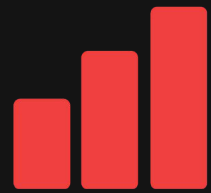


The state of the tech industry *in 2026*

what changed, what did not, and what broke

Gergely Orosz ·  THE PRAGMATIC ENGINEER

Hi, I'm Gergely



THE PRAGMATIC ENGINEER

Software engineering + AI engineering
at Big Tech, from the inside



The Software Engineer's Guidebook

Gergely Orosz

Navigating senior, tech lead, and staff engineer positions at tech companies and startups

 The Pragmatic Engineer



Hands up if you're an IC



Hands up if you're someone with reports



You pushed / contributed to production code the
last week



If you wrote it by hand



... if you had agents write the code



Is this not wild to see?

Would you have believed this even
~a year ago?

Today, we'll talk about this a massive change

**But first, an example of how unbelievable
our current timeline is**

March

“Meta is planning sweeping layoffs that could affect 20% or more of the company as Meta (...) prepares for greater efficiency brought about by AI-assisted workers.”

—Reuters, 17 March 2026

April

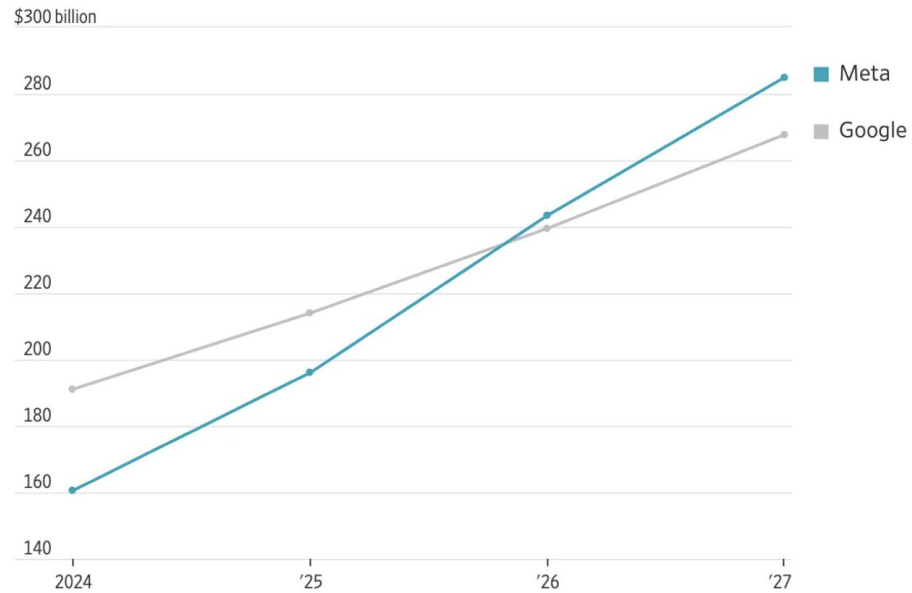
The Pragmatic Engineer

Meta on track to become a bigger ads business than Google

The Wall Street Journal [reports](#) that Meta is [on track to generate more ads revenue than Google this year](#), thanks to Meta's ads business growing around 24%, while Google's growth is around 11%.

Meta Rises

Global net ad revenue, annually



Note: Net ad revenue after companies pay traffic-acquisition costs. 2026 and 2027 numbers are estimates.
Source: Emarketer

Projected annual revenue from ads, in billions. Source: [The Wall Street Journal](#)

April

“Meta is planning sweeping layoffs that could affect 20% or more of the company as Meta (...) prepares for greater efficiency brought about by AI-assisted workers.”

—Reuters, 17 March 2026

May

The Pragmatic Engineer

3. Meta forcefully assigns engineers to data labelling, pre-layoff

Another detail I unearthed this week was from talking with Meta engineers, who say many teams there have had around 20-40% of their devs forcefully reassigned to data labelling jobs. There are easily 1,000+ software engineers doing data labelling at Meta now.

May

10% layoffs happen on 20 May

May



SID'S

BLOG



The Newest Instagram "Exploit" is the Goofiest I've Seen

Jun 1, 2026 | 609 words | 4 min read

Yesterday, a slew of Instagram accounts, including some high profile ones like the Obama White House account, seemingly got hacked.

Look, I'm no spring chicken. I've spent almost a decade and a half identifying vulnerabilities and exploits at unicorn scale, but this is hands down the most unserious, "almost too stupid to be true" of them all.

The Takeover Flow

- **Step 01: Faking the Location & Initiating Support**

All the attacker needs to kick this off is your account username. Then, they hop on a VPN or proxy close to your city so Instagram's security algorithms don't suspect a thing. (You can quite easily get this from your public profile or "About" section or a hundred other ways.) Once it looks like the request is coming from the correct region, they tell the Meta support AI that the account is hacked and ask it to send the verification codes to an arbitrary email address they control.

- **Step 02: That's It**

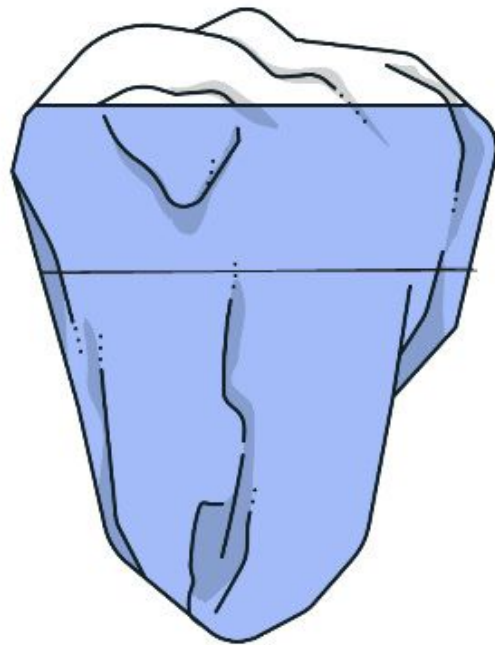
Really, that's it. The first proper zero auth password reset I've seen in production. There appears to be no additional check as to whether the email being given is actually something the user has used before. Once the AI sends the security code to the attacker's email, the attacker passes it right back to complete the verification. The platform hands over a fresh password reset link, granting full ownership to the attacker.

