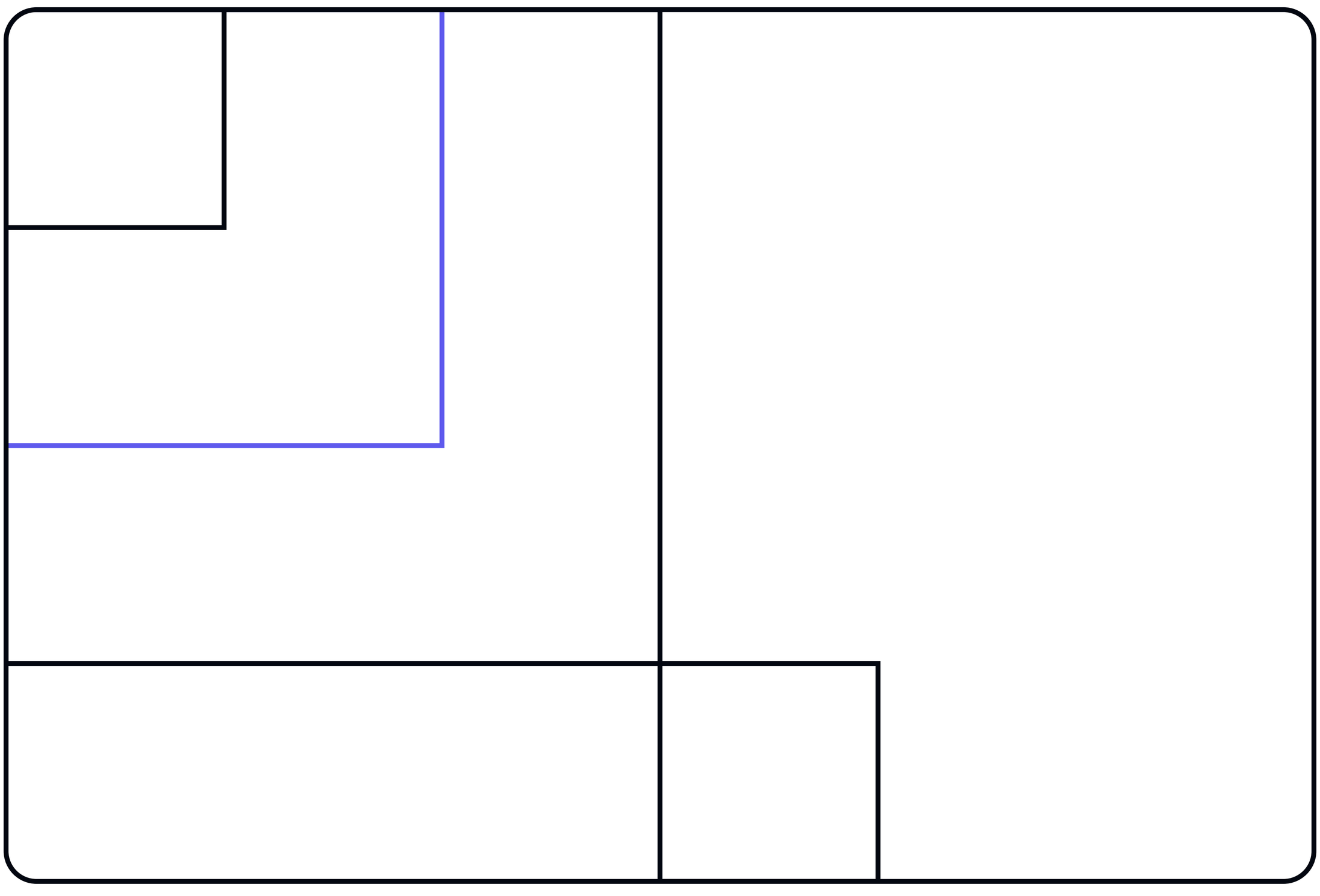
REPORT

AI-assisted engineering: Q4 impact report



AI-Assisted Engineering: Q4 Trends Report

Laura Tacho, CTO at DX

The current impact of AI

Over the past year, we've had a front-row seat to one of the most significant transformations in software development: the adoption of AI.

At DX, we help organizations measure developer productivity through a combination of qualitative and quantitative data. This unique vantage point has allowed us to see both patterns in the data and the real stories playing out on the ground.

To capture a clear picture of AI's current impact on our industry, this report draws on actual AI coding assistant usage data from over 135,000 developers across 435 companies[1]. We'll look at high-level trends and also dig into some deeper analyses, providing some clarity with a moment-in-time snapshot of what the current data shows.

Each analysis in this report reflects the subset of developers and companies with available data for that specific metric.

How to interpret these numbers

While industry averages can show broad trends, it's important to highlight that there is no "average experience" when it comes to AI impact. AI is an accelerant that is helping some companies accelerate and improve quality, while other companies are speeding up as quality and maintainability degrade.

There are many contextual factors that influence AI impact. At DX, we're placing more emphasis on longitudinal trends and cohort comparisons, rather than taking an industry average as an overall indication of what to expect across the board. The high adoption of AI tools makes it difficult to compare AI users with non-AI users across the industry, as the sample size for non-users gets smaller and smaller as weeks go by. Instead, many companies are opting to compare regular users (daily and weekly) with light users (monthly) to understand how increased usage impacts productivity.

[1] See the Methodologies section for more information on our data samples.

Industry-wide AI adoption is over 90%

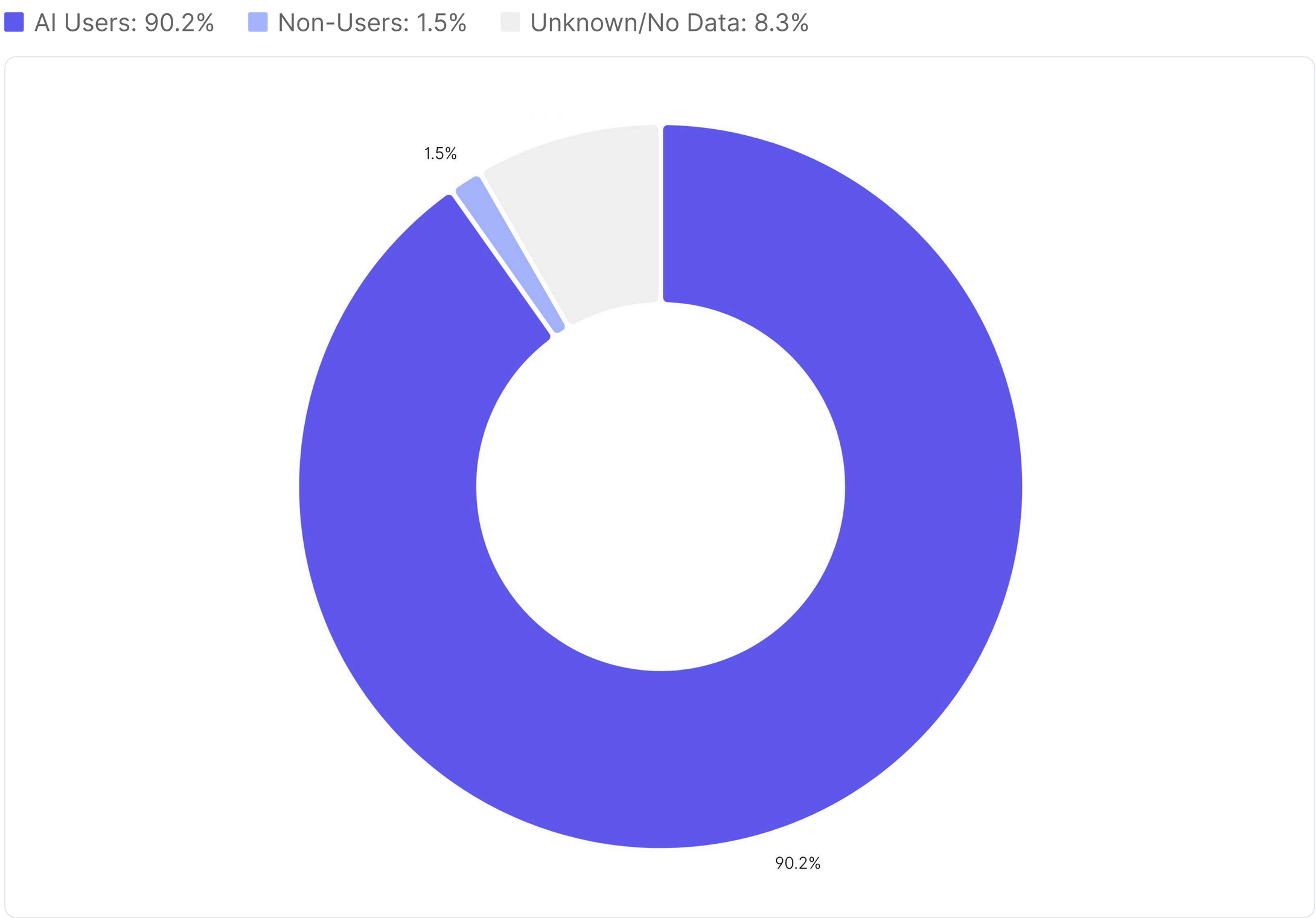
The use of AI coding assistants is now ubiquitous in engineering organizations, with overall adoption reaching 91%. Other industry analyses independent from DX, such as the 2025 DORA report, also stated an adoption rate of 90%. It's clear that using AI coding assistants is no longer an experiment – it's a core part of engineering strategy.

But adoption doesn't equal impact. Companies are struggling more than ever to understand how their AI investments are impacting their engineering performance.

To ground the conversation, we'll start with a fundamental question: how is AI changing the way that developers get their work done right now?

Industry-wide adoption of AI coding tools

Sample of 85,350 developers at 435 companies



Developers save an average of 3.6 hours per week with AI coding tools

One of the high-level signals we track at DX is self-reported time savings.[2] This gives an early indication of AI usefulness, and is a leading indicator for downstream impact, like faster time to market or more time for innovation.

Developers reported saving an average of 3.6 hours per week thanks to AI coding tools. Daily users saved the most time, with 4.1 hours, followed by weekly users at 3.5 hours a week.

Interestingly, though AI tool adoption has soared in the last six months, time savings per developer has hovered around the 4 hour mark. We haven't seen time savings rise as industry-wide adoption has increased.

One hypothesis is that, while developers may successfully use AI tools regularly for daily tasks, they're still hitting significant barriers at the team and system level. Another hypothesis is that developers have a harder time comparing their time savings to pre-AI levels, since many users have been using AI tools in their daily work for 6 months or more. However, it's important to note that overall time savings have nearly doubled since Q4 2024, when the average was closer to 2 hours.

We also see significant time savings reported from users without system-level usage data from their organization's AI tools, suggesting that AI use is pervasive, and confirming the high adoption rates mentioned in the previous section.

Weekly AI time savings by AI usage level

Sample of 58,330 developers across 352 companies

[2] See the Methodologies section for more on self-reported time savings.

22% of code is AI-authored

The percentage of AI-authored code is one of the most talked-about metrics in the industry right now—and also one of the hardest to measure accurately. Many of the figures in headlines aren't grounded in rigorous measurement. While we at DX are developing more sophisticated telemetry-based tools with our customers, we've also introduced a self-reported measure to establish a consistent baseline.

