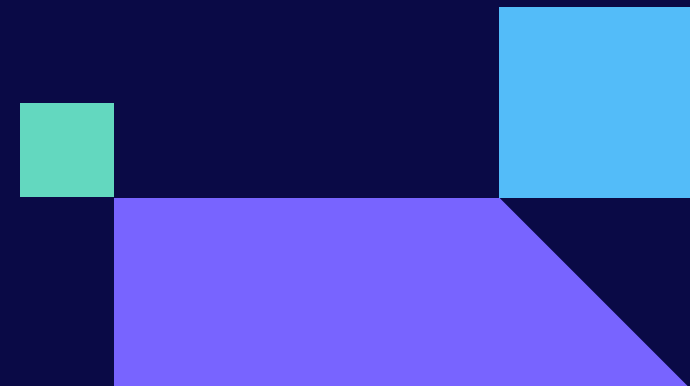




The **LeadDev** Engineering Team Performance Report 2023



in partnership with



Foreword

Books have been written about it, consultants have designed PowerPoint decks that say they can solve it for you, developers have built and rebuilt dashboards, and academics have developed complex frameworks, yet we still don't have an agreed-upon way to measure and optimize software engineering team performance.

We know that teams build software, not individuals. But we also know that every organization craves a way to measure the performance of those teams so that they can make better business decisions. Individuals also want to progress in their careers, and finding an objective way to measure that performance remains a tough nut to crack for engineering managers all over the world.

Despite the emergence of DORA, SPACE, and many more frameworks, there is still no single way to measure developer team performance, adding to the cognitive load of engineering leaders tasked with quantifying the output of their team against wider company goals. That's because each organization will differ in what it wants to optimize this performance for. What we do know is that any effective measure of team performance must account for both raw output data and developer team health.

With this report, we set out to establish what is best practice for engineering managers, better understand which measurement methods are rising and which are falling, and identify the organizations that are managing engineering performance the best.

This summer, we spoke to dozens of engineering managers from the likes of Etsy, Shopify, and Stripe, and surveyed more than 600 engineering managers to find out how most organizations measure their team performance, and the biggest challenges they face in doing so. We hope this helps you identify where you are in your team performance journey, and perhaps spark some ideas on how to better measure their performance.

Now go build epic teams!

Scott Carey

Editor in Chief, LeadDev



A note from Swarmia

Our mission at Swarmia is to empower modern software teams with the insights they need to do their best work. That's why it took us exactly one Zoom meeting with our friends at LeadDev to decide we wanted to support this research project. We saw the project as a unique opportunity to learn more about the challenges and opportunities engineering leaders are experiencing when it comes to measuring and improving their teams' performance.

To us, the findings of this study lament what we had already been hearing from our customers: leading a high-performing engineering team requires a bullish commitment to continuous improvement and a holistic approach to measurement.

Indeed, the highest-performing software teams today have forgone the idea of “one metric to rule them all” and instead, have become masters of measuring their efforts across three key areas:

- **Business outcomes:** focusing on work with the highest perceived business impact
- **Developer productivity:** systematically eliminating what gets in the way of delivery
- **Developer experience:** minimizing the negative impact of wait times and interruptions

And while tooling is only one part of the puzzle, sooner or later, you'll find that added visibility, healthy insights, and automation might come in handy. When that time comes, you can find us at swarmia.com.



At a glance

1 Cycle time was ranked as the most useful engineering productivity metric

2/3 leaders say finding the right metrics is a key challenge to measuring team performance

Half of respondents who are familiar with DORA metrics and SPACE find them to be effective or very effective measurement frameworks

1/5 engineering leaders found team resistance to measurement to be a key challenge

7% of respondents don't set engineering team goals at all

Goal setting

Goals

“Why are we here?” is not an existential question for engineers. Goals and objectives provide the structure within which engineering teams work and measure their success. You can’t measure team performance unless you know what goal you are trying to achieve.

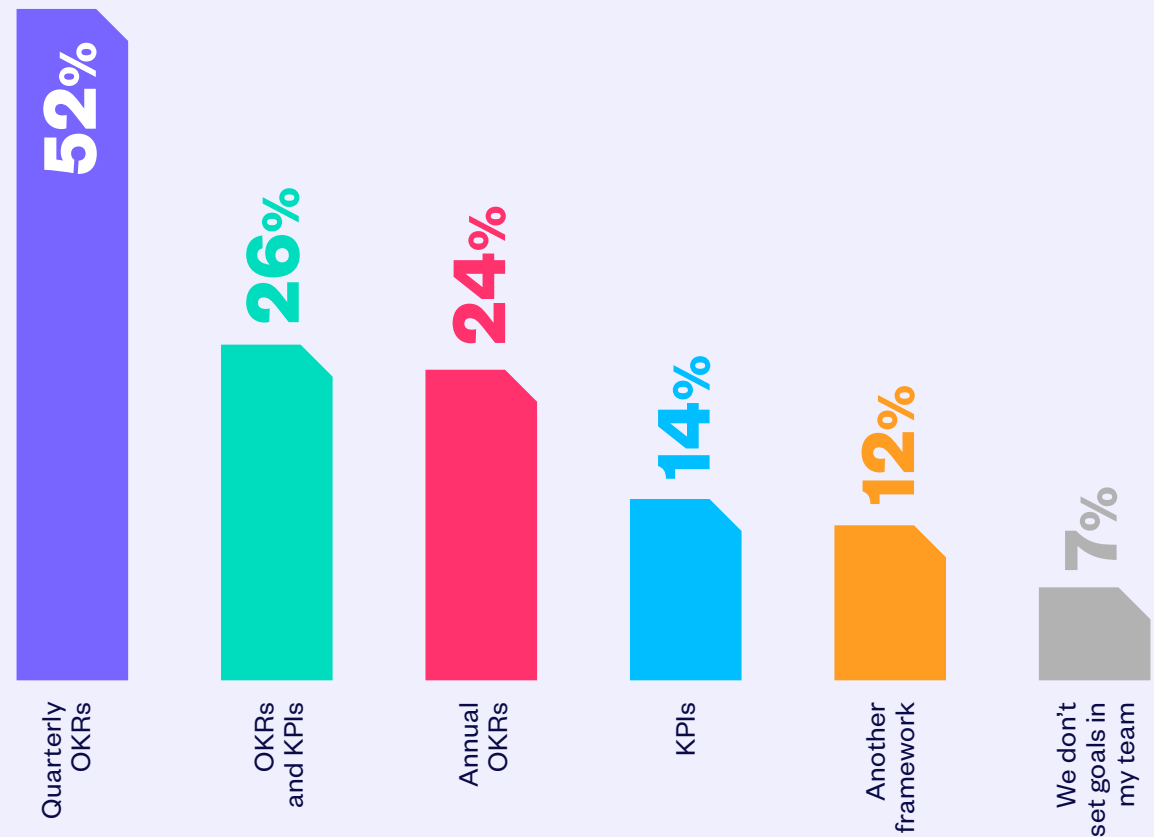
While company-wide objectives and key results (OKRs) appear to be the broad consensus way to define and communicate business goals, how often you do this is vital. As we have seen clearly this year, things are liable to change, and fast. So it’s no surprise to see that half of respondents set quarterly OKRs, while just under a quarter wait to do this exercise on an annual basis.

The other key framework for setting goals was key performance indicators (KPIs), but very few organizations rely on them alone, at just 14%.

Another 26% use a combination of OKRs and KPIs, making it the second most popular approach.

The most popular process for setting goals are quarterly OKRs

Q: What’s your process for setting goals within your team?