State of Meta in May 2026:

- 10% laid off
- + core Instagram / WhatsApp / Facebook teams lost 20-40% staff to manual data labeling
- + ongoing pressure to show high token usage (aka 'tokenmaxxing')

State of Meta from May 2026:

Most engineers at Meta, today,
will be looking for a way out



Laid off: 10%
obviously looking for work

Reassigned to data
labeling: ~20-30%
"I need a new job that is
software engineering"

Not laid off, not
reassigned: ~60-70%
"I can be reassigned to data labelling
or laid off anytime. I need to get out."

... and engineers, indeed, started to leave

The Pragmatic Engineer



THE PULSE

The Pulse: Meta's self-inflicted resignation-wave

The social media giant is offering \$1M+ retainer equity grants to staff who are leaving: and even this is not effective.



GERGELY OROSZ

AUG 14, 2026 • PAID

So, why did Meta decide to nuke its engineering
org?

It all started in Jan 2026:

Jan 2026:

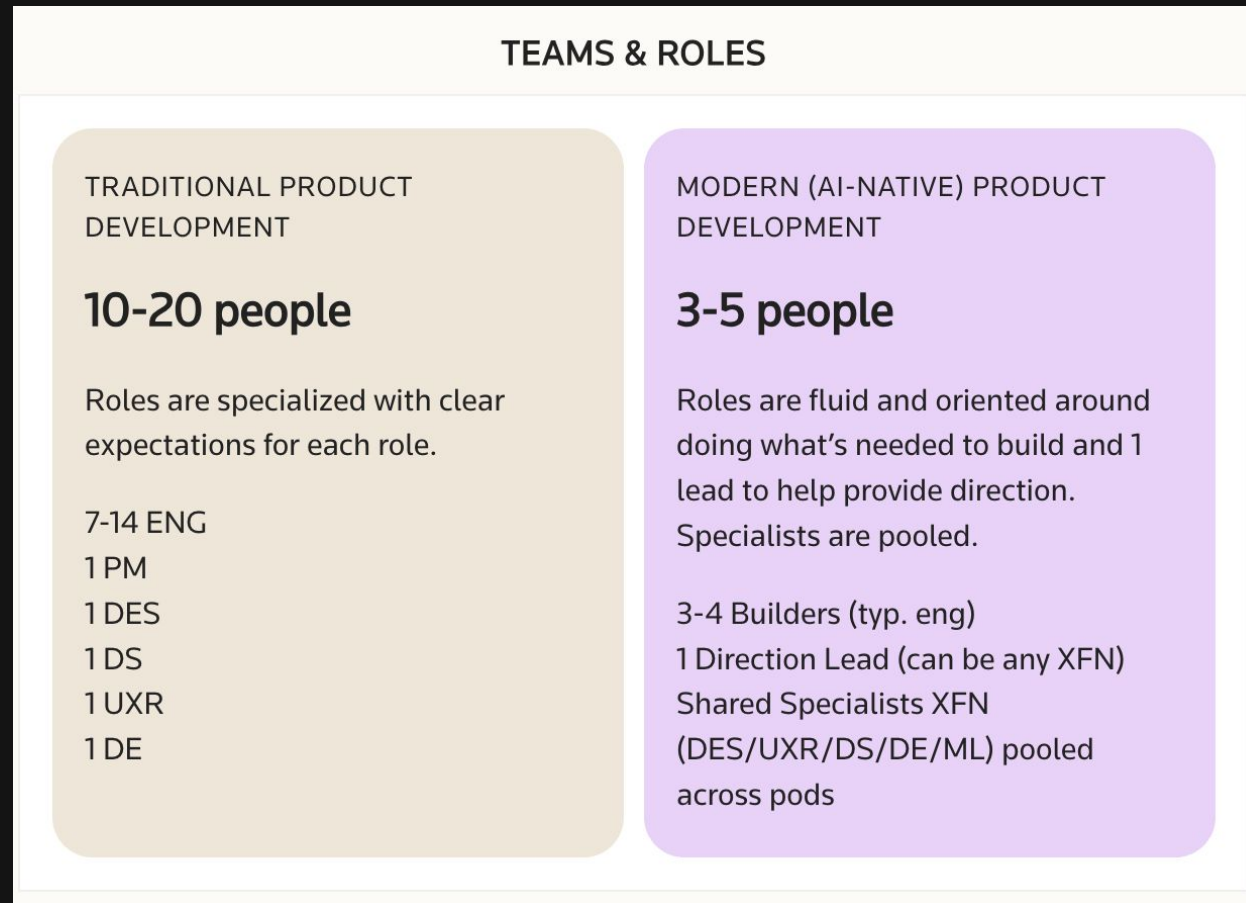
“In January, Meta CEO Mark Zuckerberg and his top lieutenants gathered for their annual leadership retreat at his Hawaii compound.

In scenario-planning exercises, two of these people said, executives explored slashing the size of many teams across Meta by as much as 60%.”

– Reuters, 26 Aug 2026

Meta's reason for wanting to disrupt itself, fast:

Wanting to get to “AI-native” teams ASAP



Meta: 22 years old

75,000 staff

\$68B profit

\$228B revenue

\$2.8M rev/ employee

... and still, they decided to do a massive,
risky change!

Was Meta acting irrationally

... or is there a *massive* change coming,
which many others *also* see?

Let's get this out of the way:

We are in the middle of the
largest-scale scale technology change
where
the timeline is so short (vs past shifts)



“Nothing before has hit with the magnitude of AI. This is a whole size difference from anything that we've faced before.”

– Martin Fowler, Feb 2026

We are most certainly at the beginning of a massive
technology transformation:

Once it finishes, software will be built
differently than how we've been used to it,
the last 20+ years

How does all this look like inside
AI labs, Big Tech and startups?

The last few months I visited the HQ + talked with:

- OpenAI
- Anthropic
- Cursor
- Ramp
- Uber
- Linear



Anthropic



OpenAI



Ramp



Cursor

Today, we'll cover:

#1 – What changed

#2 – What did not change

#3 – What broke

#4 – What comes next?

Today, we'll cover:

#1 – What changed

#2 – What did not change

#3 – What broke

#4 – What comes next?

WHAT CHANGED

1. Nobody writes code by hand anymore

The Pragmatic Engineer

DEEPDIVES

When AI writes almost all code, what happens to software engineering?