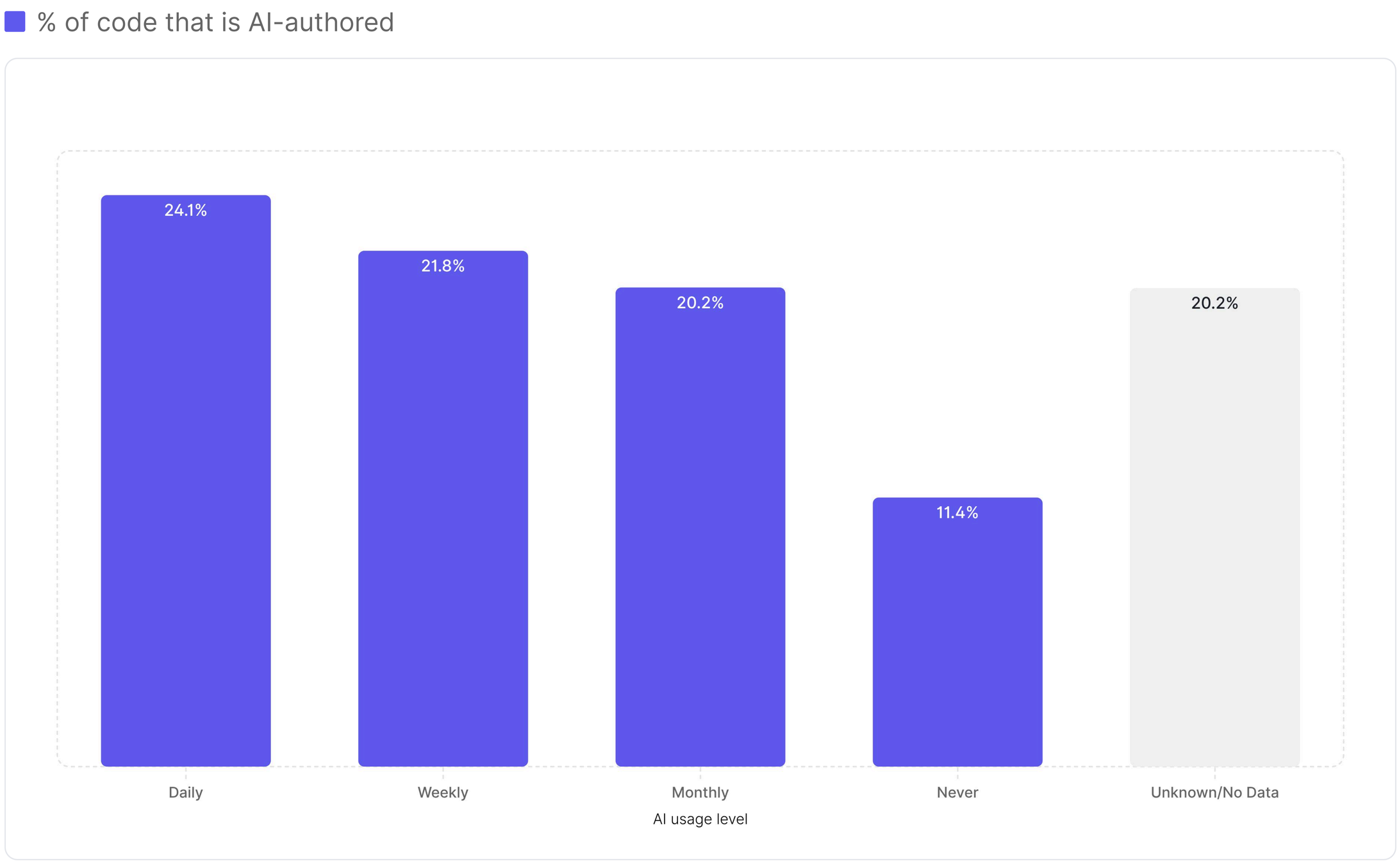
In our definition, "AI-authored code" is code generated by AI that was merged without major human rewrites or modifications. Using self-reported data aligned to this definition, data from Q4 2025 shows that 22% of code is AI-authored, looking at a sample of 266 companies.

Interestingly, the percentage of AI-authored code doesn’t change much based on AI usage intensity. Daily users of AI coding assistants report that 24% of their merged code is AI-generated, with monthly users reporting just over 20%.

The number provides an important reality check against vendor marketing claims. It also highlights a key point: while AI-generated code does not tell the full story of impact, it can be a meaningful signal when tracked alongside other data points.

Percent of merged code authored by AI, by AI usage level

Sample of 34,491 developers across 248 companies



Daily AI users ship 60% more PRs than non-users

One of the most common questions we hear is whether AI adoption is actually helping developers ship more code.

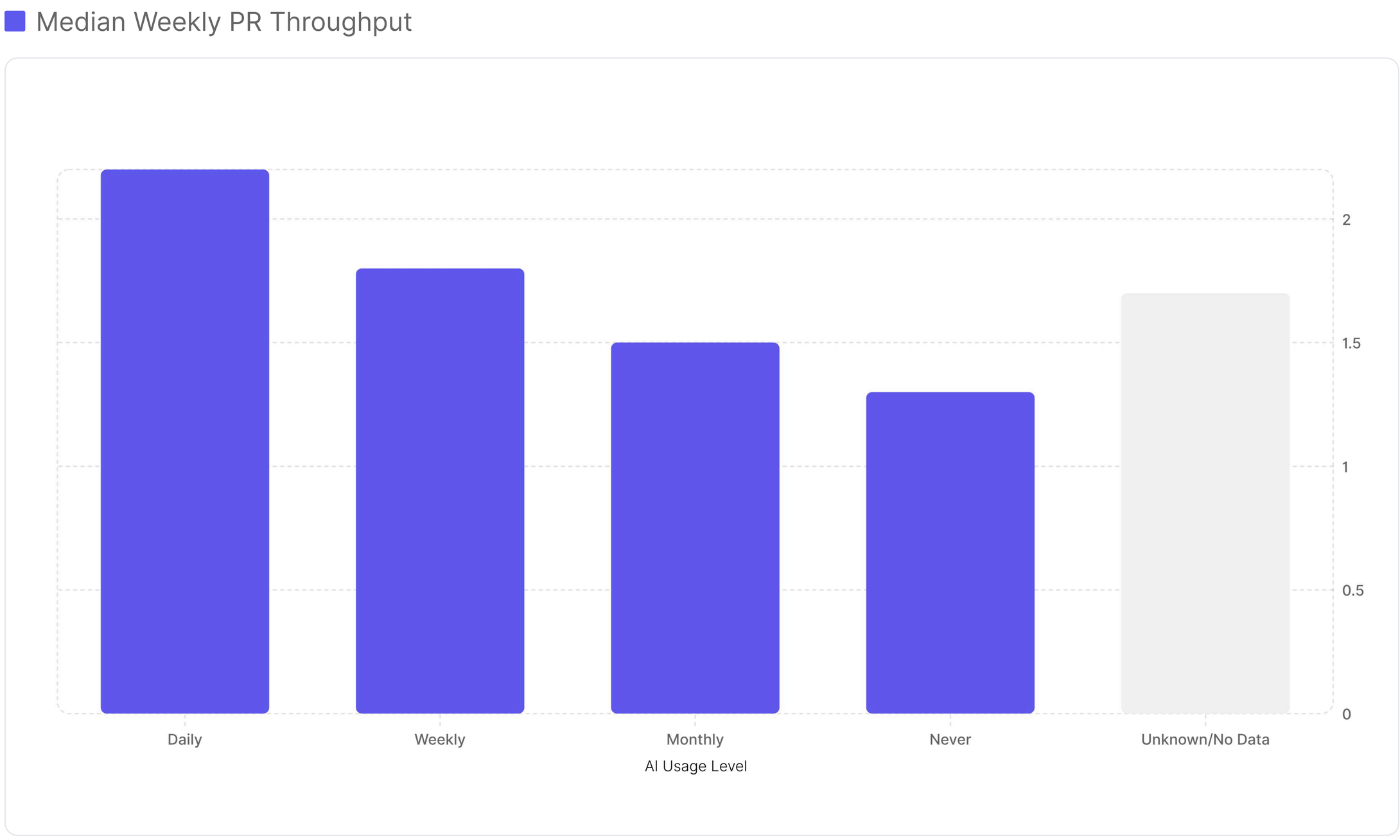
Looking across more than 51,000 developers, there is a correlation between frequency of AI usage and pull request (PR) throughput. Daily users of AI merge a median average of 2.3 PRs per week, followed by 1.8 PRs/week for weekly AI users, and 1.5 PRs/week for monthly AI users. A small group of users who report they don't use AI at all, or they don't have access, ship a slightly lower 1.4 PRs/week.

This throughput gap between heavy and light AI users has remained steady since the previous quarter, where the median value for PRs/week for daily AI users was 2.2, compared to 1.5 PRs/week for monthly AI users.

PR throughput, AI-authored code, and time savings are correlated with AI usage. Daily users of AI save more time coding, merge more AI-authored code, and merge more PRs overall compared to developers who use AI less frequently. However, it's still possible that this increase in speed doesn't lead to better business results. This scenario underscores the need for comprehensive, continuous measurement beyond just coding habits.