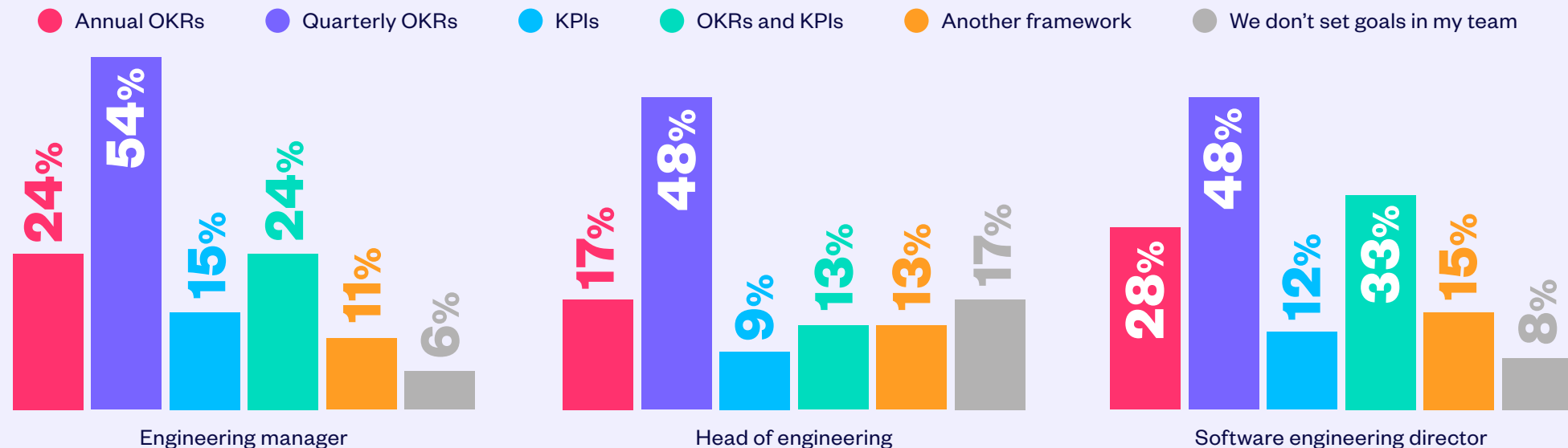
There is also a world beyond OKRs and KPIs. Some organizations or teams might use the Goal Reality Options Will (GROW) framework, or define Key Results Areas (KRAs). Others might develop their own hybrid or less formal approaches. Nevertheless, just 12% of respondents cited “another framework” for setting goals. That leaves only 7% of respondents who don’t set goals in their team at all.

Equally intriguing was an appreciable drop in enthusiasm from North American respondents for quarterly OKRs – where 48% chose this process, compared to 55% amongst their EMEA and Latin America counterparts.

The popularity of quarterly OKRs was also reinforced the higher up the career ladder you go. Software engineering directors were significantly more likely than most to look at a mix of OKRs and KPIs, with 33% of that group using this approach. This was substantially higher than the 13% of heads of engineering, who used the combined approach. The proportion of heads of engineering who said their team worked without set goals rose to 17%.

Quarterly OKRs are the most popular for all job functions, but software engineering directors are significantly more likely than others to use OKRs and KPIs

Q: What’s your process for setting goals with your team?



Business objectives

OKRs, KPIs, or a combination thereof can be used to guide the team in helping the business meet its strategic objectives. But doing this successfully assumes those goals are communicated throughout the organization.

It's clear that the senior leadership team is largely responsible for communicating business objectives, with 83% of respondents naming this as the dominant way to share business objectives across an organization.

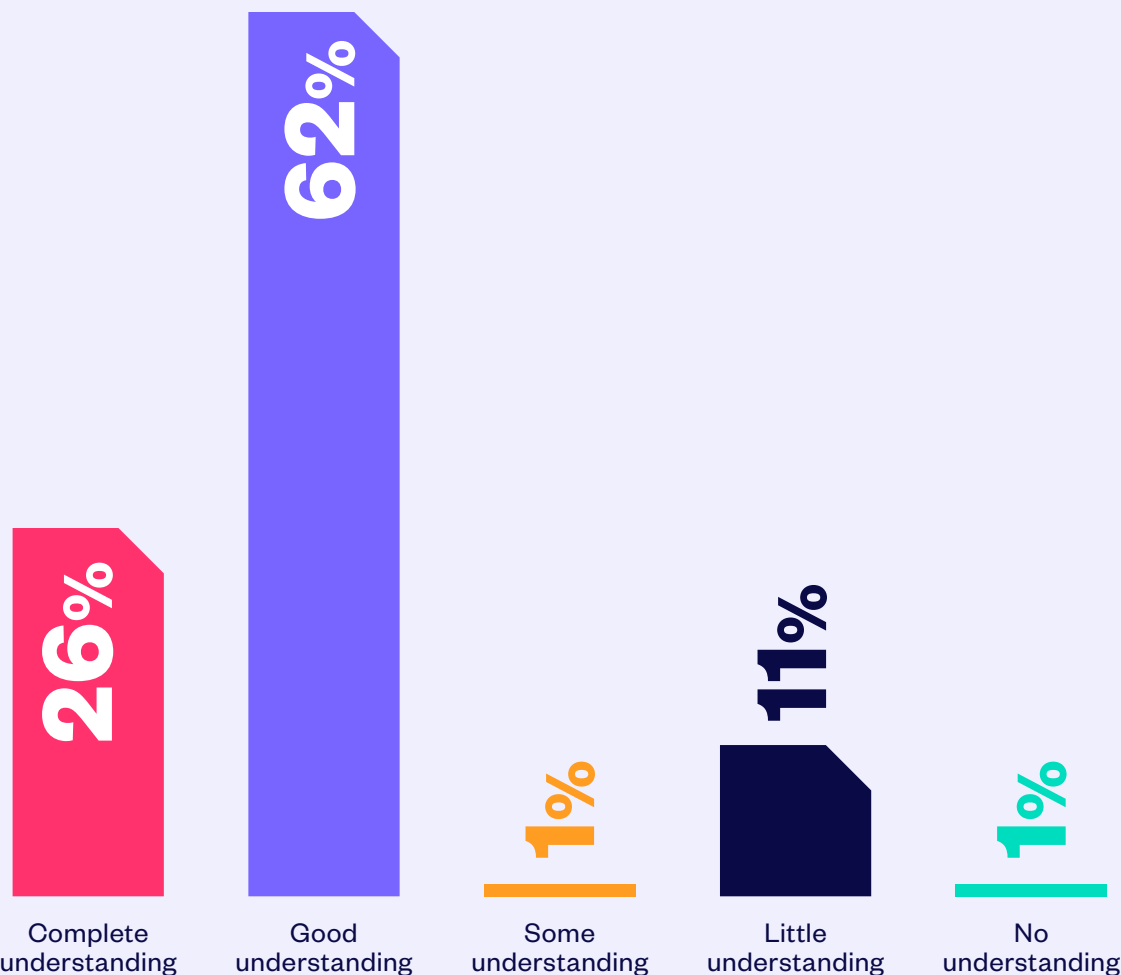
However, over half of engineering leaders said that business objectives were communicated by product managers. Then, just under one in ten said business objectives were communicated via service level objectives (SLOs).

Whoever is driving this initiative, only 26% of engineering managers are getting a complete understanding of their organization's strategic business goals, with almost two thirds saying they had a good understanding. This may seem high, but is a "good understanding" enough to guarantee success in steering engineering performance to meet those goals?

That leaves a worrying 12% of respondents who are still in the dark about their company's strategy.