No longer a hypothetical question, this is a mega-trend set to hit the tech industry



GERGELY OROSZ

JAN 06, 2026 · PAID



1,010



40



98

Share



Over the past few weeks, experienced software engineers have shared personal “a-ha” moments about how AI tooling has become good enough to use for generating most of the code they write.

WHAT CHANGED

2. Working with several parallel agents: increasingly common

“I have 5 terminal tabs. Each one of them has a checkout of their repository. I’ll round robin and start Claude Code in each one.

I also run 5-10 Claudes on Claude Web, in parallel with my local Claudes.”

—Boris Cherny, creator of Claude Code, March 2026



WHAT CHANGED

2. Working with several parallel agents: increasingly common

“It used to be a simple world: one mouse, one keyboard, one screen. You physically can't work on more than one thing at once. Now, you no longer have that limitation.

I end up having 5-10 (literally) worktrees locally that I rotate between. Prompt one agent, while it is working move on to another to test output or review code. ”

—Dima Zaytsev, software engineer, Linear

WHAT CHANGED

2. Working with several parallel agents: increasingly common

“Oftentimes, I’m doing things in parallel. I find my cognitive overhead is about 5-10 agent sessions concurrently. But sometimes those sessions will have many subagents doing things.”

—Peter Mattis, cofounder & CTO, Cockroach Labs



WHAT CHANGED

3. The importance of the IDE is fading

- “Heavyweight” IDE usage: trending down
- Antigravity: the last major IDE forking VS Code (Nov 2025)
- Codex: considered forking VS Code.... and are glad they did not
- JetBrains: pivoting to “Air” (Aug 2026)

WHAT CHANGED

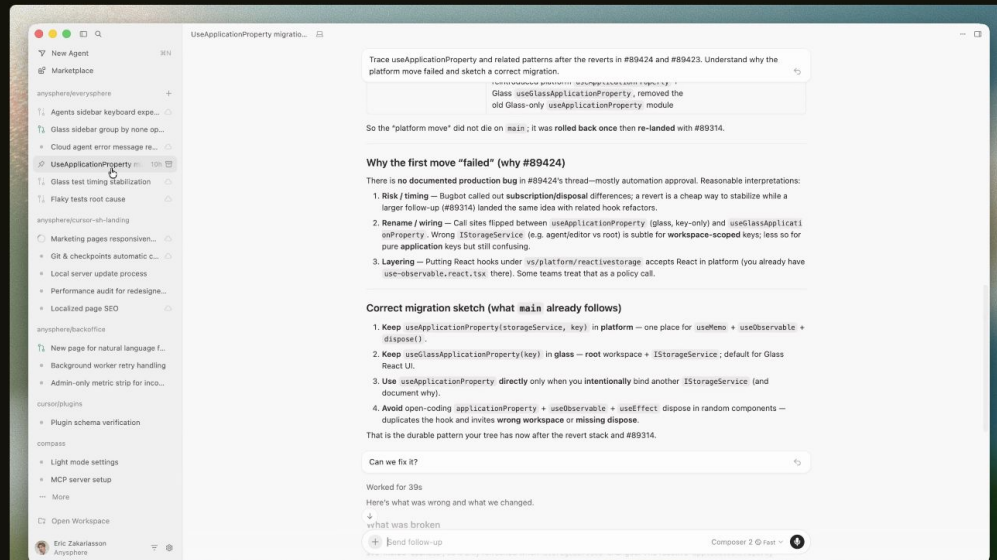
3. The importance of the IDE is fading

What's new in Cursor 3

When we started building Cursor, we forked VS Code instead of building an extension so we could shape our own surface. With Cursor 3, we took that a step further by building this new interface from scratch, centered around agents.

All your agents in one place

The new interface is inherently multi-workspace, allowing humans and agents to work across different repos.



WHAT CHANGED

3. The importance of the IDE is fading

“On the death of the IDE. It’s not that we don’t need more perspective & context for our human-based decisions. It’s that the context has changed.”

—Kent Beck

WHAT CHANGED

4. Everyone is building their own harness / agent platform



Gergely Orosz



@GergelyOrosz

If you've not built your own AI coding harness by now, are you even a serious tech company?

4:46 PM · Aug 17, 2026 · **197.7K** Views



101



95



1.3K



567



WHAT CHANGED

4. Everyone is building their own harness / agent platform

Ramp (Inspect)	Stripe (Minions)	Uber (Minion)	Block (Goose)	Shopify (River)
Google (Agent Smith)	Meta (Devmate)	Amazon (Kiro + Crew)	Dropbox (Nova)	Spotify (Honk)
DoorDash (Flux)	Grab (LLM-Kit)	WorkOS (Horizon)	Monzo (Agent Chip)	Hubspot (Crucible)
Sierra (Pinecone)	Harvey (Spectre)	Browserbase (bb)	... and many more...	

WHAT CHANGED

5. Dev work often starts in Slack

- Inside OpenAI: “@Codex implement this”
- Inside Anthropic: “@Claude do this”
- Startups: building their own Slack AI agent bot integrations
- Inside Linear: “@Linear work on this”

WHAT CHANGED

6. The agentic software factory: it's being built

- Startups: building their own Slack AI agent bot integrations
- Inside Linear: “@Linear work on this”
- Inside OpenAI: “@Codex implement this”
- Inside Anthropic: “@Claude do this”