Weekly PR throughput by AI usage level

Sample of 61,800 developers at 376 companies



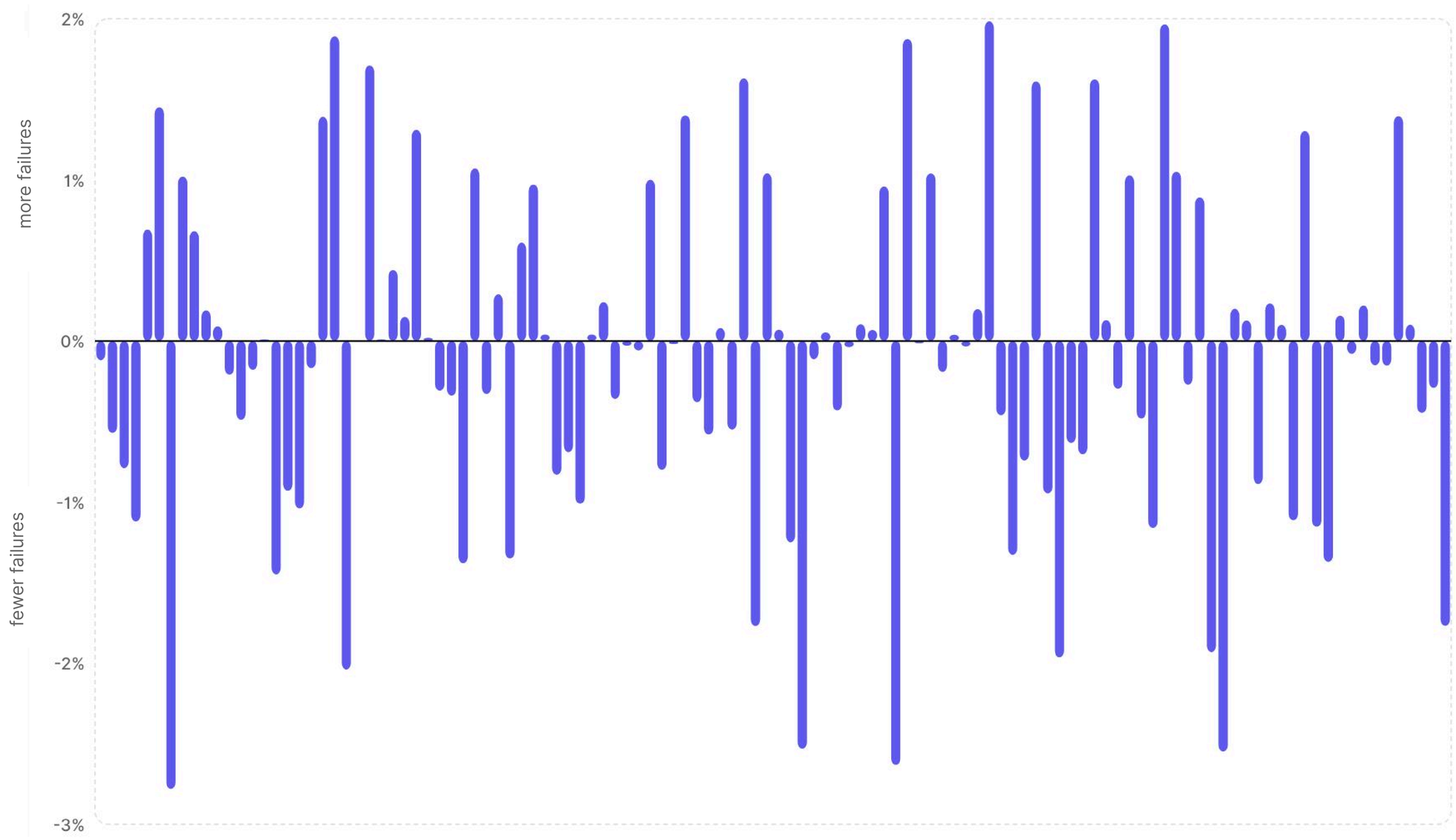
Impact on quality is varied

To track changes in quality, we looked at the relationship between AI tool usage and three measures for quality:

- Change Failure Rate: the percentage of changes that result in degraded performance that must be immediately fixed
- Change Confidence: how confident a developer is that a change will not break in production
- Code Maintainability: how easy it is to understand and modify code.

While some organizations are seeing clear improvement to quality as AI usage increases, others are seeing serious degradation. There can be multiple reasons for this, stemming from existing code hygiene practices, the availability of formal training on the best use of AI, and the size, complexity, and domain-specificity of the codebase.

AI's impact on Change Failure Rate, in % points from pre-AI measurement



For engineering leaders, these findings underscore that successful AI integration requires more than just tool deployment; it demands thoughtful measurement, implementation strategies, and proper training to ensure that the promise of increased velocity doesn't come at the expense of the code quality that underpins long-term software sustainability.

Interesting trends

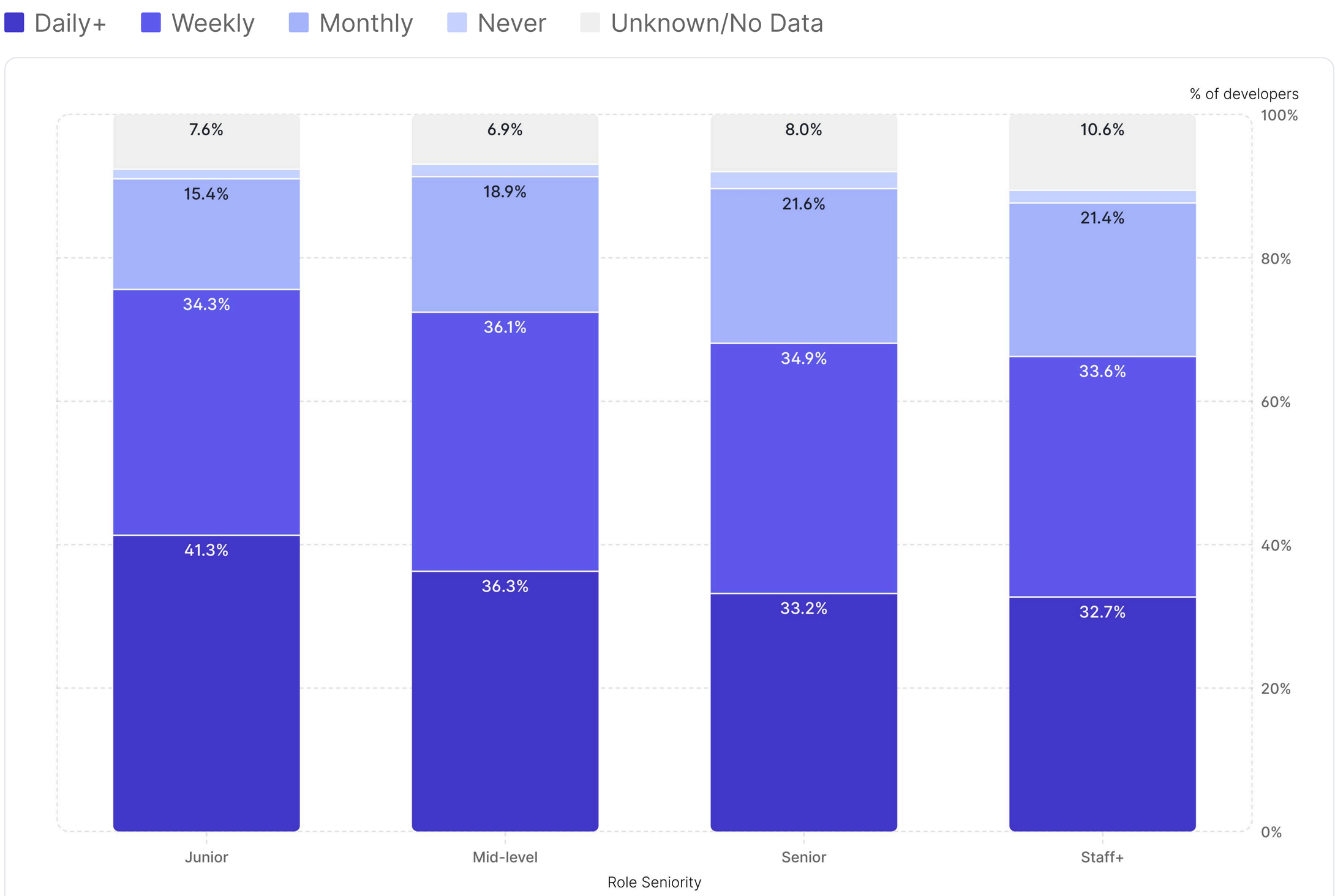
Junior engineers use AI the most

Adoption is an important prerequisite for impact, with many companies focusing their AI strategies on rollout and onboarding over the past 18 months. These efforts have paid off, with overall adoption reaching 90%. Still, it's important to analyze adoption patterns by developer attributes to understand which groups may need additional enablement support.

Our data suggests that junior developers use AI more frequently than more senior developers. Patterns like this in your own data can spark useful conversations. Are junior engineers more eager to experiment with new tools, or is it that AI coding tools are better suited for smaller, well-defined tasks?

AI adoption by usage level and role seniority

Sample of 20,200 developers from 125 companies



Big time-saving opportunities for senior and staff+ engineers

Time savings based on seniority level have levelled out since last quarter, with Staff+ engineers still saving the most time, but by a smaller margin.

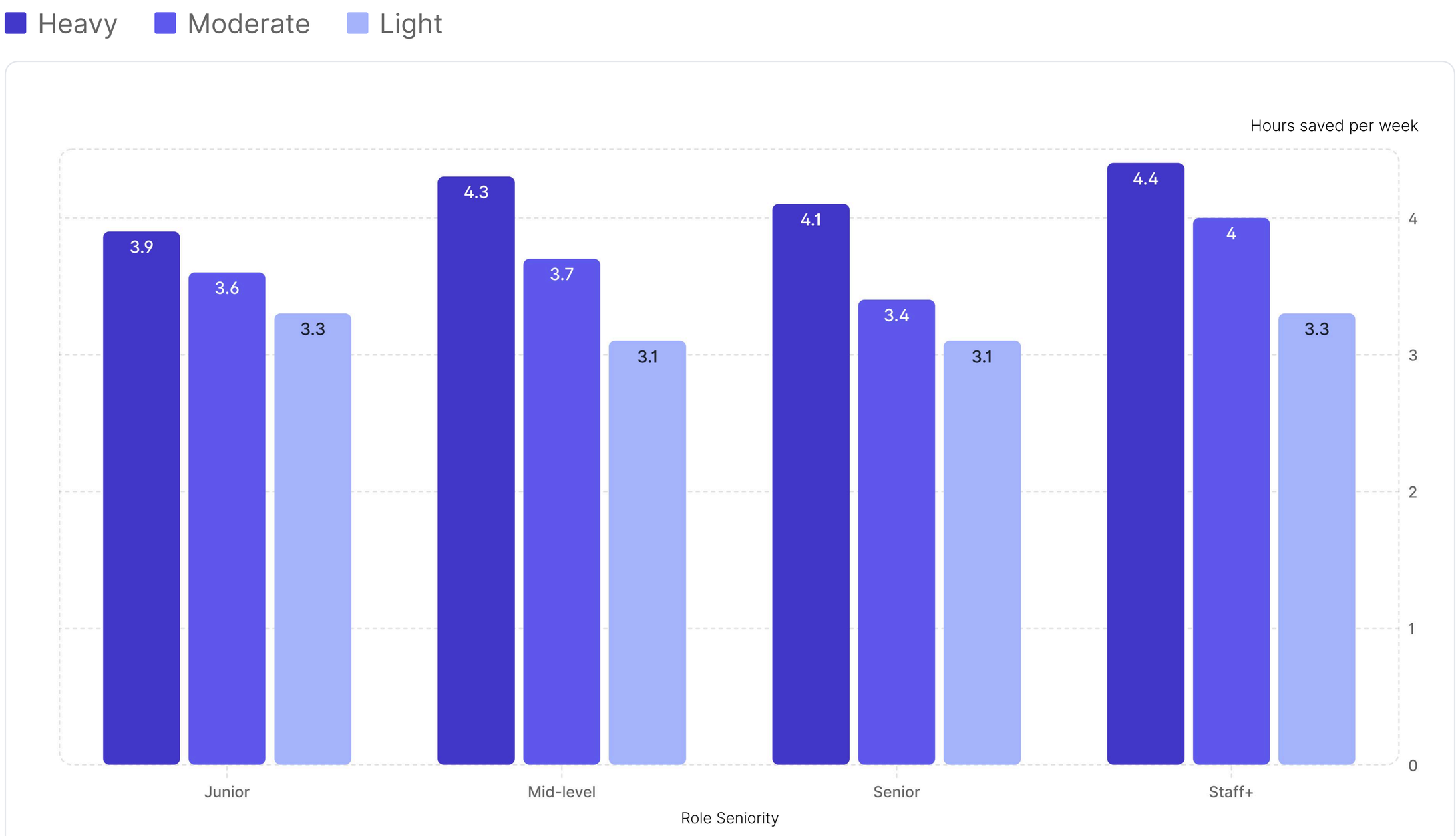
Since July of 2025, Staff+ engineers who use AI daily or more report saving 4.4 hours a week, and those using AI monthly report a 3.3 hour time savings each week.

