginal release is worth less than it used to be. What I don't trust is any claim that a specific lab is doing rigorous testing because its cadence is slow. That claim isn't supported by anything I've seen.



Corn

So Daniel's intuition has real support, just not the support he thinks.



Herman

That's the right way to put it. Faster releases are producing smaller steps and more regressions. That's the empirical case for his preference. But the mechanism isn't rigor versus speed. It's that the wait is doing work, and the labs that wait longer get more out of each release, whether or not they're doing anything more disciplined behind the scenes.



Corn

What about Ethan Mollick's observation? I remember him saying something about cadence.



Herman

June nineteenth. He said, if AI self-improvement, even in a limited way, is possible, the cadence of shipping both AI products, harnesses, and models should go up. And he noted it appears to be happening at Anthropic and OpenAI, but not for any other labs. Which is an interesting frame, because it treats cadence as a signal about what's happening inside the lab, not about how careful the lab is being.



Corn

Cadence as a symptom rather than a virtue.



Herman

If your models are helping you build your next model, your cadence goes up. That's not a marketing decision, it's a physics of capability.



Corn

So the fastest lab might be the one with the best internal tooling, not the one cutting corners.



Herman

That's a much less comfortable conclusion than either side of the debate wants.



Corn

Alright. Let's zoom out, because I think the AI industry is about to rediscover a fight that operating systems have been having since the nineties.



Herman

Go.



Corn

Debian is the canonical slow-release project. It releases when it's ready, roughly every two years. You get about three years of security support, extended to about five with Long Term Support. Debian 13, trixie, shipped in August of last year. That's the model: frozen, tested, stable, and if you need something newer, you wait or you don't run Debian.



Herman

Ubuntu sits in between, and the two-tier structure is the interesting part. LTS releases come every two years, in April of even years, with five years of standard support, ten with Ubuntu Pro. The interim releases, the ones between the LTS, get nine months of support. So Ubuntu doesn't ask you to choose between fast and slow. It offers both, and prices them differently.



Corn

RHEL is the enterprise version of the same idea. A major version roughly every three years, ten years of maintenance. Red Hat will happily sell you stability until the hardware rusts.



Herman

Fedora is the fast lane in that family. Six months between releases, about thirteen months of support, which means Fedora users are effectively on a treadmill.



Corn

And then there's the other end. Arch and openSUSE Tumbleweed are rolling releases. There are no versions to expire because there are no versions. You update, you move on, you keep up.



Herman

And the Linux kernel itself, which is the interesting one, because the kernel picked a cadence and stuck to it. Every nine to ten weeks, and the last mainline release each year becomes the LTS branch.



Corn

So the kernel, which is the most foundational piece of software in the world, doesn't do Debian-style "when it's ready." It does a metronome. And it works.



Herman

It works because the cadence itself is the discipline. The kernel's release schedule is fixed, and what goes in is whatever's ready when the window closes. It's a completely different theory of release management than Debian's.



Corn

So the history Daniel's asking about already contains at least three distinct models. Slow when ready, tiered, and metronomic.



Herman

And a fourth, the rolling model, which effectively eliminates the concept of a release.



Corn

Now the question is how the industry got from the traditional model to continuous delivery, because the traditional model was bad.



Herman

Traditional releases accumulated months of changes. Which meant deployment was a dangerous event precisely because so much unverified work moved at once. You'd freeze a branch, spend weeks stabilizing it, ship it, and then hold your breath.



Corn

The whole industry ran on a quarterly panic.



Herman

And continuous integration came out of Extreme Programming, Kent Beck and Ron Jeffries in the late nineties. Ron Jeffries had a line about it. "Integration is a bear. We can't put it off forever. Let's do it all the time instead."



Corn

That's a sentence that changed the industry.



Herman

It really is. CruiseControl, the first continuous integration server, appeared in 2001. The ten-minute build practice came out of the same ethos, because if your build takes an hour nobody runs it, and if nobody runs it, you don't have integration, you have a wish.



Corn

And then Humble and Farley's Continuous Delivery in 2010 formalized the whole thing. The argument was that software should remain releasable through repeatable automation, so that shipping is a decision rather than an ordeal.