WHAT CHANGED



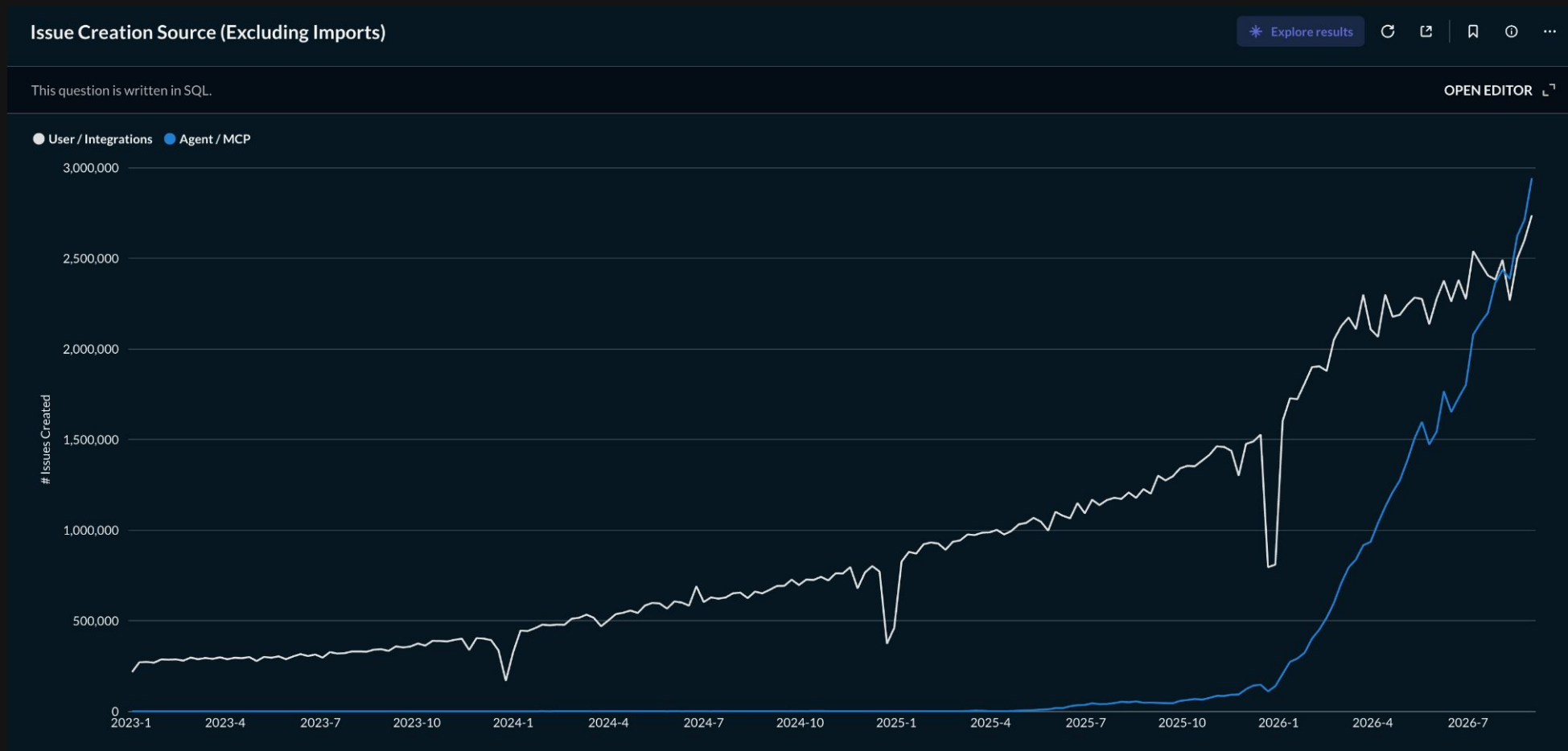
WHAT CHANGED

7. Migrations no longer take years

- **Anthropic:** Bun migration Zig → Rust: 11 days (vs ~years!)
- **OpenAI:** API migration Python → Rust: months (vs years!)
- **Airbnb:** Enzyme → RTL migration: 6 weeks (vs years!)
- **Asana:** Enzyme → RTL migration: 2 weeks (vs years!)
- **Uber:** JUnit 4 → 5 migration: 4 months (vs years!)