Given that Staff+ engineers have the lowest adoption levels out of the seniority bands, this represents a large opportunity for acceleration.

Understanding the adoption barriers for more senior engineers—and remediating them—can unlock significant time savings for an organization, especially as these engineers have wide scopes of influence and can apply AI in higher-leverage ways, where the impact compounds more meaningfully.

Weekly AI time savings by AI usage level and role seniority

Sample of 20,000 developers



Traditional enterprises see higher adoption rates

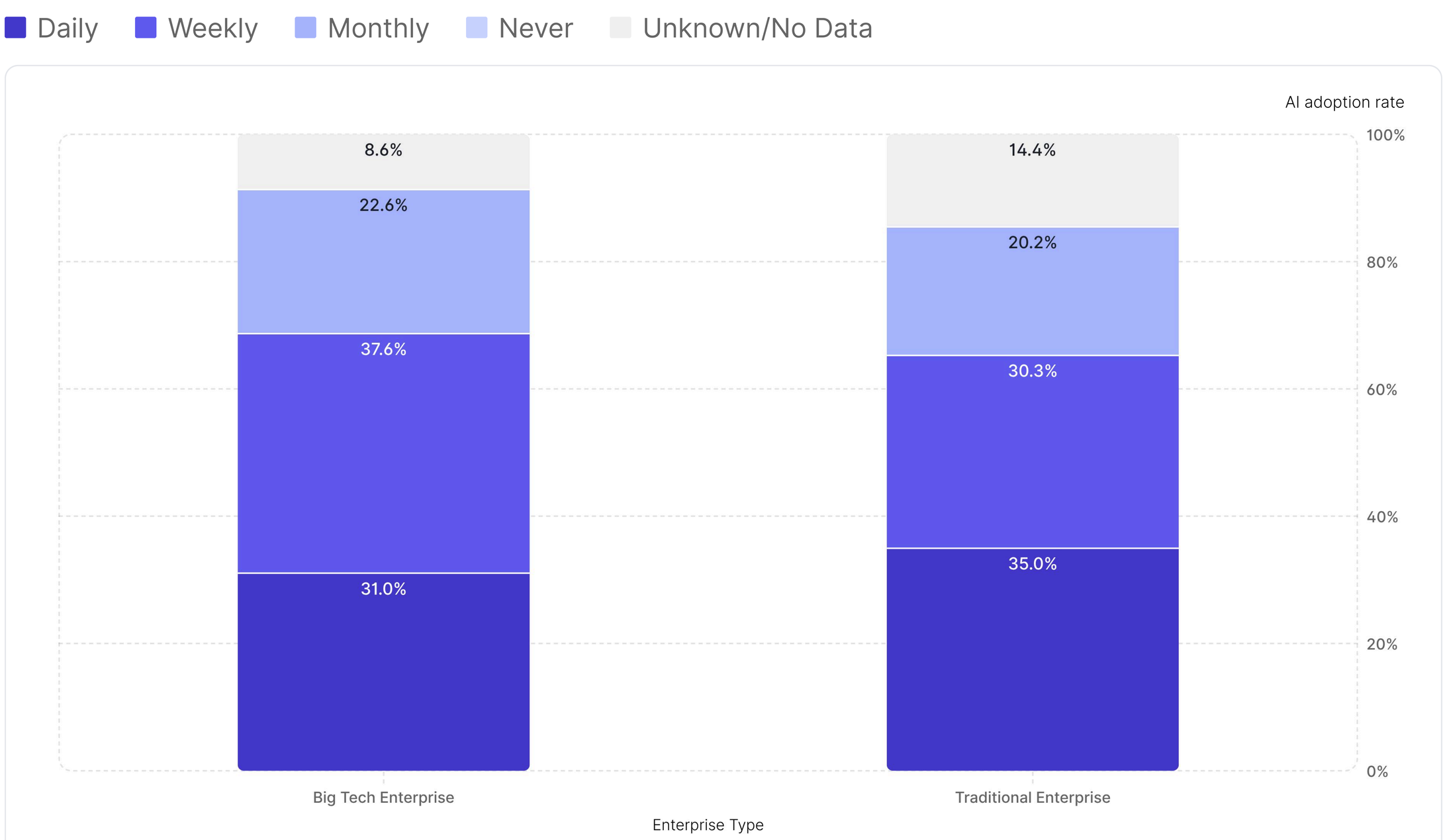
While headlines often focus on big tech companies like Google, Meta, and Microsoft, our data reveals an interesting twist: non-tech enterprises, particularly in regulated industries, are currently seeing higher AI adoption rates.

These organizations may move more slowly, but their rollouts tend to be deliberate and structured, with a strong emphasis on enablement, training, and governance.

Across all companies and industries, we see evidence that this kind of structured enablement is a key indicator of AI success. We cannot assume developers will simply "get it" with AI tools. Adoption requires a systematic, measurable approach to rollout and support in order to maximize the impact of these tools.

AI adoption rates: big tech vs traditional enterprise

Sample comparing 12 traditional to 12 big tech enterprises



Smaller companies use AI more frequently

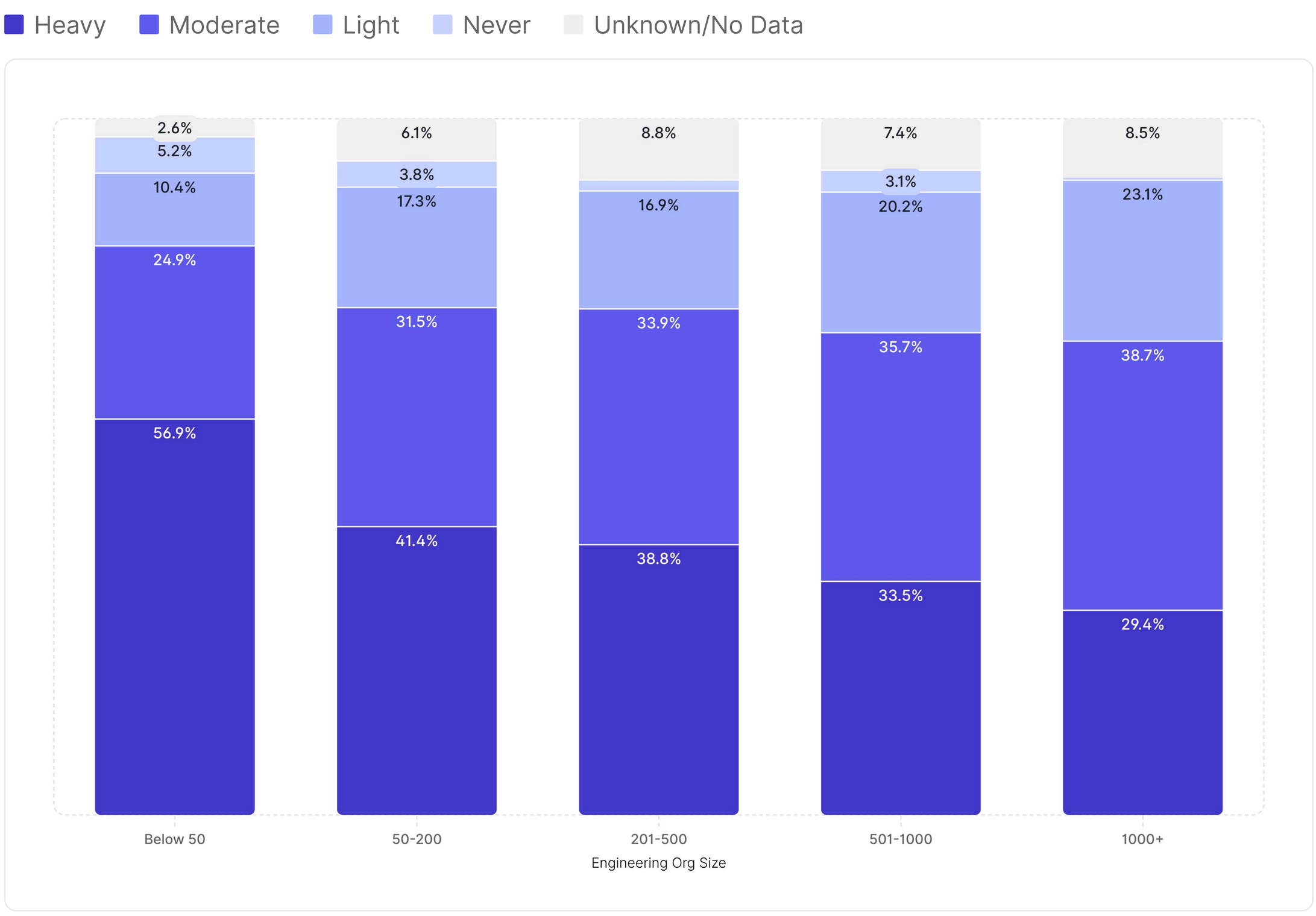
Eighty percent of developers at companies with fewer than 50 developers, and just over 70% of developers at companies with 50-200 developers, are using AI at least weekly.

This is likely not a surprising pattern, as smaller organizations typically operate with more modern tech stacks and fewer legacy systems, enabling faster adoption of AI tooling. Additionally, smaller companies often have lighter procurement processes and fewer regulations, allowing developers to experiment with and integrate AI tools more quickly.

While smaller companies may be ahead of the adoption curve, larger companies have significant opportunities for acceleration as they catch up.

AI adoption by usage level and engineering organization size

Sample of 20,200 developers from 125 companies



Enterprises lag in adoption, but are leading in AI-driven time savings

Right now there is a huge amount of opportunity for enterprises to accelerate software delivery with AI.