88% of respondents have a good or complete understanding of their company's strategic business goals

Q: How well do you understand your company's strategic business goals?



Measuring progress

No matter what the company's broader strategic goals are, engineering leaders need a way to measure how their teams contribute to them. This has historically proved to be difficult, as the complex web of inputs that make up software engineering do not neatly map to measurable outcomes, leaving each manager with the task of putting the jigsaw together.

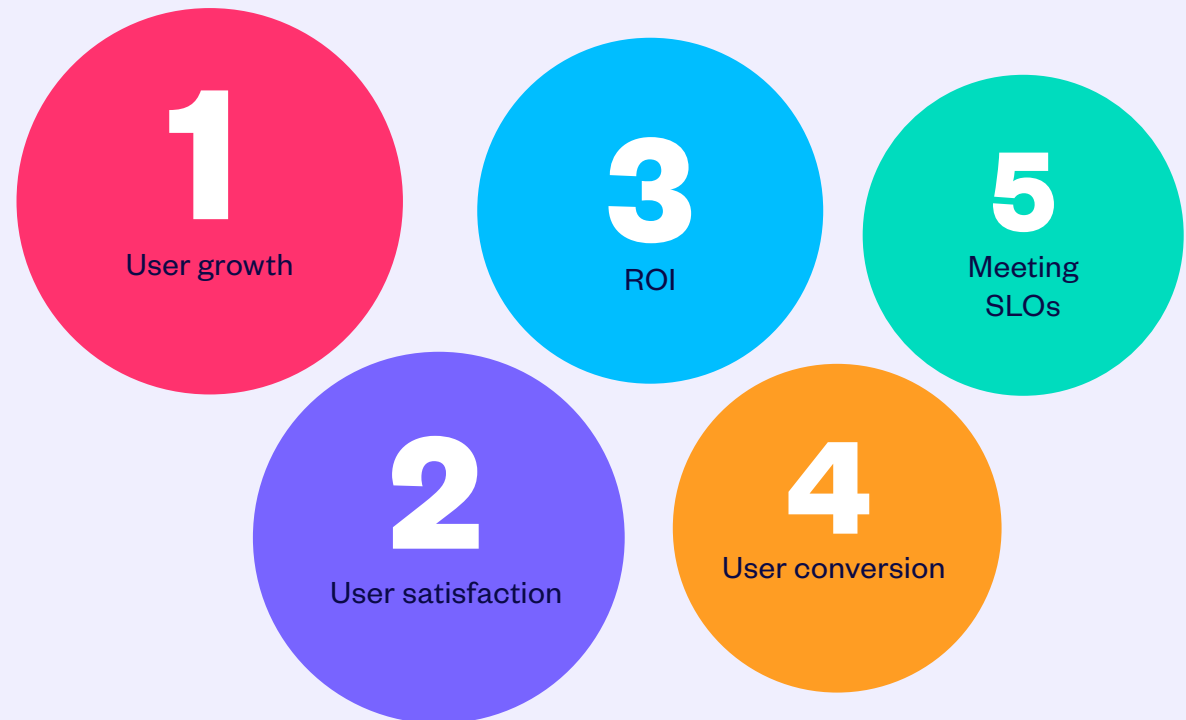
The first place many look to turn is their users. The highest-ranked metric for linking engineering performance to broader business goals was user growth, closely followed by user satisfaction. Next came return on investment (ROI), with meeting SLOs ranked fifth.

But those aren't the only key metrics teams are watching, with some choosing to measure revenue and annual recurring revenue churn, retention rates, Net Promoter Scores (NPS), and social media excitement and sentiment as indicators of the impact of their work.

In terms of how they measure this impact, over half of engineering teams use quarterly reviews, with slightly fewer setting their own goals for managers to review. Code quality was tracked by 40% and just 6% used pre-mortems. Measures such as bug quotas or sprint points carried over were named by less than one in five. This suggests leaders are taking a more holistic view of their team's impact, rather than pulling together a grab bag of datapoints.

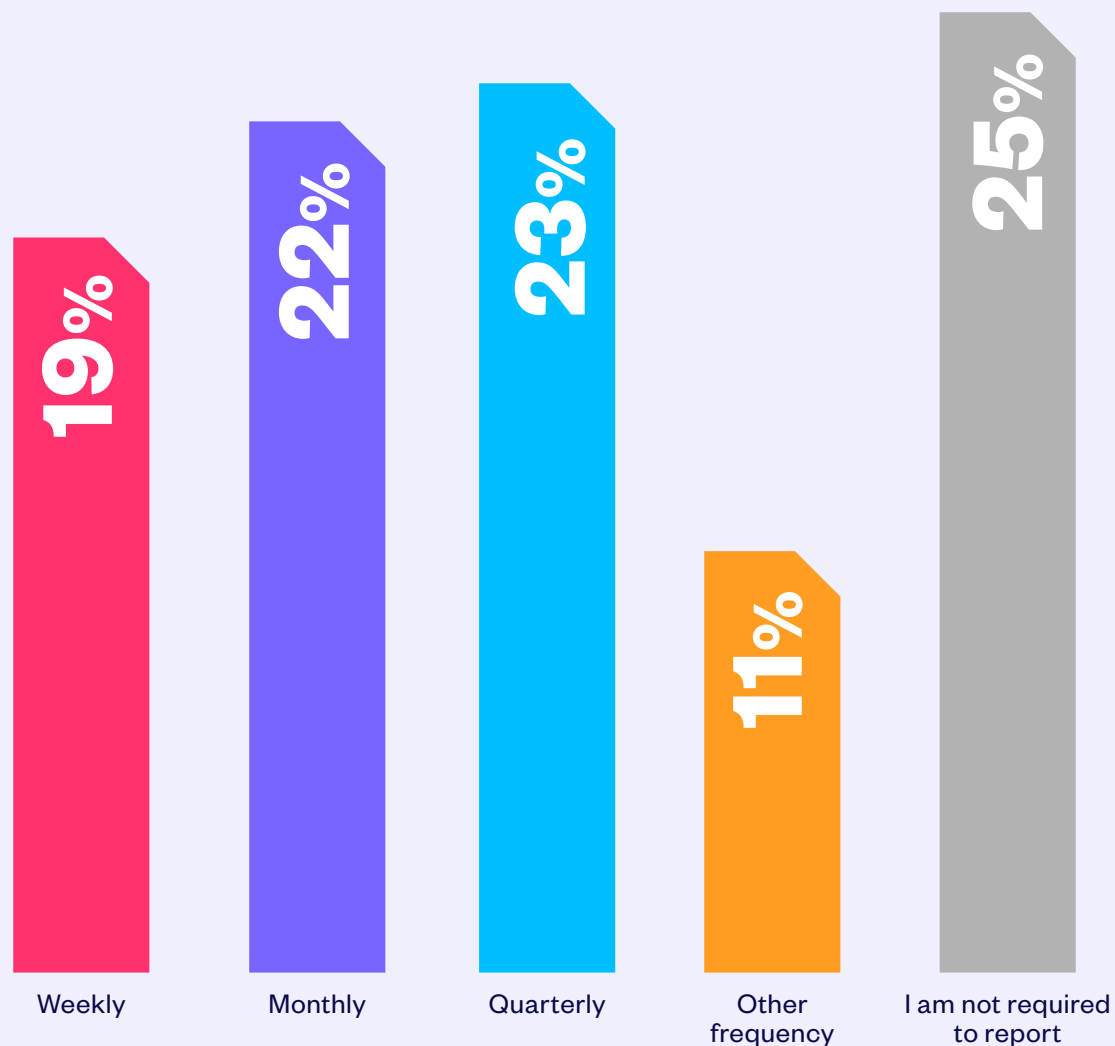
User growth and user satisfaction were ranked as the two most important metrics that are employed by engineering organizations to measure impact against strategic business goals

Q: Which of the following metrics are used in your engineering organization to measure impact against strategic business goals?