WHAT BROKE

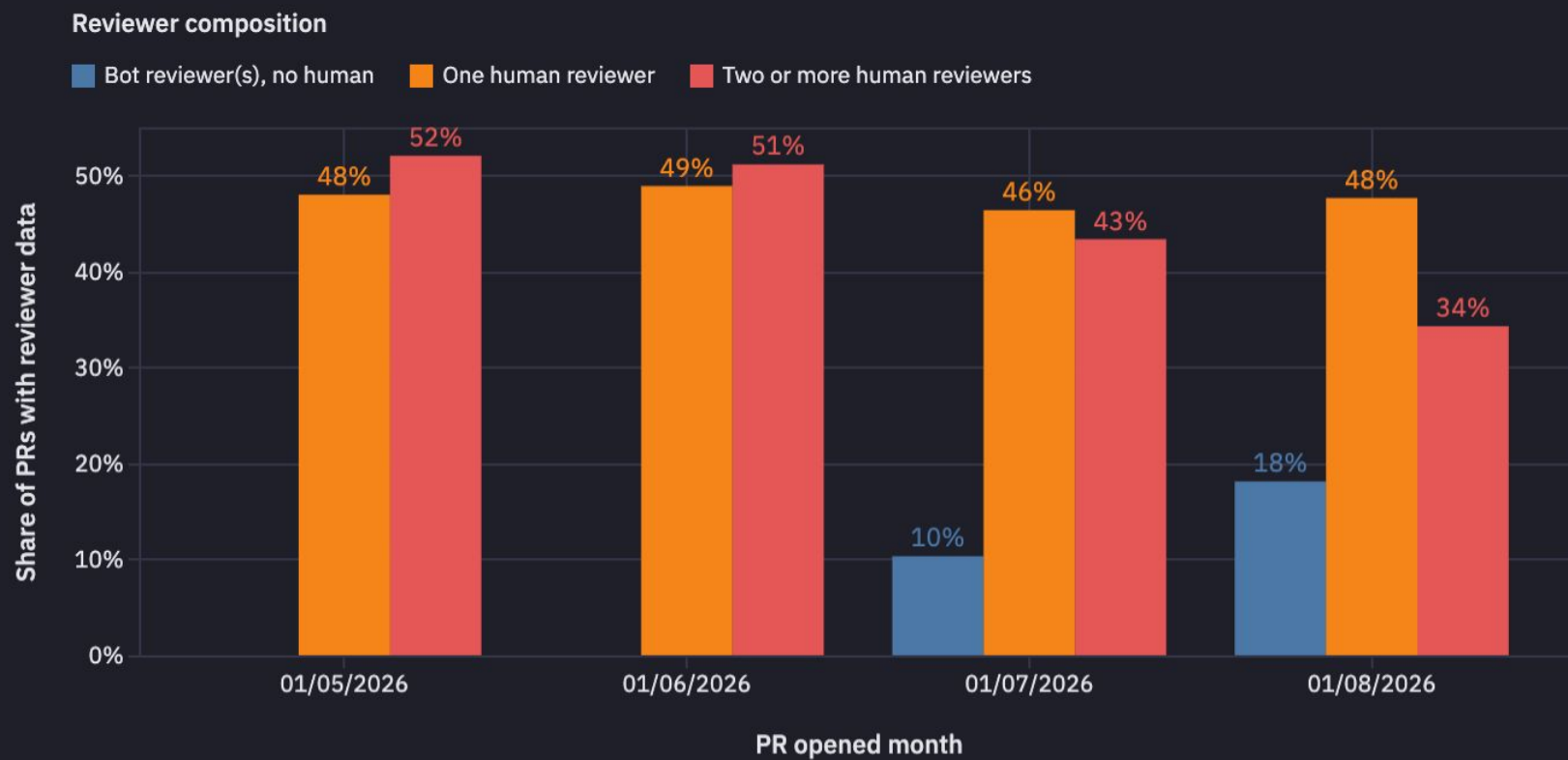
8. Agent-generated issues: rapid rise



Source: Linear

WHAT CHANGED

9. Agent-generated PRs: rapid rise

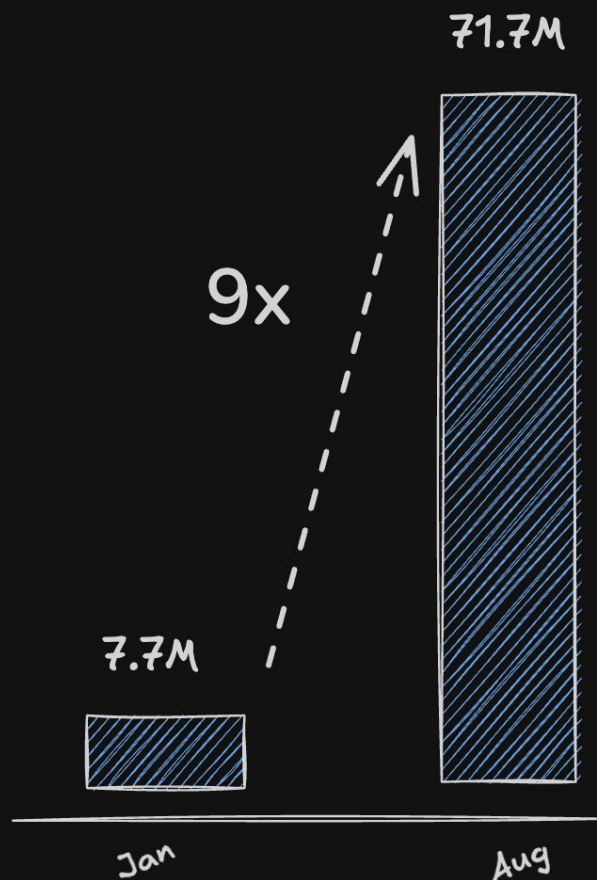


Source: Linear

WHAT CHANGED

9. Agent-generated PRs: rapid rise

of Agent-authored PRs

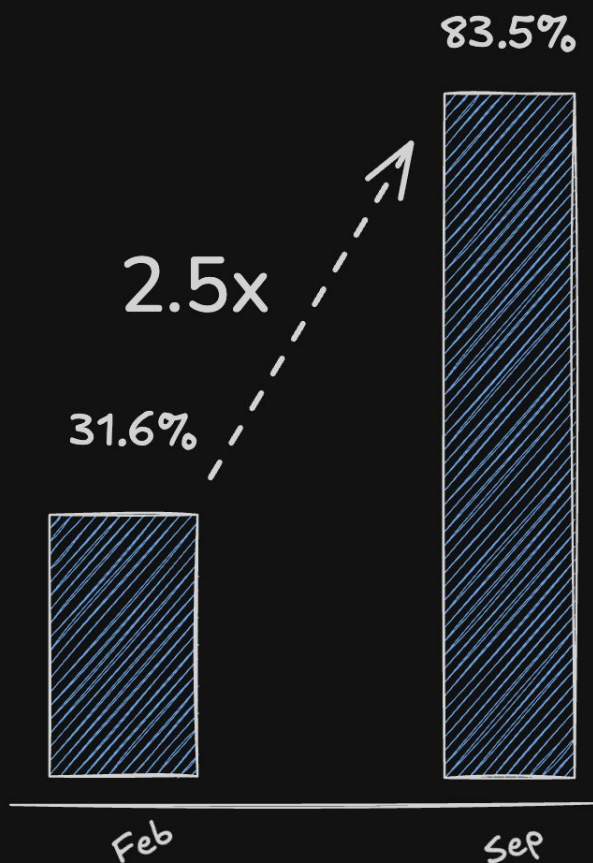


Source: GitHub

WHAT CHANGED

10. Agent skills: usage is now widespread

% of users using skills



Source: Factory AI

WHAT CHANGED

11. AI cost: a major engineering concern

The Pragmatic Engineer

- **OpenCode: customer demand for optimizing spend is spiking.** Yesterday, on the podcast episode with OpenCode creator, Dax Raad, he said demand for OpenCode's hosted inference service ([OpenCode Zen](#)) surpassed all expectations because larger companies want cheaper, but still capable, AI models. He revealed that over the past month, *every single* inbound enterprise request was about optimizing spend. So, there's some widespread concern about AI bills.

– May 2026

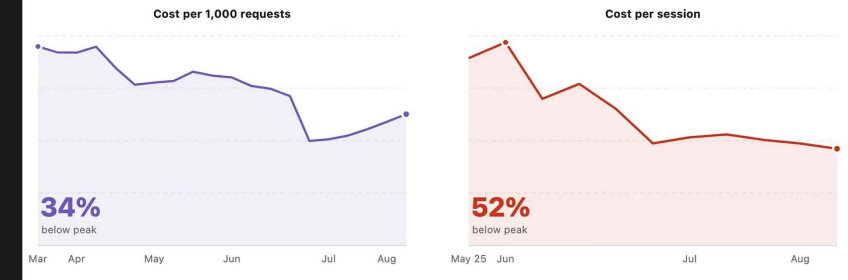
WHAT CHANGED

11. AI cost: a major engineering concern

- Uber: blew through 2026 AI budget by March
- Started to optimize AI spend
- Cut back cost-per token: open models
+ smart routing

The Pragmatic Engineer

UNIT COST WITH THE MODEL HELD CONSTANT, 2026



Reducing per-token and per-session spend. Source: [Uber](#)

AI usage keeps going up; overall costs now stay the same

WEEKLY ADOPTION AND COST, FEB-AUG 2026



THE PRAGMATIC ENGINEER

Usage up, cost stable Source: [Uber](#)

WHAT CHANGED

11. AI cost: a major engineering concern

The Pragmatic Engineer

THE PULSE

The Pulse: tech companies move to open AI models

Cost-saving efforts reveal that moving simpler workloads to open AI models is the easiest way to save ~50% on AI bills. Uber, Pinterest, Stripe, Coinbase, Ramp, and AT&T are making large savings on their AI bills by dropping proprietary models and using smart model routing.