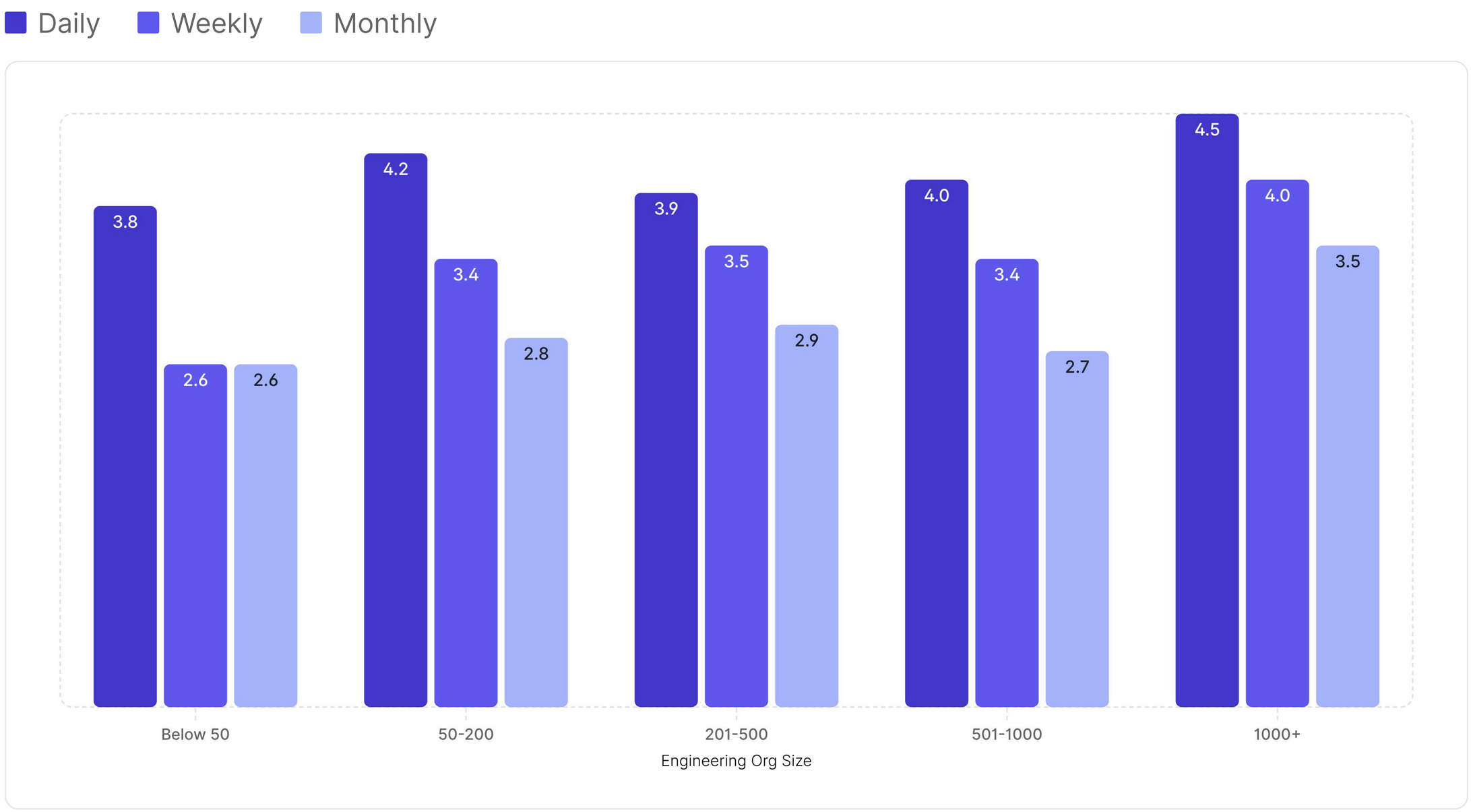
In our sample, AI users at enterprises with more than 1,000 developers saved more time per developer than their counterparts in smaller organizations, regardless of AI usage level.

With substantial developer populations, the upside of increasing AI adoption compounds quickly. It's important for these companies to continue to analyze adoption trends based on developer attributes, to see what types of developers may need more training and enablement.

For example, by focusing on enablement, Booking.com was able to increase their adoption from fewer than 10% of developers to now having close to 70% of developers using AI tools regularly. Hear more about how they scaled adoption across more than 3,000 devs here.

Weekly AI time savings (median) by AI usage level and engineering org size

Sample of 18,000 developers at 151 companies



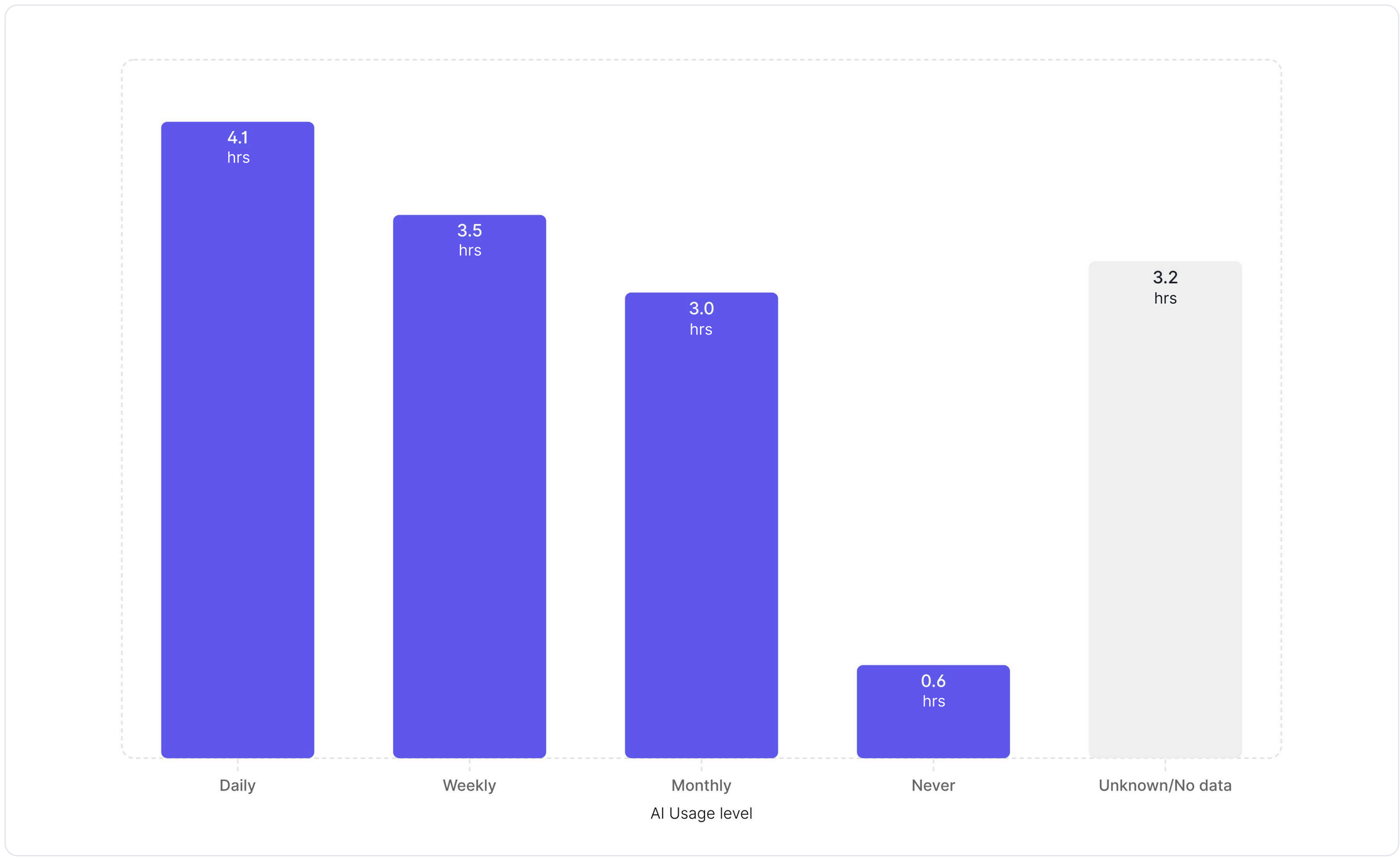
“Shadow AI” can’t be ignored, making acceptable use policies critical

Organizations should expect that developers use AI tools that they pay for out of their own pockets. This can happen for several reasons: they may have existing habits with a tool like ChatGPT, prefer a different AI tool their organization has approved, or they’re experimenting with tools outside of code authoring.

This so-called “shadow AI” has strong evidence in DX’s data. Users who have no system data from their organization’s enterprise AI tools are still reporting weekly or daily usage, and they report time savings.

Weekly AI time savings by AI usage level

Sample of 58,330 developers across 352 companies



Acceptable use policies are critical to avoid security or license breaches. Shadow AI can introduce vulnerabilities if sensitive code, personal data, or confidential information is entered into external AI tools without proper safeguards. Policies should clarify what types of data are safe to use with which AI tools.

Organizations shouldn’t be too eager to squash experimentation, especially with new tools, but rather find the right way to balance it with security and privacy requirements.

Newer AI-native tools outperform others

Looking across vendors, we see noticeable differences in outcomes. Even in side-by-side deployments, AI-native tools like agentic IDEs are associated with higher throughput compared to older or less specialized solutions.

This doesn't mean older tools aren't delivering value. These newer tools are often entering an organization where developers have gained fluency in working with AI coding assistants—an advantage that incumbent tools didn’t have. Organizations also know more about the value of training and enablement, meaning that developers are also getting more support for newer tools than they previously had.

Existing organizational performance also influences these outcomes. Some tools, like Tabnine, are more commonly found in enterprises, where PR throughput is lower overall. By contrast, smaller companies tend to adopt newer tools like agentic IDEs sooner, and also tend to ship code more frequently. However, in the results below, each tool was well represented across companies of all sizes.

These differences are also a reminder of how quickly this market is evolving. The performance landscape can shift in a matter of weeks, which is why DX recommends avoiding vendor lock-in and taking a multi-vendor approach.

Weekly PR throughput by AI adoption level and AI tool

Sample of 32,700 developers at 170 companies

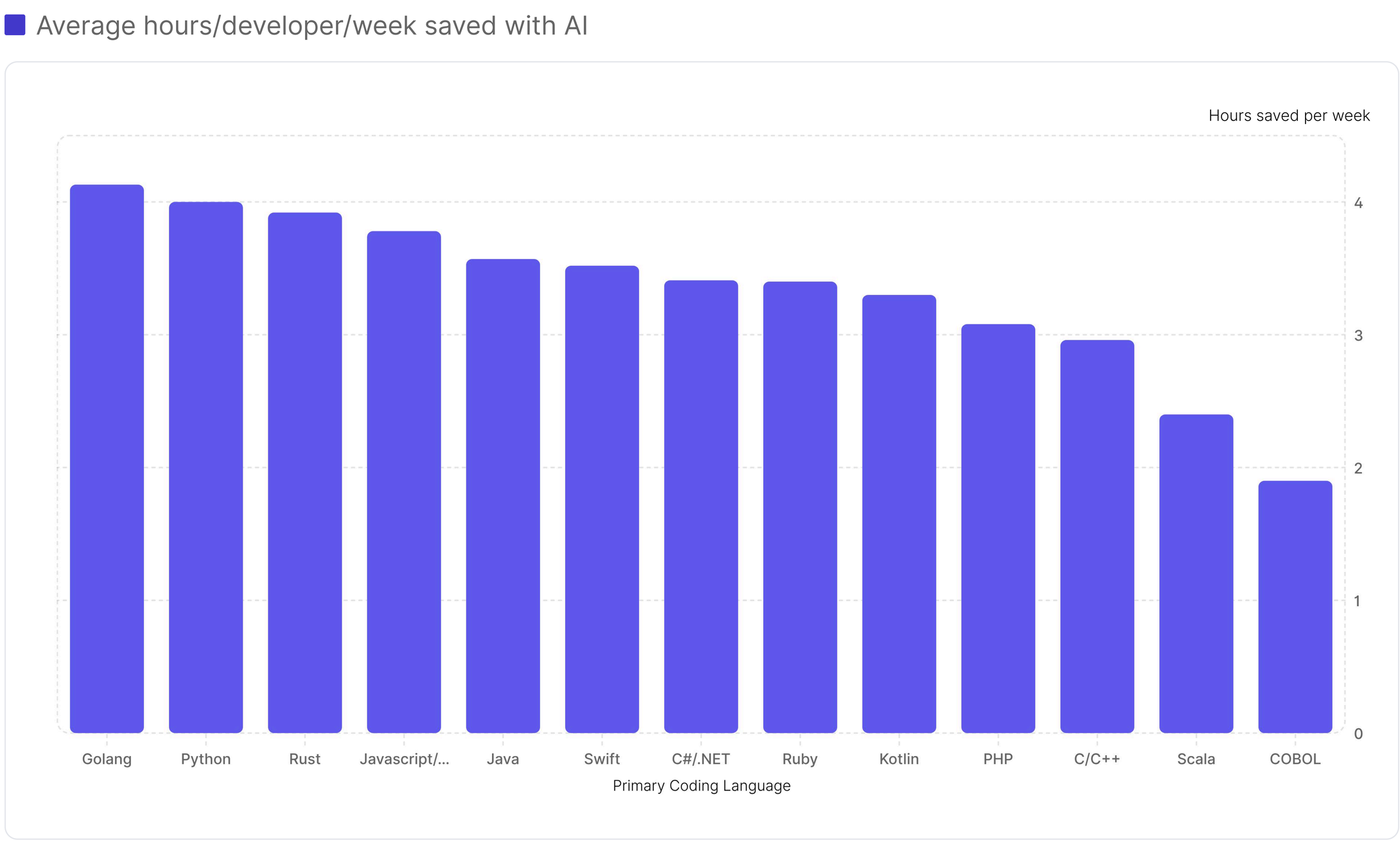
AI delivers bigger gains in modern languages

Not surprisingly, AI delivers bigger gains in modern programming languages. Because today's models are trained primarily on widely available, publicly accessible code, languages like Python, Java, and Go see the strongest time savings.