Engineering managers report on team performance at a range of frequencies: 19% report weekly, 22% report monthly, and 23% report quarterly

Q: How often are you required to report on team performance?



There were some differences by job title of note here. Both engineering managers and software engineering directors were most likely to opt for quarterly reviews, with teams setting their own goals, in line with the overall figures. Just 22% of heads of engineering cited quarterly reviews. However, engineering heads were far more likely to focus on code quality and number of customer complaints – both of which were named by 52% of this group.

While quarterly reviews were the most popular mechanism overall for tracking how teams are meeting strategic goals, it is clear that performance is still being monitored more frequently, at least tactically. One in five respondents report on performance weekly, and just a few more report monthly. Software engineering directors were the most likely cohort to report quarterly.

When it comes to reporting tools, there is no shortage of choice for engineering managers. Despite a wave of new entrants to this market, there are still two clear standouts: GitLab (66%) and Jira (65%), with the humble dashboard next at 38%. Other reporting tools weren't as popular, with none exceeding a 6% use rate, while a third cited other tools, which included Google Sheets.

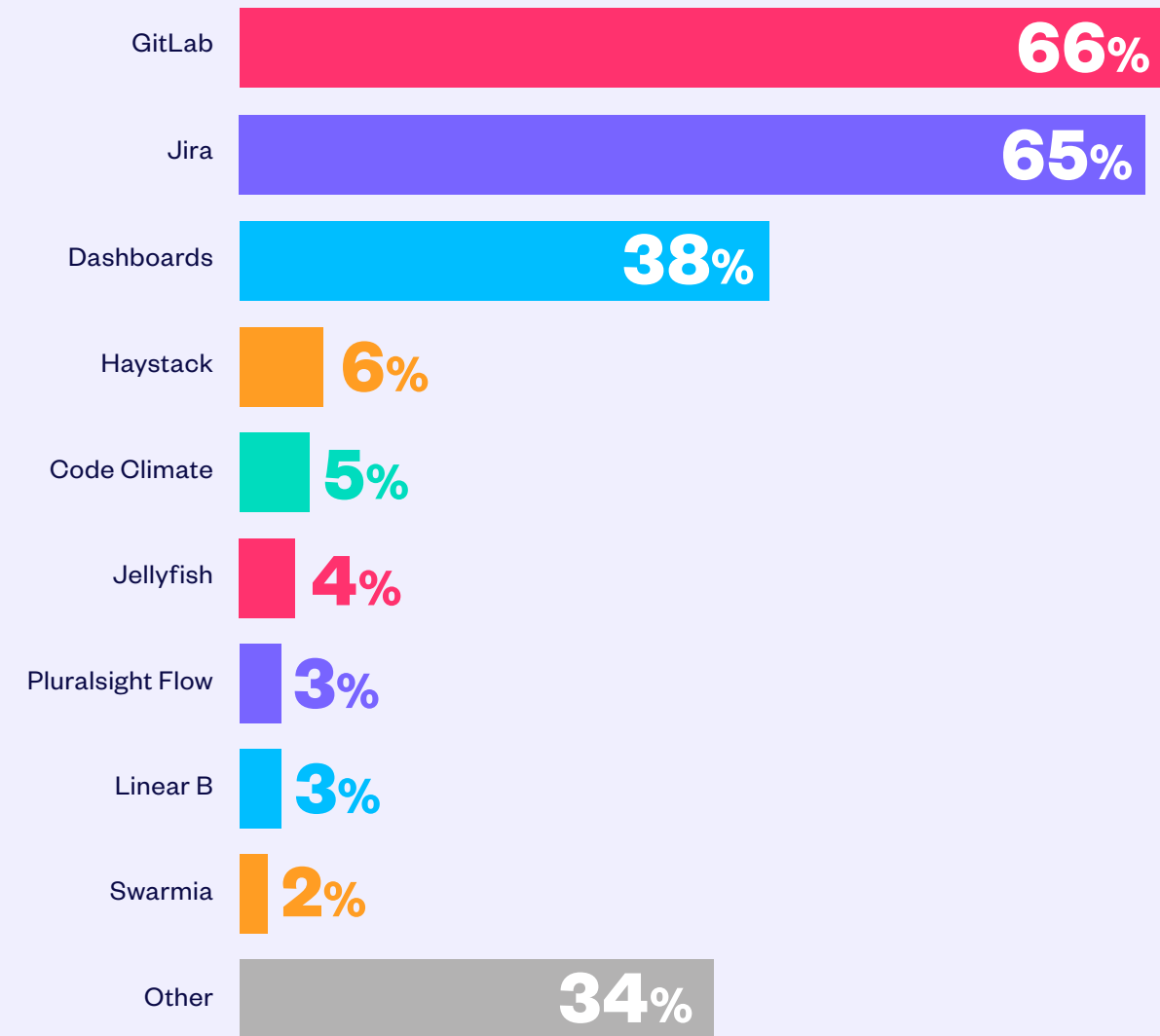
A bigger challenge is choosing the precise metrics to measure, with 67% of respondents struggling here, while 41% found a lack of tooling or dashboards to be a challenge. Interestingly, only a fifth of engineering leaders identified team resistance to being measured as one of their main challenges.

This doesn't reflect the long-held suspicion that software developers don't like to be measured on their performance. It also suggests that the majority of engineering managers are not falling into the trap of attempting to measure individual performance using crude metrics like lines of code.

On the flip side, managers need to be careful about which metrics they choose and identify effective counterbalances to avoid incentivizing teams to "game" these measures. Instead, by carefully combining the most important metrics for you and your team, you can drive a team towards working collaboratively to deliver quality work and real impact, rather than raw outputs.

GitLab and Jira were the most cited team performance reporting tools

Q: What tools do you use to help you with reporting team performance?



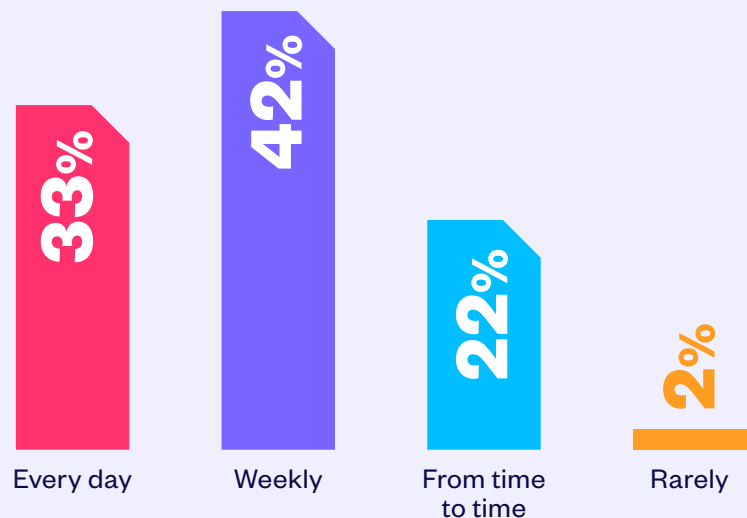
Team coordination

No team exists in complete isolation though, and 75% of leaders reported that they coordinate with other teams either weekly, or daily, with 22% doing so from time to time. Just 2% said they did so rarely.