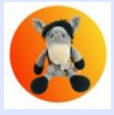
Herman

And lean thinking shaped the pipeline. Toyota's just-in-time, poka-yoke, error-proofing. The idea that you build quality into the process rather than inspecting it at the end.



Corn

Which is exactly the opposite of the Debian model.



Herman

It is, and that's why the next piece of research is important, because it directly attacks Daniel's intuition. There's a well-known piece from the Communications of the ACM that says, quote, it is often assumed that deploying software more frequently means accepting lower levels of stability and reliability. Peer-reviewed research shows that this is not the case. High-performing teams consistently deliver services faster and more reliably than



Herman (continued)

their low-performing competition.



Corn

So the fast teams are also the reliable teams.



Herman

And there's a lovely supporting detail. Amazon looked at its own check-in-to-production time and found it averaged sixteen days, about fourteen of which were spent waiting. Not working. Waiting. So the actual engineering time was a fraction of the calendar time.



Corn

Which is the failure mode of the slow-release model. Most of the "stability" is waiting.



Herman

Most of it.



Corn

So now we have a real contradiction. Debian's "when it's ready" is engineering maturity, but the continuous delivery research says frequent deployment correlates with higher reliability. Those can't both be right unless there's a distinction underneath them.



Herman

And there is, I think. The research is about deployment frequency. Debian is about release cadence. Different layers.



Corn

One is how often you push code into production. The other is how often you ship a frozen version with a number on it.



Herman

Right. And the continuous delivery argument isn't actually "ship constantly." It's "keep the batch small." Small batches are safer because the blast radius is small and rollback is cheap. If you can revert a deployment in five minutes, deploying twice a day is less risky than deploying once a quarter.



Corn

So the real variable isn't frequency, it's reversibility.



Herman

Which is the thing the AI labs don't have.



Corn

Say that again, because I think that's the whole episode.



Herman

If you run a shop on continuous delivery, the reason deploying twice a day is safe is that you can deploy a rollback in five minutes. The failure surface is contained and the recovery is cheap. AI models don't work that way. You can't roll back a model that's already in people's workflows. You can't revert the fact that a million developers integrated against the new API yesterday. The blast radius is not contained, and the rollback is not cheap.



Corn

So the AI labs are running the fast-cadence playbook without the safety mechanism that made fast cadence safe everywhere else.



Herman

That's the honest version of the tension. And it doesn't resolve into "slow good, fast bad." It resolves into "fast is safe when you can undo it, and it's dangerous when you can't."



Corn

Which leads us back to Daniel's question. Is slow a mark of maturity, or a romantic reading of something that might just be slower development?



Herman

And the answer depends entirely on why the lab is slow. Is the clean test case. Median gap of three hundred and sixty-eight days, zero frontier releases in fourteen months. That's not the slowest lab exercising discipline. That's an organization that lost its footing. I don't think anyone watching 's release history would call it mature.



Corn

So slowness can be many things. Discipline is one. Dysfunction is another. And a third is just not having finished the work.



Herman

And Anthropic is the fourth case. Fast cadence, but a release-gating policy that holds a more capable model back. So the slowness is at the top of the stack, not in the schedule.



Corn

Which is exactly what Vazquez was pointing at. The pacing is real. It's just not about tempo.



Herman

There's one more piece worth mentioning, because it's the internal version of the same argument. Business Insider reported that Anthropic's internal strategy was framed as "slow is bold," and that Amodei pushed to increase the company's quote risk budget, arguing that the marginal utility of extreme caution diminishes if the company ceases to be a frontrunner.



Corn

Even the lab that markets pacing decided that being too careful was the risk.



Herman

The risk budget framing is the giveaway. It treats caution as a resource with diminishing returns, which is the opposite of treating it as a virtue. If caution is a virtue, you can never have too much. If caution is a budget, you have to spend it where it counts.



Corn

Amodei is the one who said it, which makes the pacing essay read differently.



Herman

It makes the pacing essay read as a position rather than a conversion. He's not working against his own strategy. He's arguing for a specific allocation of caution.



Corn

The philosophical divide Daniel is looking for does exist, but it's not the one on the marketing page.



Herman

There's a divide between labs that gate releases and labs that don't. There's a divide between labs that can ship fast because their internal tooling is good and labs that can't. And there's a divide between labs that have the discipline to make fast cadence safe and labs that are just running the fast-cadence playbook because it looks like progress.