Looking ahead, the real opportunity may lie in optimizing models for specific contexts. Rather than relying on generalized, off-the-shelf models, companies are beginning to explore fine-tuned and bespoke approaches.

Weekly AI time savings by primary coding language

Sample of 23,500 developers at 43 companies

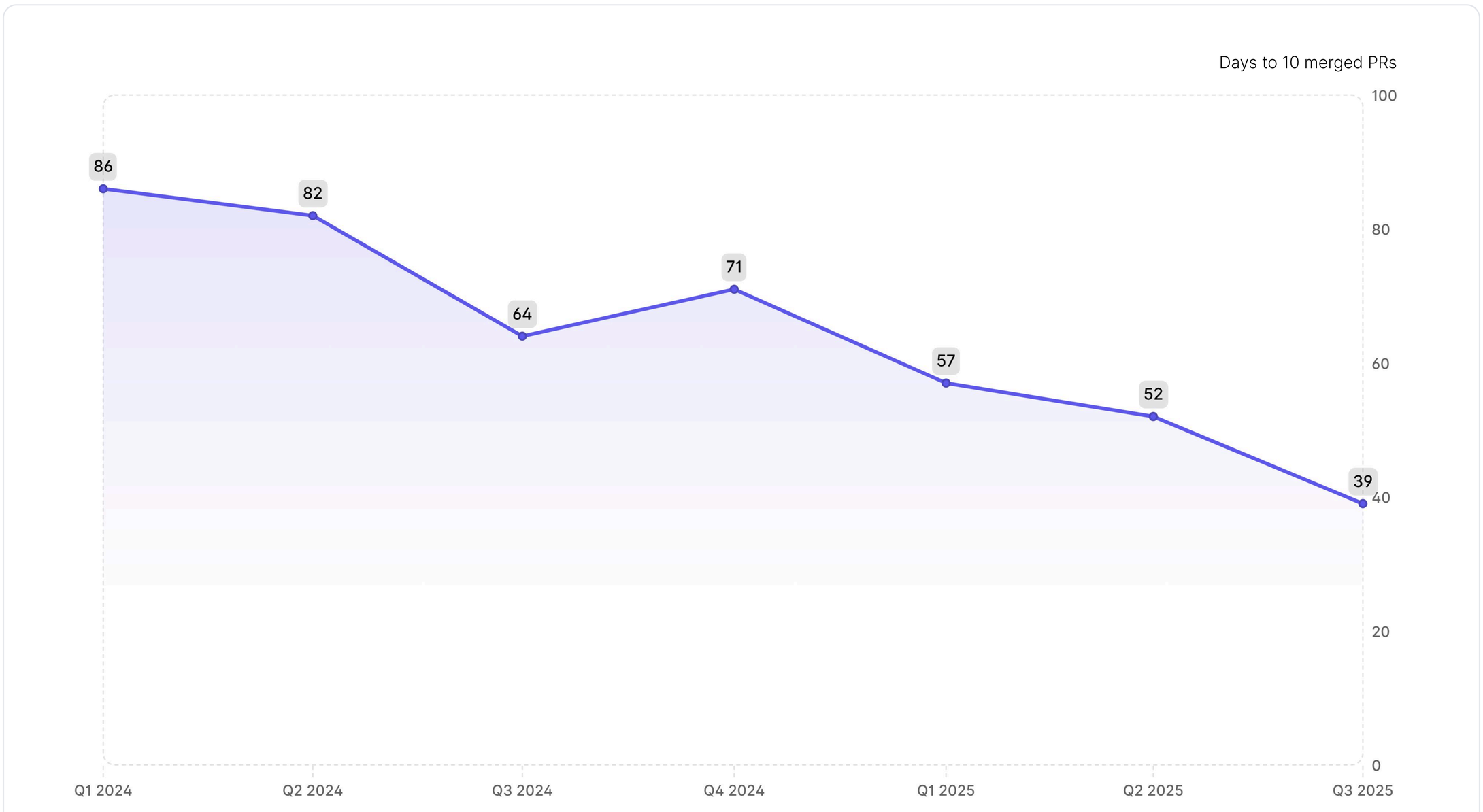


Significant impact on dev ramp-up

One of the clearest impacts we're seeing is on developer onboarding and ramp-up time.

Time to 10th PR

Sample of 14,000 developers who started new roles after January 1, 2024



Note: these numbers will change as developers continue to hit their 10th PR milestone

Since Q1 2024, time to 10th PR has consistently decreased, correlating with the rise in adoption of AI tools.

Not only does AI usage correlate with hitting this onboarding milestone faster, but AI users start with a higher PR throughput, and stay ahead of their peers who use AI less frequently. This is significant, shares Brian Houck from Microsoft, a co-author of the SPACE framework. Onboarding patterns stick with a user. “By a developer's 10th PR, I have a greater than 50% chance of predicting what their code output patterns will look like two years in the future,” says Houck based on research carried out at Microsoft.

Data from July to September 2025 at six multinational enterprises showed their onboarding time being cut in half, from 91 days with no AI usage to just 49 days with daily use.

This makes onboarding an ideal moment to introduce AI into the developer workflow. Users who start with AI start ahead, and stay ahead. By embedding AI tools into training and ramp-up materials, organizations can help new hires adopt these practices early and establish them as a core part of how work gets done.

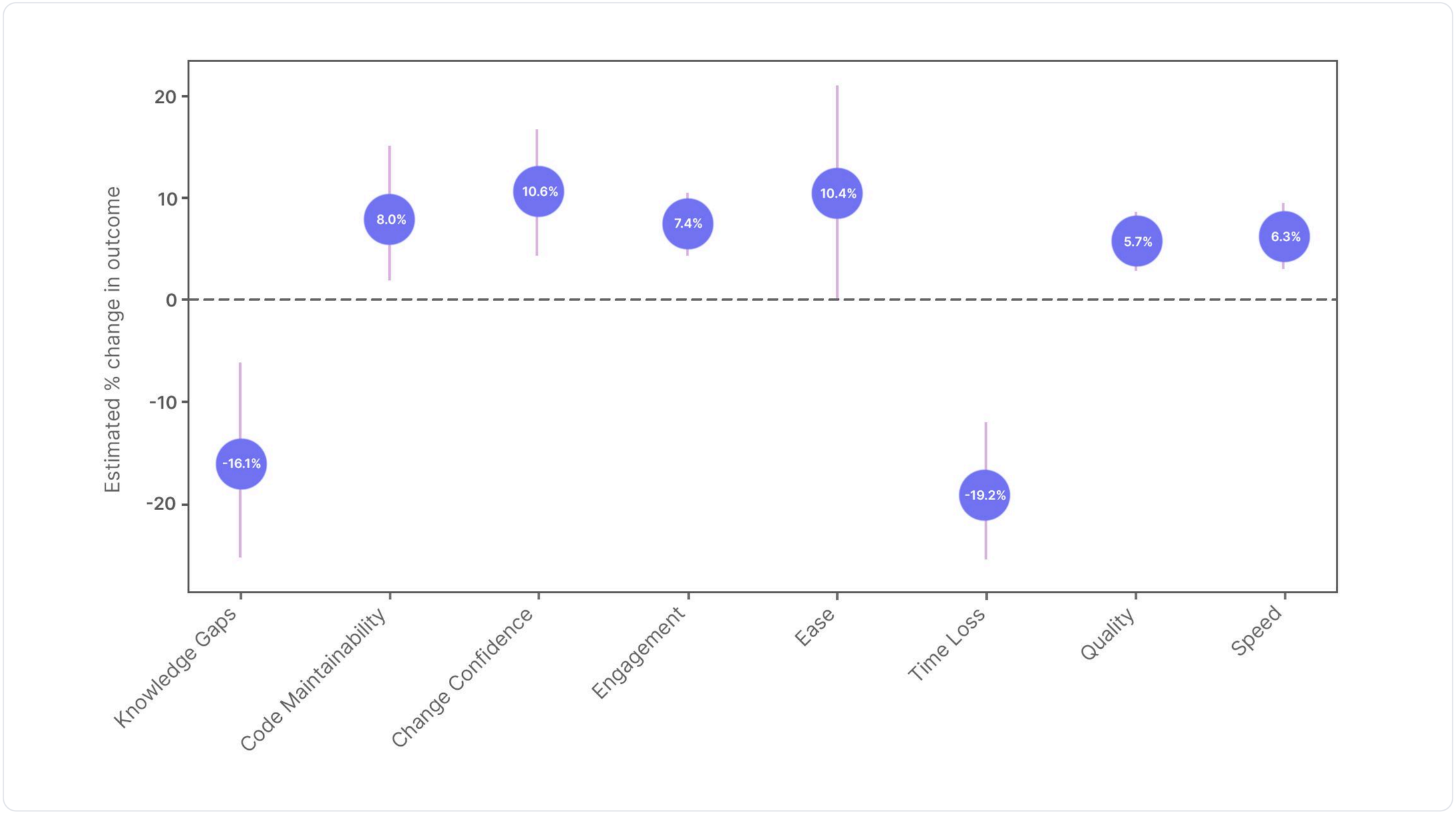
Big payoffs from structured enablement

One of the strongest signals we've observed is the impact of structured enablement on AI adoption. In one study, we measured how much support developers felt they had—through training, rollout programs, and guidance—and correlated that with outcomes.

Developers who received structured enablement reported significantly better results across multiple dimensions, from code maintainability to confidence in changes, engagement, and speed. Conversely, teams left to figure things out on their own saw higher time loss and wider knowledge gaps.

In short, structured enablement pays off in a big way. Simply handing out licenses is not enough.