But that doesn't mean collaboration is straightforward. Prioritization of tasks was cited as a key challenge to collaboration by 71% of respondents, while ownership and accountability were close behind at 67%. Lack of information and visibility was an issue for just over half of respondents. But for all the debate over remote or synchronous working, just 22% of respondents saw this as an issue.

75% of respondents coordinate with other teams on a daily or weekly basis

Q: How often do you have to coordinate work with other teams?



Case study

How Gorgias minimizes the cognitive load of engineers by focusing on collaboration

Gorgias is a SaaS customer service platform specializing in serving e-commerce businesses. Today, their engineering organization consists of 80+ software engineers across 10 teams, further split into tribes and squads.

Like many software organizations, Gorgias started out with a monolithic architecture, which served them well while the team was still small. Over time, the business grew in size and complexity, which caused some clear fracture lines to appear with every new integration and application. Suddenly, the engineers found themselves spending more time on navigating dependencies and less time on solving the specific customer problems their team was assigned to.

One of the biggest bottlenecks, according to co-founder and CTO, Alex Plugaru, was cross-team collaboration – and more specifically, the cognitive load it created for the engineers. The monolithic architecture was increasingly turning the core team into a bottleneck, as they were the only ones able to review a large proportion of new code from the other teams.

While there was no simple solution to reducing waiting times and cross-team dependencies, Alex started by introducing service-oriented architecture both inside the monolith and by introducing new dedicated standalone services.

Additionally, Gorgias started using Swarmia to recognize and address the concrete bottlenecks in cross-team collaboration. By giving engineers more visibility into what was slowing them down, they were able to proactively analyze and address the root causes of delays – and ultimately, focus more of their time on solving customer problems.

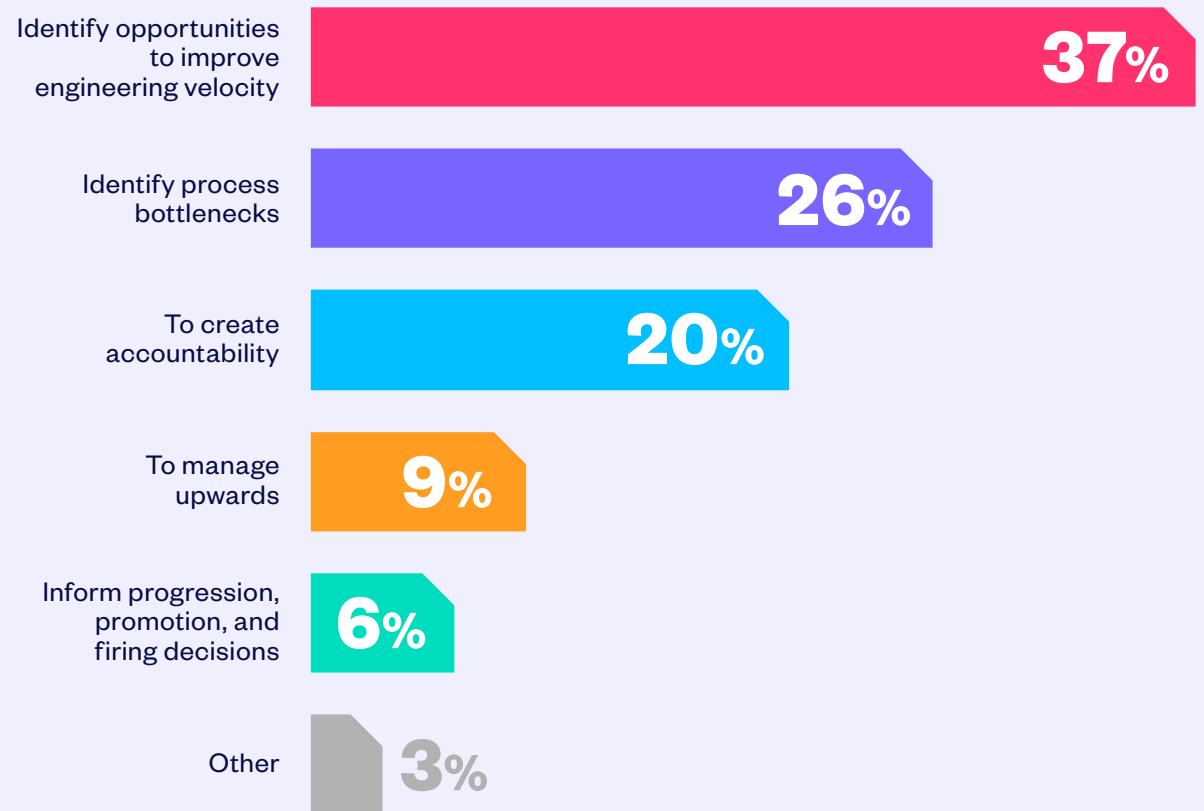
Performance metrics and measurement

Engineering leaders have multiple reasons to measure team performance, but rarely treat it as a personnel issue. This should comfort those more resistant team members who fear their career hinges on how many pull requests (PRs) they submit.

The most cited reason to measure team performance was to identify opportunities to increase velocity (37%), with a quarter citing an urge to pinpoint bottlenecks. Just 6% saw it as a way to inform progression, promotion, or firing decisions, which is heartening for an industry that continues to be impacted by widespread layoffs.

Improving velocity is the most common reason for measuring team performance

Q: Why is it important to measure your team's performance?



Likewise, the key methods to measure team performance were largely not focused on individuals. Meeting deadlines was the highest ranked, followed closely by productivity metrics, or the number of tickets completed. More qualitative and time-consuming pulse surveys came much further down the rankings, with heavyweight annual developer surveys lowest of all.

When it came to the most useful productivity measures, the highest ranked was cycle time, followed by deployment frequency, and lead time. Bottom of the list was days worked, as measurable outcomes outweigh raw effort measurement.

Meeting deadlines is the most important method for measuring team performance

Q: Rank the methods that you typically use to measure team performance at your organization



Cycle time is the most useful productivity measure

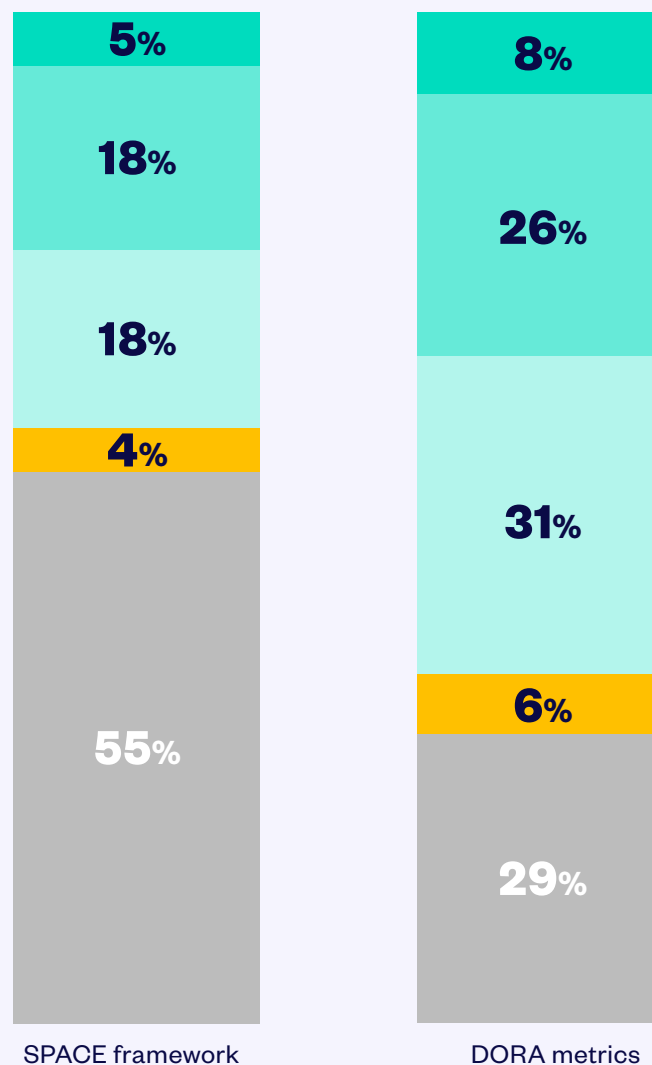
Q: Rank these productivity metrics in order of usefulness



Of those who commented, 50% think the SPACE framework is effective/very effective; and 48% think DORA metrics are effective/very effective

Q: How effective are DORA and SPACE at measuring team performance?