GERGELY OROSZ

SEP 03, 2026 • PAID



90



8



3

Share



WHAT CHANGED

12. Engineering specializations are disappearing

“Although I always thought specialization in engineering would, one day, go away, it’s still surprising how quickly it has happened. Last year, teams were still looking for an Android or iOS engineer for specialized work. But no more. Today, if something needs to get done, you just need a builder.”

—Sulman Choudhry, ChatGPT’s Head of Engineering, June 2026

WHAT CHANGED

12. Engineering specializations are disappearing

“Although I always thought specialization in engineering would, one day, go away, it’s still surprising how quickly it has happened. Last year, teams were still looking for an Android or iOS engineer for specialized work. But no more. Today, if something needs to get done, you just need a builder.”

—Sulman Choudhry, ChatGPT’s Head of Engineering, June 2026

WHAT CHANGED

13. Projects: one or two engineers

“On an individual project, you often cannot have more than two people working on it.

This is because each engineer is already running several agents. And so as an engineer, you’re already fighting against your agents, which are stepping on each other’s toes on implementation. And in this setup, you just cannot have that many humans, who also come with all their agents!”



—Katelyn Lesse, Head of Engineering for Claude Platform, July 2026

WHAT CHANGED

Other changes

- 14. Teams: getting smaller
- 15. Junior hiring: less happening
- 16. Agentic infra: becomes its own discipline

#1 – What changed [16 things]

#2 – What did not change

#3 – What broke

#4 – What comes next?

WHAT DID NOT CHANGE

1. Teams are still here

“One thing I’ve heard from some people is “we have two humans and a bunch of agents.” I reply that this isn’t where we’re at:

I still have teams whose job it is to own a piece of software, iterate on it, own oncall, and so on. While each of these humans are supercharged by AI, the size and shape of the team is still similar. We still have two-pizza teams.”

—Katelyn Lesse, Head of Engineering for Claude Platform, July 2026



WHAT DID NOT CHANGE

2. Planning: still here, for complex work

- Complex projects at Anthropic: still have similar planning as “pre-AI”
- Building the wrong thing is still a waste: and planning helps alignment
- Planning especially important for:
infra complexity; needing to align teams; communicating with customers

WHAT DID NOT CHANGE

3. Tests + validation: still very important

“The AI writes pretty much all the code, but we have the AI write pretty much all the tests as well. Before AI, for production-ready software, we spent about the same time writing tests as we did on writing code. This is still the case with AI.

You need to have a way to trust your code, and tests are probably the best way to.”

—Jarred Sumner, creator of Bun, July 2026

WHAT DID NOT CHANGE

4. Non-engineers: still not shipping prod code

Non-engineers *should* be able to ship production code...

... but they're not shipping to prod!

As per details from Anthropic, OpenAI, Linear, Ramp.

WHAT DID NOT CHANGE

5. “Old” patterns are being rediscovered, again

*“After discovering ‘**tracer bullets**’ from *The Pragmatic Programmer*, I just started diving into different books, all the books I could find to try to mine them for leading words. Another one was John Ousterhout's book, *Philosophy of Software Design*, where I picked up tons of great stuff like **deep modules**.”*

—Matt Pocock, September 2026



#1 – What changed [16 things]

#2 – What did not change [5 things]

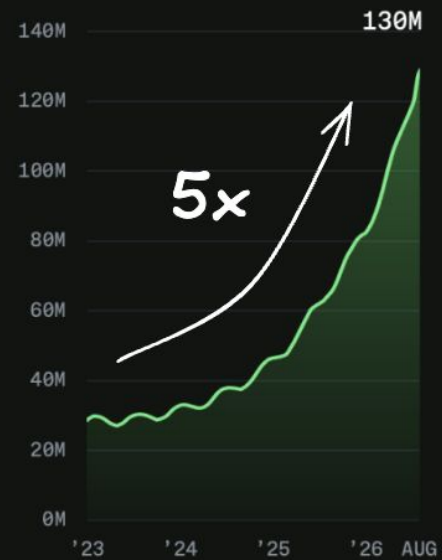
#3 – What broke

#4 – What comes next?≈

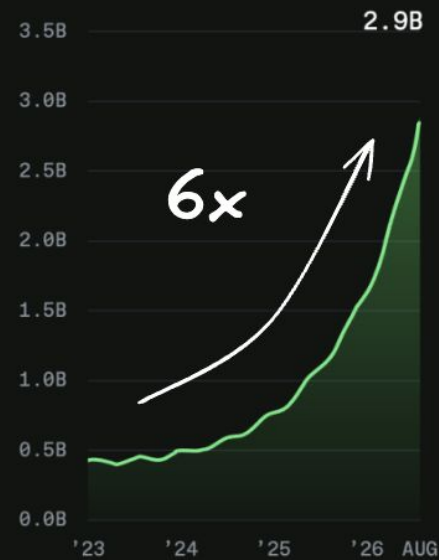
WHAT BROKE

1. Code quantity & frequency assumptions

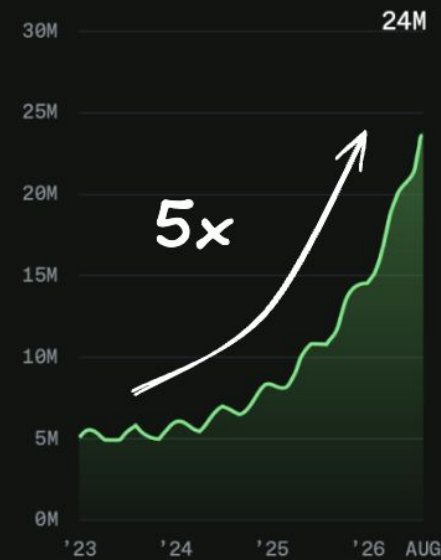
Merged pull requests per month



Commits per month



New repos per month



Source: GitHub

WHAT BROKE

2. Code reviews

“Everyone is playing the theater of doing reviews, but with the volume of changes that get thrown your way I observed that people just find a path of least resistance - some teammate that gave up on reviews and just stamps everything with LGTM.