If GenAI enablement increases by 25% then:



The definition of developer is expanding

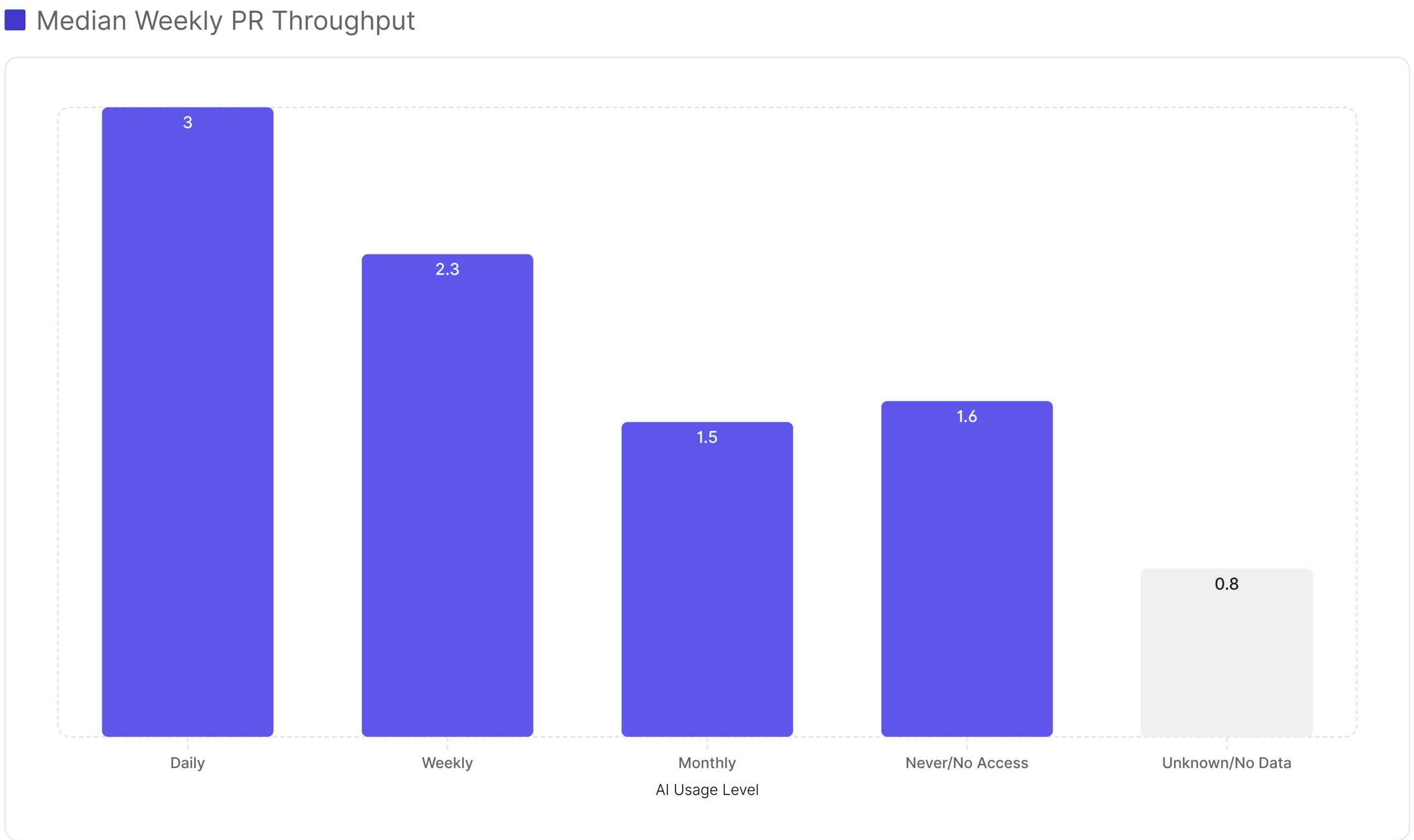
AI isn't just making engineers more efficient. It's allowing product managers, designers, and analysts to create working software, breaking down the old walls between technical and non-technical roles and fundamentally reshaping how some teams get work done.

Looking at PR data from 385 engineering managers, those using AI daily shipped twice as many PRs as those who rarely used AI, or had no access to tools. For further comparison, the median PR throughput for managers in Q1 2025, regardless of AI usage level, was 1.5 PRs/week, roughly the same as light AI users and non-users.

This increase in coding activity reflects the changing shape of the engineering manager role. As some companies are opting to flatten out their org structure, engineering managers are getting closer to the code. In interviews, engineering managers also report getting hands-on with AI tools in order to evaluate them, which can also explain the increase.

Weekly PR throughput by AI usage level

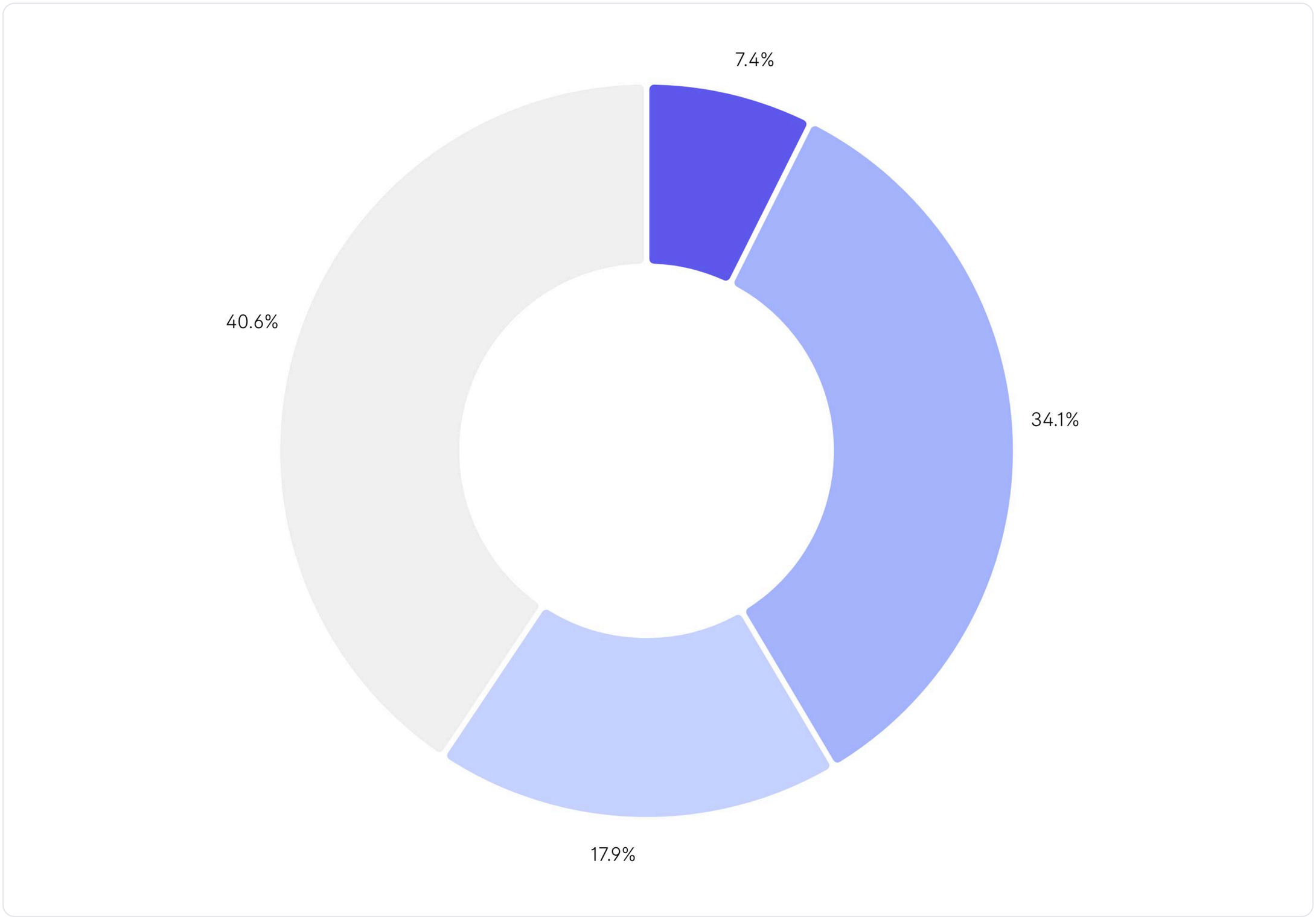
Sample of 385 engineering managers



We're also seeing designers and product managers contributing more code than before. In a sample of 245 designers and product managers at six companies, close to 60% of them use AI tools in their daily work.

AI usage levels, designers and product managers

Sample of 352 designers and product managers at 6 global enterprises



With AI making software creation more accessible to roles adjacent to development, some companies are rethinking team composition and processes. Designers and PMs are now able to prototype and validate ideas much faster with fewer engineering resources. Teams must find the right balance to ensure that engineering is brought into planning conversations at the right time.

This expanding definition of developer also highlights the need for quality standards and review processes. Code review and testing processes need to deal with AI-generated code and contributors with varying technical depth, while maintaining security and reliability standards.

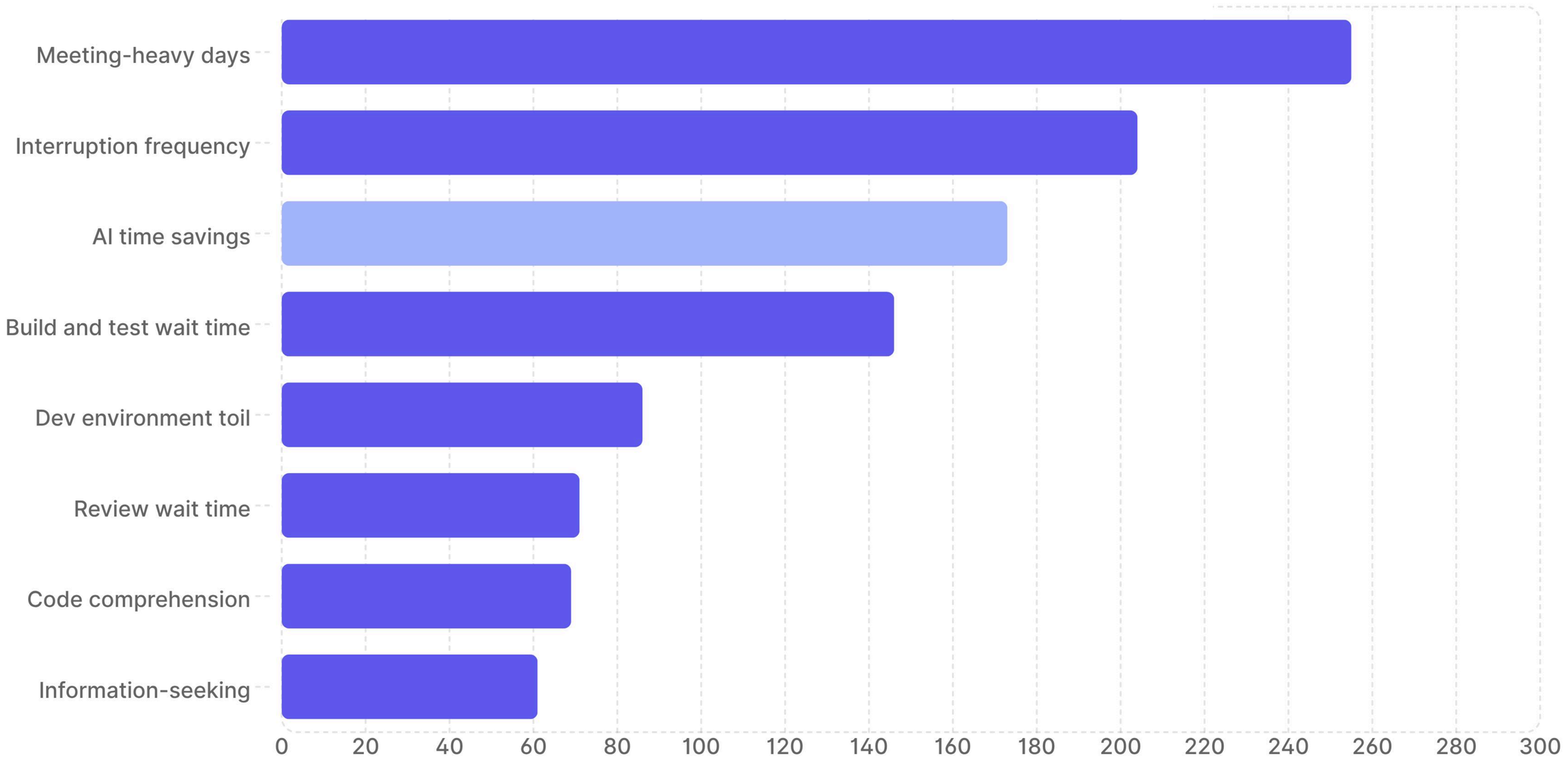
Non-AI factors still outweigh AI benefits

Although it's clear that AI is saving developers time and accelerating workflows, when we look at how much time is being lost to other tasks, those savings are eclipsed. Meeting-heavy days and interruptions remain the biggest obstacles for developers, and other aspects of developer experience—like waiting for feedback from a CI job or code review—introduce delays.