- Very effective
- Effective
- Somewhat effective
- Not effective
- Don't know/unsure



Many of these metrics will look familiar to anyone who has come across DevOps Research and Assessment (DORA) metrics. First released in 2014, these metrics looked to measure how effectively organizations were bringing their developers and operations teams together to deliver better software faster. Similarly, cycle time has its origins in agile software development, and aims to measure how long it takes to complete a certain task.

However, these metrics tend to focus on raw outcomes, and don't account for overall team health and effectiveness. Then, in 2019, some of the original DORA authors developed the satisfaction, performance, activity, communication, and efficiency (SPACE) framework, with the aim of providing a more holistic picture of team performance.

While these two frameworks have set themselves apart from others as close to an industry standard, when asked about the effectiveness of DORA metrics in measuring performance, 29% of respondents said they didn't know, or were unsure. Then for the newer SPACE framework, well over half were unable to give an opinion.

However, amongst those that were familiar with the two frameworks, the response was largely enthusiastic. Roughly half of respondents see SPACE as either a very effective or effective framework, with roughly the same for DORA. Less than one in ten respondents found either framework to be ineffective.

While these metrics have proved popular, there are a plethora of other things companies have come to measure, from the number of times a ticket circulates between developers and QA, to time utilization, and opportunity costs. All of which can be measured and combined according to the priorities of your team and business.

There is also a clear interest in establishing some common developer well-being metrics, such as cognitive load, team happiness, and the prevalence of context switching. When it comes to the most useful methods to gauge team health, a simple 1:1 meeting was the clear leader, named by 94% of respondents, with a continuous feedback mechanism named by 79%.

As you would expect, there were strong feelings expressed when respondents were asked which metrics they might avoid and why. Lines of code was commonly identified as “gameable”, and a cheap way to measure quantity over quality. Likewise, velocity needs a clear definition to be useful: it could mean a team is moving quickly – but on the wrong things.

By focusing on satisfaction, well-being, and collaboration alongside more traditional output metrics, engineering leaders can start to establish an effective way of measuring the overall performance of their engineering teams.

“By focusing on satisfaction, wellbeing, and collaboration alongside more traditional output metrics, engineering leaders can start to establish an effective way of measuring the overall performance of their engineering teams.”

Engineering tools and processes

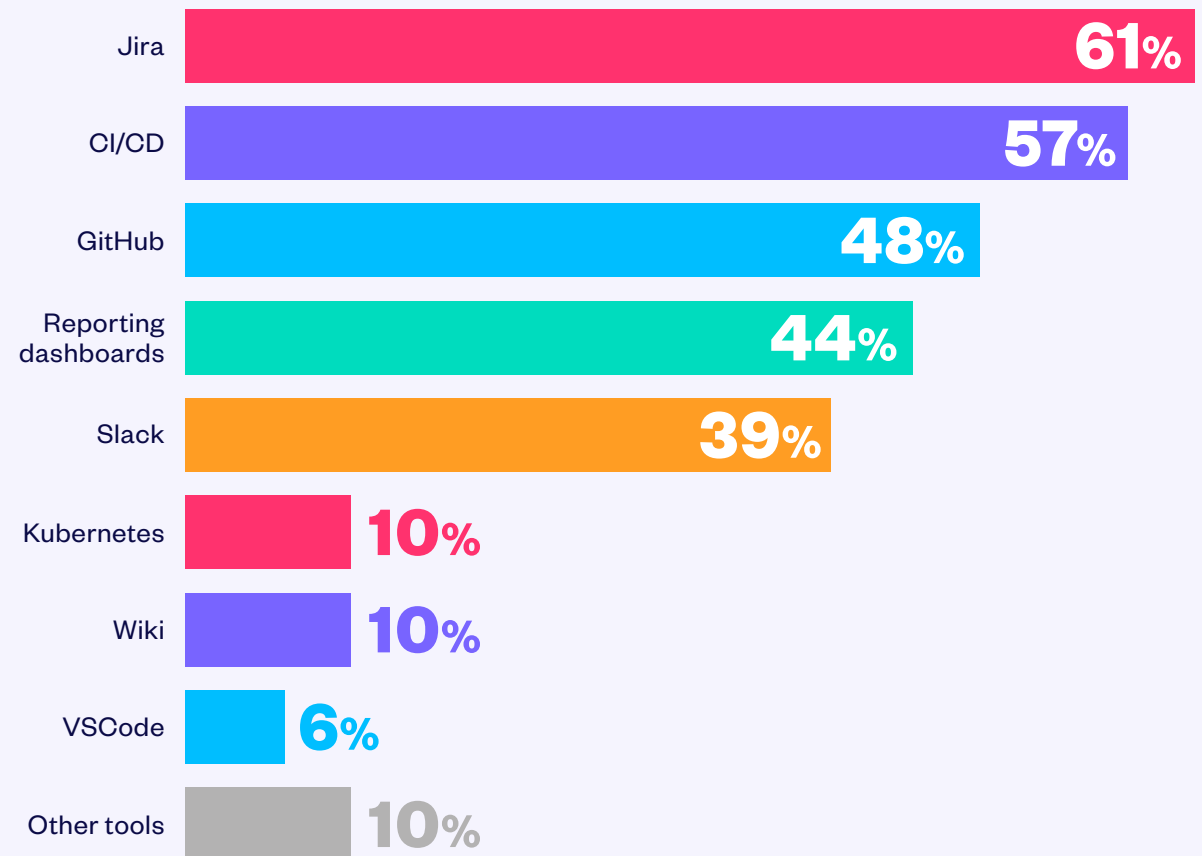
Engineering leaders will want to be clear about the OKRs, KPIs, and output metrics they are working towards. But the tools and processes they are using to get there are incredibly diverse.

Jira might not be the most fashionable tool – with many practitioners complaining it is over-complicated, unwieldy, and restrictive – but it was cited by 61% of respondents when asked which tools their teams used to do their best work.

Continuous integration/continuous delivery (CI/CD) tools were cited by 57% respectively, with version control system GitHub coming in at 48%. While reporting dashboards and the collaboration platform Slack were helping a large number of organizations, the number saying the notoriously complex container orchestration tool Kubernetes helped their teams do their best work was just 10%.