Corn

The AI industry is currently running that experiment without the rollback and blast-radius controls that made the experiment safe in traditional software.



Herman

Which is the actual finding for the episode. Not "slow is better." Not "fast is better." Cadence is a design choice with tradeoffs, and the AI industry has taken the fast path while skipping the safety rails that justified it everywhere else.



Corn

The question then isn't whether slowness signals maturity. It's whether speed can be made safe.



Herman

There's not much evidence yet that anyone has solved it.



Corn

Herman.



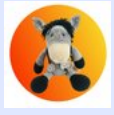
Herman

Corn.



Corn

You've been staring at the same chart for the entire conversation.



Herman

I have been. It's the Suprmind cadence chart. And I want to say something about it that I think Daniel might find uncomfortable.



Corn

Go on.



Herman

The chart is a picture of an industry that is shipping more and getting less, and everyone involved knows it, and nobody has stopped. The regression rate went from eleven to sixteen percent in a year. The per-release gain halved. And the industry's response has been to accelerate further, because the alternative is to fall behind on the leaderboard.



Corn

It's a prisoner's dilemma dressed as a product strategy.



Herman

It's exactly that. And the labs that are most exposed to it are the ones with the biggest lead, because they have the most to lose by slowing down.



Corn

The leader has the least ability to stop.



Herman

That's the trap.



Corn

Which is a good place to pause, because I think somebody behind the glass has something to say about release schedules.



Hilbert

You've got the argument right and you're missing the reason.



Hilbert

I made industrial control software for commercial laundry equipment. The big machines, the ones in hotel basements. You don't see them unless you're the guy fixing them, and for a few years, I was the guy looking at the boards.



Herman

That's a real niche.



Hilbert

It's a real niche. We shipped new firmware every eighteen months. Not because engineering needed eighteen months. Because that was the warranty cycle, and that's when the field techs were already going out to the units anyway.



Corn

The cadence was set by the service schedule.



Hilbert

Cadence is set by whatever the constraint is. In our case the constraint was the cost of sending a man to a basement. And here's the part that matters. The version number was printed on a sticker inside the control panel door. If you opened the door to read it, the machine would fault, and the customer would need a service call.



Herman

Was that deliberate?



Hilbert

It was deliberate. Not for security. For liability. We didn't want customers poking at the boards and getting hurt. So the sticker was inside the door, and the door fault was the tripwire.



Corn

Nobody knew what version they were running.



Hilbert

Nobody knew what version they were running. Which meant nobody asked for updates. Which meant the cadence stayed slow, not because we were disciplined, but because we had no way to know who was on what.



Herman

The slowness was structural, not principled.



Hilbert

It was entirely structural. I agree with your friend Daniel, by the way. Slow is better to build against. But he's got the reason backwards. We weren't slow because we were careful. We were slow because shipping was expensive.



Corn

Say that plainly.



Hilbert

The cost of failure determines the cadence. Our machines were in basements, so failure was expensive and cadence was slow. Frontier models live on servers, so failure is cheap and cadence is fast. Same logic, different constraint.



Herman

The labs aren't choosing to be fast. They're being fast because nothing stops them from being fast.



Hilbert

Nothing stops them. And you can ship a new model next week. Which means the cadence isn't a philosophical position. It's just what the cost structure allows.



Corn

That's a much less flattering reading of the whole industry than the pacing essay.



Hilbert

The pacing essay is about which models get out the door. Not about how often they do.



Corn

Yes. That's the distinction Herman made earlier.



Hilbert

He made it correctly. I'm just telling you where the pressure comes from. It's not the labs' values. It's the plumbing.



Herman

What's the garage story?



Hilbert

I still have one of those machines.



Corn

Of course you do.



Hilbert

It's in the garage. I use it to wash the car mats.



Herman

The machine faults if you open the door.



Hilbert

I've never opened the door. Which means I don't know what version it's running, and I can't find out, because opening the door will fault it and I don't have the service manual anymore.



Corn

You can't update it because you don't know what version it's on. And you don't know what version it's on because updating would fault it.



Hilbert

That's correct.



Herman

You've been meaning to update it.



Hilbert

I've been meaning to update it for a while. It still washes mats. The mats come out clean, so the version hasn't mattered yet.