Many of these bottlenecks are things that can be addressed, at least in part, by strategically integrating AI throughout the SDLC. But organizations need to think beyond just code creation with AI to see bigger gains long-term.

Annualized distribution of developer time by workflow phase

Self-reported data from sample of 134,085 engineers



> To unlock developer productivity at scale, organizations must continue to identify and address bottlenecks and evaluate investments in workflow improvements alongside investments in AI.

Measuring engineering productivity in the AI era

Continuous measurement is a key piece of a successful AI strategy. In this section, we'll look at how to measure the impact of AI tools inside your own organization.

AI Measurement Framework

To address the question of how to measure AI's impact on productivity, we collaborated with researchers, leading AI vendors, and customers to develop the AI Measurement Framework—a research-based set of metrics for tracking utilization, impact, and ROI of AI-assisted engineering.

The framework is vendor-agnostic and designed to be practical. Any company can begin using these metrics—or even a subset of them—to establish a clearer, more consistent picture of how AI is influencing developer productivity and ROI.

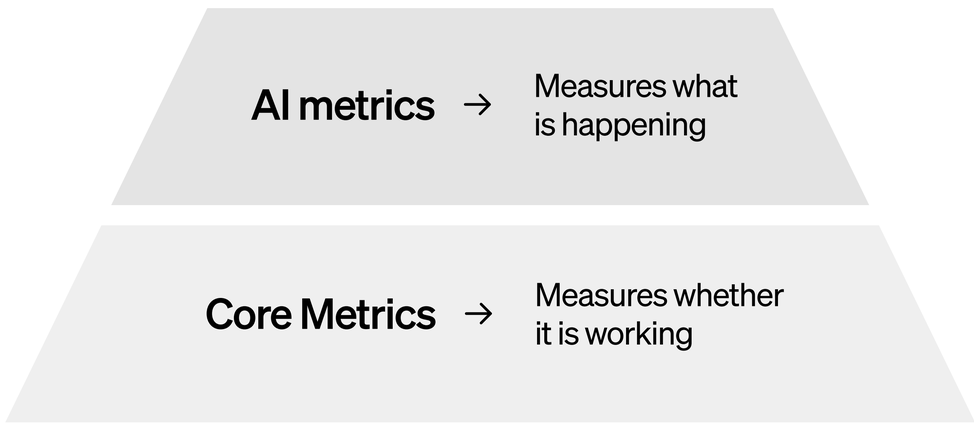
Utilization How much are developers adopting and utilizing AI tools?	**Impact** How is AI impacting engineering productivity?	**Cost** Is our AI spend and return on investment optimal?

• AI tool usage (DAUs/WAUs)
• Percentage of PRs that are AI-assisted
• Percentage of committed code that is AI-generated
• Tasks assigned to agents *
	• AI-driven time savings (dev hours/ week)
	• Developer satisfaction
	• DX Core 4 metrics, including:
	  • PR throughput
	  • Perceived rate of delivery
	  • Developer Experience Index (DXI)
	  • Code maintainability
	  • Change confidence
	  • Change fail percentage
	• Human-equivalent hours (HEH) of work completed by agents *
		• AI spend (both total and per developer)
		• Net time gain per developer (time savings - AI spend)
		• Agent hourly rate (HEH / AI spend) *

* Metrics for autonomous AI agents

Core metrics and benchmarks are as important as ever

AI introduces new dimensions to track, like adoption and cost, but it doesn't fundamentally change how we measure developer productivity. The foundations remain the same. Core measures provide the baseline against which AI's impact can be judged. In other words, AI metrics tell us what's happening, but core metrics confirm whether it's actually driving improvement.



How top companies measure AI's impact in engineering

Companies like GitHub, Google, and Dropbox are examples of blending foundational engineering metrics, like those found in the DX Core 4, with AI-specific metrics to track how and where AI is being used by developers.

The AI Measurement Framework is based on our experiences working with companies like Dropbox and Booking.com, along with the data from 425+ companies that you've seen in this report. If your company is looking for a starting point to better understand how AI is impacting your organization, the AI Measurement Framework is a recommended place to start.

GitHub	Google	Dropbox
• AI time savings • PR throughput • GitHub Copilot CSAT • Change Failure Rate • Engaged users • Innovation rate: feature expenses/all engineering expenses • Developer experience, e.g. code maintainability, focus time, overall satisfaction	• % of code by AI (drafted & submitted) & Acceptance rate • Weekly active users (WAU) • Usage intensity • Change throughput • Active coding time, investigation time, & review time per change • Change cycle time • User sentiment and trust in AI • Perceived productivity • Developer friction	• Daily active users (DAU) / weekly active users (WAU) for AI tools • AI tool CSAT, PR throughput (comparison by AI usage) • Time saved per engineer per week • Change fail percentage • AI spend total and per developer
Microsoft	**monzo**	**ATLASSIAN**
• Usage of AI tools and agentic workflows • PR cycle time (created to merged) • Developer satisfaction • Change failure rate • Bad developer days • Idea-to-customer (innovation velocity)	• DAU/WAU for AI tools • PR throughout • Time savings per developer per week • Change failure rate • Deployment frequency • CSAT for AI tools • Developer experience	• DAU/WAU across AI tools (including multiples) • Acceptance rates of AI-generated suggestions • % of PRs assisted by AI • Spend and throughput of issues assisted by AI • Qualitative developer satisfaction
adyen	**Booking.com**	**grammarly**
• PR throughput • Change failure rate • DAU/WAU for AI tools • AI tool CSAT • Developer experience, esp code maintainability, understanding complex code, knowledge gaps • AI time savings	• DAU and WAU for AI tools • Time saved per dev • % of PRs that are AI-assisted • PR throughput • Developer CSAT • Change failure rate	• Cycle time • DAU/WAU (per tool) • Acceptance rate • Diffs per engineer • % of time spent on feature development (innovation ratio) • AI tool CSAT • AI spend (total and per developer)

DX × THE PRAGMATIC ENGINEER

Summary

AI has been fast reshaping how software is built. The data shows real, measurable gains—time savings, faster ramp-up, higher throughput, and broader participation in coding across roles. Yet the impact is uneven, and the picture is more complex than the headlines suggest.

1. Avoid vendor lock-in: AI capabilities are evolving quickly. The smartest organizations are adopting measurement frameworks and practices that remain vendor agnostic, ensuring flexibility as the ecosystem shifts.
2. Expand the definition of "developer": AI is enabling engineering managers, designers, and product managers to contribute code at unprecedented levels. This broadens the definition of who participates in software creation, and also means that organizations may need to rethink collaboration processes as well as testing and automation in order to accommodate more AI-generated code, and more participation from people who don't have a background as a developer.
3. Measure adoption, impact, and cost: Utilization tells us if tools are being used, core productivity metrics show whether they're making a difference, and cost ensures we're getting the return we expect. All three dimensions are essential.
4. AI isn't a silver bullet: While AI is unlocking real time savings and throughput gains, non-AI bottlenecks like meetings, review delays, and CI wait times continue to outweigh its benefits. Productivity at scale requires tackling both.

We'll continue sharing what we're seeing in DX data quarterly as adoption evolves. We encourage others to share their experiences as well, so together we can build a clearer understanding of AI's impact on engineering and where it is heading.

Methodologies

This report looks at data from 435 companies, across a combined sample of over 135,000 developers, from the time period of July 1, 2025 to October 16, 2025. Specific sample sizes are noted alongside the data visualisations in this report. Companies range in size from <50 to over 10,000 employees in both technical and traditional sectors, with headquarters in North America, Latin America, Europe, and Asia-Pacific.

AI usage levels

AI usage tags are derived from system data collected from AI coding assistants like GitHub Copilot, Cursor, and Claude Code. For a full list of DX's data connectors, refer to our documentation.

- Usage levels are determined by looking at the previous 4 weeks of AI usage
- If a user has access to more than one AI code assistant, their usage is aggregated across all tools
- If a user has no evidence of AI tool usage, they are classified as "Unknown/No data", except for users who have self-reported that they do not use AI tools. Those users have been classified as "Never"

Time savings

DX collects information about AI time savings through self-reported data in a periodic survey using the following closed-option question.

On average, how much time do you save per week thanks to AI code assistants?

- ◯ Less than 1 hour per week
- ◯ 1-2 hours per week
- ◯ 2-4 hours per week
- ◯ 4-6 hours per week
- ◯ 6-8 hours per week
- ◯ 8+ hours per week
- ◯ I don't know / Not applicable

Point-in-time Snapshot

We do expect this data to change rapidly as organizations undergo significant transformation to adopt AI technologies.

Acknowledgements

Special thanks to Andrew Boyagi, Gergely Orosz, Jennifer Riggins, Justin Reock, Michael Carr, and Nathen Harvey for their contributions and support on this report.

Thank you also to everyone who contributed to our report covering how 18 top companies measure the impact of AI in engineering (page 22): Antoine Daignan, Brad Vandehey, Brian Houck, Bruno Passos, Chris Chandler, Ciera Jaspan, Collin Green, Dan McKenzie, Fabien Deshayes, Frank Fodera, Kazuaki Okumura, Kelly Anne Pipe, Maryna Veremenko, Meirav Feiler, Micaela Stump, Paul Giglio, Ramesh Periyathambi, Raj Patel, and Shelly Stuart.

About DX

DX is a developer intelligence platform designed to help engineering organizations measure, understand, and improve developer productivity. It gives engineering leaders and platform teams the data and insights they need to take the right actions to drive higher ROI per developer. DX serves hundreds of the world's most iconic companies, including Dropbox, Block, Pinterest, and BNY.

getdx.com

Engineering intelligence

Learn more at getdx.com

Copyright © 2025