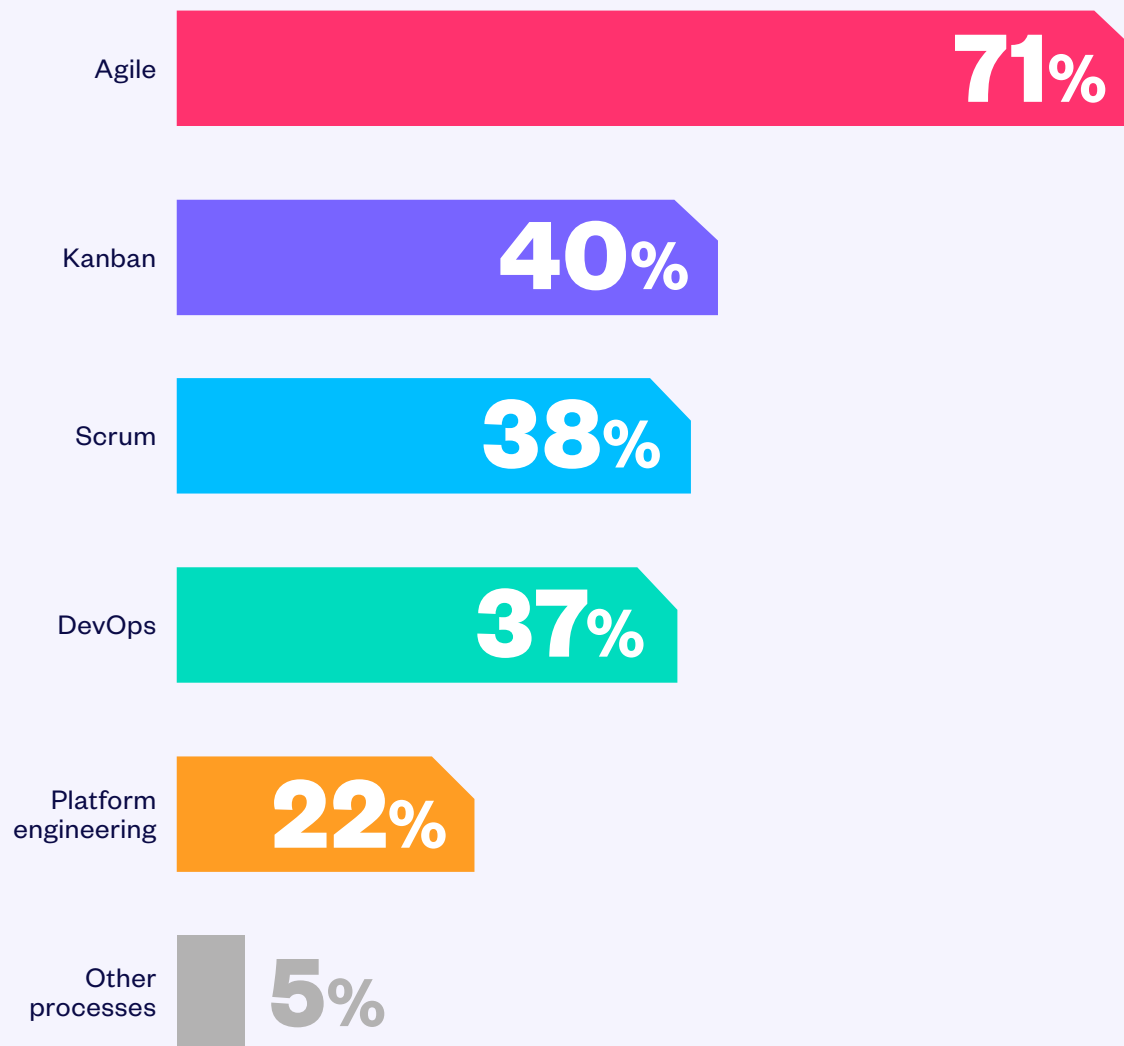
The most effective tools used are Jira and CI/CD

Q: What tools do your teams use to do their best work?



Agile continues to help teams do good work

Q: What processes do your teams use to do their best work?



Likewise, the agile methodology isn't new or exciting, but it was the most cited process enabling that quality work, at 71%. Next was Kanban and Scrum, both of which are established approaches to implementing agile ways of working, being cited by around 30% of respondents.

DevOps and its fast-emerging cousin platform engineering were named by 37% and 22% respectively. This is a nice reminder that what grabs headlines often takes time to deliver real value and help get engineering projects out the door.

None of these tools are perfect though. Asked which tools inhibited productivity, the lineup is much the same, with Jira named by 27%, and CI/CD by 24%. The relentless ping of Slack notifications put it in third place. Notably, 29% of engineering leaders felt that no tool inhibits team productivity, showing that there are plenty of good builders unwilling to blame their tools.

When it came to which processes inhibited productivity, Scrum was cited by 27%, ahead of agile at 21%, while platform engineering made the top three at 18%. But 35% of respondents couldn't name a process that inhibits productivity.

But are tools and processes ever going to be the biggest drag on productivity? Asked to identify the number one bottleneck for engineering teams today, two key factors were each named by over a quarter of our engineering leaders: lack of clarity and prioritization, alongside headcount and staffing.

Working with other teams and brittle systems were the next most cited culprits, at 12% and 11%, respectively. Metrics were cited as a bottleneck by just 2%.

Engineering managers and their bosses both cited lack of clarity or prioritization as a key problem, but software engineering directors were clear that headcount and staffing issues were their biggest bottleneck. This suggests that staffing problems trickle down an organization, transforming into a lack of clarity and prioritization issues along the way. Neither is good. Together they can be fatal over time.

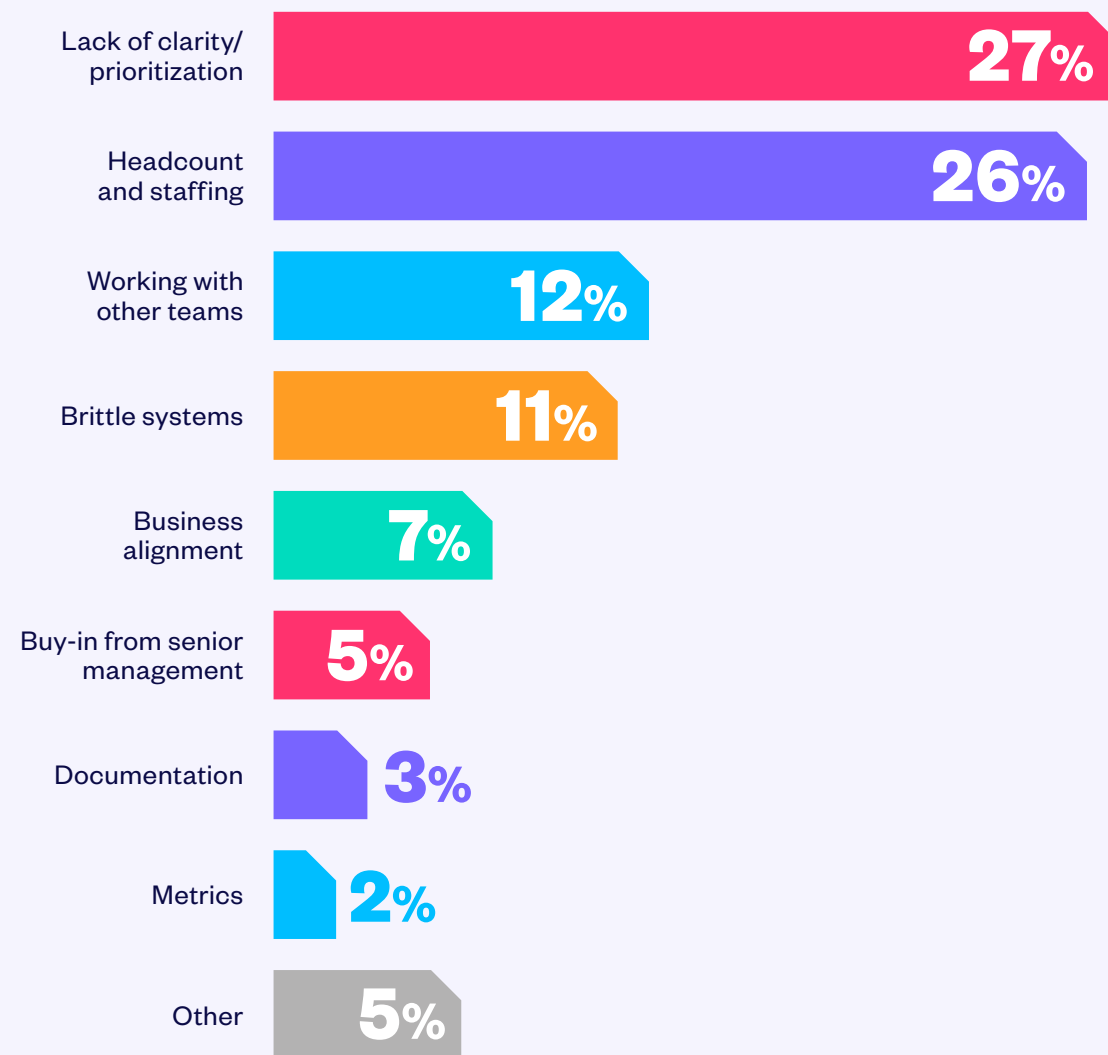
Competing priorities and limited resources make it hard to focus, while conflicts between product managers (PMs) undermines even the clearest of vision. Similarly, collaboration between teams is undermined when they have different priorities.