We are gradually phasing out this process ourselves, with some shortcuts that you are allowed to take, and that has been a major productivity booster.”

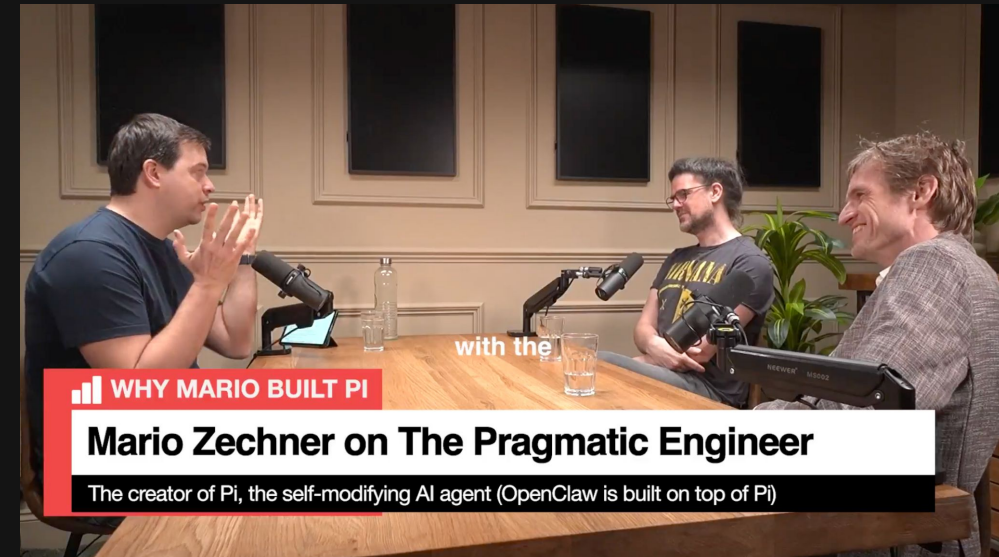
— Software engineer at a 70-person startup

WHAT BROKE

3. Quality & reliability

“Everything is broken. It sure feels like software has become a brittle mess, with 98% uptime becoming the norm instead of the exception, including for big services. And user interfaces have the weirdest bugs that you'd think a QA team would catch. I give you that that's been the case for longer than agents exist. But we seem to be accelerating.”

— Mario Zechner, the creator of Pi, March 2026



WHAT BROKE

4. Infra capacity

- GPU shortage across the industry
- Memory shortage across the industry
- CPU shortage starting now, as well

WHAT BROKE

4. Infra capacity: CPU shortage (as well)

The Pragmatic Engineer

THE PULSE

The Pulse: a new trend of CPU shortages

After a GPU shortage and memory shortage driven by AI companies, we're now experiencing a CPU shortage, thanks to AI agents using a lot more CPU with tool usage. **If you will need more compute in the future: secure it now**, while you can (even if it's expensive to do so).



GERGELY OROSZ

SEP 10, 2026 • PAID



67



4



1

Share



WHAT BROKE

5. Focus + productivity

“There’s lots of context switching: there’s always another agent that waits on your response.

Also, it’s just more work. When AI was ramping up, I felt more productive than my peers: I did in an hour, what others needed a day. Now, the expectation has become that everyone works on parallel things. And so I end up feeling as having more work, compared to pre-AI.

—Dima Zaytsev, software engineer, Linear

WHAT BROKE

6. Engineering leadership

The Pragmatic Engineer

DEEPDIVES

Headed for the Exit: the Great Engineering Leader Career Break

Trend: more CTOs, VPEs, and Heads of Engineering are walking away from their high-status, in-demand positions. There are many reasons, mostly related to AI, and to "founder mode"



GERGELY OROSZ
AUG 18, 2026 • PAID



358



11



18

Share



In my ~20 years in this industry, I've not seen as many capable engineering leaders opting out or taking prolonged breaks as now, with some high-ranking engineering leaders – CTOs, VPs of Engineering, heads of engineering, etc. – quitting their high-status roles and departing, if not into the sunset, then at least with nothing lined up.

WHAT BROKE

6. Engineering leadership: reasons folks are quitting

#1 – The job got (much) worse

#2 – The startup is “losing” and becoming worthless

#3 – Not adopting new AI workflows: and so being stuck in the “slow lane”

#4 – Smaller teams → fewer leaders needed

WHAT BROKE

6. Engineering leadership: reasons folks are quitting

#5 – Fractional CTO work: preferred over fulltime positions

#6 – AI startups pay ICs more than non-AI startups pay execs

#7 – Quitting to launch their own business

#8 – Burned out

#1 – What changed [16 things]

#2 – What did not change [5 things]

#3 – What broke [6 things]

#4 – What comes next?

WHAT COMES NEXT?

1. Cloud agents + harnesses

The Pragmatic Engineer

2. Why build your own background coding agent?

There are a few reasons why Ramp decided to turn down tried-and-tested products and create their own:

1. **Local machines are limited in how many agents they can run.** Ramp found third-party products below expectations; they liked Claude Code on day 1, but were constrained by only being able to run one or two sessions on local machines.
2. **Better frontend tooling.** The web engineering team wanted to improve their frontend tooling so designers could make small UI tweaks. There was an opportunity to use AI to automate themselves out of that loop.
3. **Need for remote dev environments.** As Ramp scaled, so did the complexity, and with it there was more work at the intersection of systems, like debugging backward compatibility, and broken API contracts. The solution was to create remote dev environments.

WHAT COMES NEXT?

1. Cloud agents + harnesses

“I started as a strong proponent of local development. The industry is still lagging behind on devpods and tooling for cloud development.

However, having worked with Linear coding sessions I can't help but dread having to pull a branch locally, starting it up, etc. It is just cumbersome.

All in all, I changed my mind - cloud is the future.

—Dima Zaytsev, software engineer, Linear

WHAT COMES NEXT?

2. We stop reading the code

“What would it take for you to be comfortable shipping code without you reading it and understanding it? Because that's engineering.”

— Charity Majors, cofounder and CTO at Honeycomb



WHAT COMES NEXT?