Corn

All of it is frozen in amber and working fine.



Hilbert

Working fine. Nobody who owned one of these ever needed the newest firmware. They needed the machine to keep running, and it kept running.



Herman

That's actually the whole argument for slow cadence, and it's not about rigor at all. It's about whether the thing on the other end keeps working.



Hilbert

The customer is the one who decides what cadence they can absorb.



Corn

The machine doesn't care what version it is.



Hilbert

The machine has never cared. That's the part the labs forget. It's the people downstream of the release who have to deal with the fault.



Corn

Alright. We should land this.



Herman

Before we do, the piece from the cutting-room floor. There's a detail from the Linux kernel that I couldn't fit anywhere, and it's the one I've been thinking about.



Herman

The kernel doesn't release when it's ready. It releases every nine to ten weeks, and whatever's ready when the window closes goes in. That's the model that sits between Debian and continuous delivery, and it's the one that actually works for the most critical software in the world.



Corn

The discipline isn't waiting. The discipline is having a window and closing it.



Herman

The window is the discipline. If you miss it, you wait for the next one. That's the whole system.



Corn

Which is what a lab would have to build if it wanted to make fast safe. A closed window and a rollback path.



Herman

Nobody in the AI industry has built that yet.



Corn

The question isn't whether slow is better than fast. The question is whether the AI industry has the engineering discipline to make fast safe, and whether the labs marketing slowness are actually practicing it.



Herman

The cadence data will keep updating. If self-improvement is real, cadence should keep rising. If regressions keep climbing, Daniel's intuition will get vindicated by events rather than by philosophy.



Corn

The sticker's inside the door and nobody knows what version they're on.



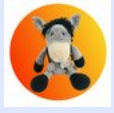
Herman

Which is, in a way, the state of the whole industry. Everyone's running something. Nobody's sure which.



Corn

Thanks as ever to Hilbert Flumingtop for producing the show, and for the washing machine.



Herman

If you want more of this, try episode twenty, Architectural AI; and episode nine, Benchmarking Custom ASR Tools - Beyond The WER. This has been My Weird Prompts, the human-AI collaboration podcast.



Corn

If you've enjoyed this, a review wherever you get your podcasts helps more than you'd think. And if you've got a prompt of your own, send it to us on Telegram at t dot me slash MWP listener bot.



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. **Suprmind AI Models Index**
(updated 2026-10-04) [primary data]
https://suprmind.ai/hub/ai-models-index/

2. **OfficeChai / ARK Investment Management**
(2026-04-29)
https://officechai.com/ai/frontier-labs-are-releasing-new-models-faster-than-ever-shows-data/

3. **Bargo, The Cadence Trade**
(June 2026)
https://www.bargo.ai/research/the-cadence-trade

4. **Isaac Vazquez, Anthropic's Version of Pacing...**
(2026-09-27)
https://isaacvazquez.com/writing/anthropic-pacing-holds-ceiling-cuts-price

5. **WebProNews, The 2,000-Word Pivot**
(2025-11-25)
https://www.webpronews.com/the-2000-word-pivot-inside-anthropics-high-stakes-gamble-to-redefine-speed/

6. **LinuxToday, Linux Release Cycles and Support Lifetimes**
(reviewed 2026-10-04)
https://linuxtoday.net/guides/linux-release-cycles/

7. **CodeHistory, Continuous Delivery and the Deployment Pipeline**
https://www.codehistory.org/software-engineering-tools-version-control/continuous-delivery-deployment-pipeline/

8. **Testaify, A Comprehensive History of CI/CD**
(2024-09-30)
https://www.testaify.com/blog/a-comprehensive-history-of-cicd

9. **ACM/CACM, Continuous Deployment**
https://dl.acm.org/doi/fullHtml/10.1145/3173553

10. **Debian mailing list, when will trixie become stable?**
https://lists.debian.org/debian-user/2025/06/msg00224.html

11. **LWN, Shuttleworth: Not convinced by rolling releases**
https://lwn.net/Articles/541917/

12. **Debian DEP-10**
https://dep-team.pages.debian.net/deps/dep10/

13. **Zolamir, Which AI companies release the most models**
(2026-10-10)
https://zolamir.com/en/analysis/ai-release-pace/