While some teams have doubtless felt the pain from layoffs, hiring freezes, and attrition this year, others simply struggled to find capable people and develop leadership skills internally. Still, others pointed to a generational gap between junior and senior members of staff as compounding staffing headaches.

These are the sorts of systemic problems that can scupper an organization, and metrics, methodologies, and tooling alone can't fix them.

The no.1 bottleneck for engineering teams today was a lack of clarity or clear prioritization

Q: What is the no.1 bottleneck for your teams today?



Final thoughts

We're all living through an acutely volatile period of change in the technology industry, with increasingly compressed cycles of change and disruption, and engineering leaders are being tasked with mapping and navigating these shifts.

This research highlights the range of challenges engineering leaders are facing when it comes to measuring and improving the performance of their teams. It's clear that some of these – headcount and staffing issues in particular – are partly due to forces outside their control. But senior management can impact the way these factors ripple through the organization, particularly in how they are communicated, and how they influence your organization's priorities.

It also highlights the diversity of approaches, processes, and tools that engineering leaders are still utilizing at a team or organizational level to set goals and measure progress towards achieving them. This makes it harder to benchmark a team's performance and may explain the satisfaction leaders have with DORA and, increasingly, SPACE when they are implemented.

Of course, individuals might have their own views about how particular methods or tools contribute to the gap between dreams and reality, sometimes based on experience,

sometimes not. Jira and agile are seen by many as helping them do their best work – but, conversely, many engineers also see them as outdated inhibitors.

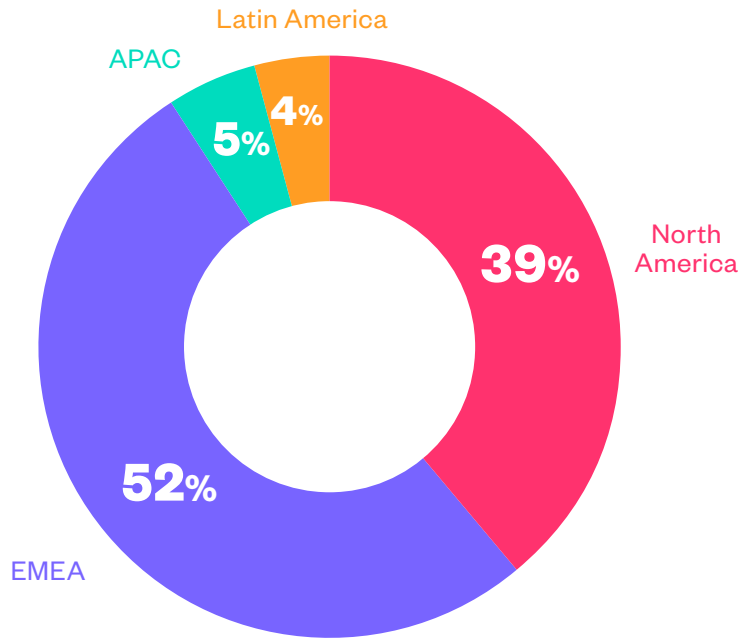
It can only be reassuring that even amidst all these challenges, there are plenty of leaders who are focused on team and individual well-being. Even if these remain tricky to measure definitively for now, efforts like SPACE could make it easier for engineering leaders to do so over time and demonstrate the importance of these factors to other team members and senior management.

We hope this research helps you put your own challenges in context with the rest of the industry, and helps guide the actions you're taking to address them. We look forward to hearing how you've fared next year.

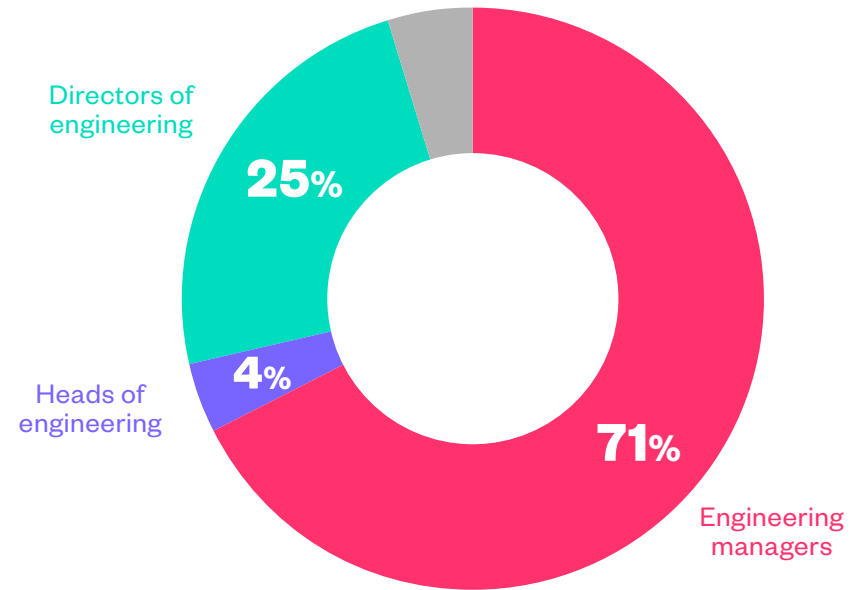
“This research highlights the range of challenges engineering leaders are facing when it comes to measuring and improving the performance of their teams.”

Methodology

This report is based on a survey of 605 engineering leaders across North America (32%), EMEA (52%), APAC, and Latin America.



Among the respondents, 95% manage a team, with 71% being engineering managers, 4% heads of engineering, and 25% directors of engineering.



The digital survey ran from 18 August until 1 September 2023.



Successful leaders never stop learning

Your options for leadership development with LeadDev:

Free articles
and videos

Inspiring
conferences

Networking
and meetups

Expert-led webinars, workshops,
and group learning courses

LeadDev can help you level up your management skills,
whatever your budget and no matter where you are in your career.

Visit leaddev.com to find out more



Follow us on LinkedIn
[Linkedin.com/company/leaddev](https://linkedin.com/company/leaddev)



Follow us on YouTube
[@LeadDev](https://youtube.com/@LeadDev)



Follow us on Twitter
[@TheLeadDev](https://twitter.com/TheLeadDev)