3. Building a new type of internal infra

- Evals part of CI/CD
- Agentic factories
- Agents part of deployments, observability, incident management

WHAT COMES NEXT?

4. A refactoring + migrations + rewriting wave

When it's cheap to refactor / migrate / rewrite: why hold off?

WHAT COMES NEXT?

5. “AI fluency” + “AI positivity”: baseline for hiring

“AI positivity was something we started to screen for, in our hiring process. We want people to join who want to help us push the limits with what we can build with AI.”

— Engineering Director, Series D startup

WHAT COMES NEXT?

6. Getting to AI maturity, ASAP

“Orgs who are moving faster and maturing do something different. They begin with a business outcome and work backward from there, building an agentic system that reduces decision handoffs, increases learning, and removes friction -- while maintaining quality.

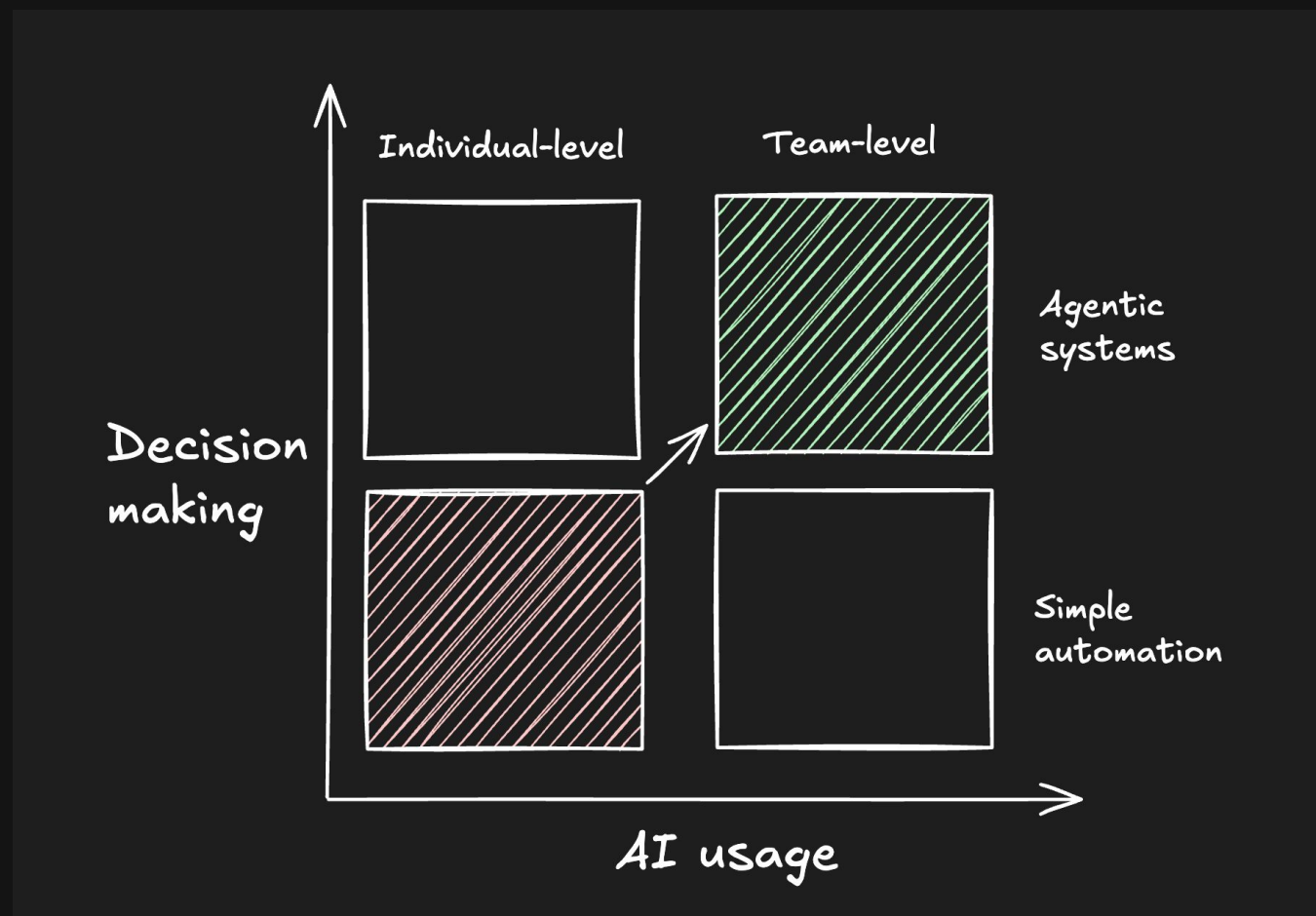
*AI maturity comes from applying AI to the *system*, not to the individual.*

— Laura Tacho, Senior Principal Technologist, Developer Experience, AWS



WHAT COMES NEXT?

Getting to AI maturity, ASAP



— Laura Tacho, Senior Principal Technologist, Developer Experience, AWS

WHAT COMES NEXT?

7. Engineers with deep domain knowledge: in-demand

“What you need to succeed: intelligence, wisdom, and charisma.”

Intelligence means: how to do it.

Wisdom means: what to do.

Charisma means: convince others to do it.

When intelligence becomes commonplace, wisdom and charisma become much more important.”

— Titus Winters, author of *Software Engineering at Google*



So, how do you thrive, as an engineering leader?

HOW DO YOU THRIVE?

Future-proof your career

- Get / stay hands-on building software
- Build AI infra + systems that help your team and company
- Use AI to remove friction: from your work, from your team's work, and your company's
- Don't "outsource" your learning
- Accept you'll do less people management (for now) – something needs to give!

HOW DO YOU THRIVE?

Lean in, learn more than before, and you'll be fine

“You just got to get in there, be using these tools all the time, and push yourself to learn.”

I feel like I've learned more in the past year than the previous five years combined, which is weird.”

—Peter Mattis, cofounder & CTO, Cockroach Labs

HOW DO YOU THRIVE?

Remind yourself: why did you get into tech?

“I got into software engineering because I like building stuff. Now I can build faster, and without some of the compromises that I had before.

It’s a little bit exhausting now, but it’s also very, very exciting.”

—Peter Mattis, cofounder & CTO, Cockroach Labs

Keep on keeping on

Subscribe to The Pragmatic Engineer to keep up with the industry.

Get three months of full access to the paid newsletter:

pragmaticengineer.com



END