Engineering team performance report
2024

LeadDev O'REILLY®

Foreword

It's been one of the hottest-debated topics in tech circles over the past few years: can you effectively measure and optimize software engineering team performance?

While DORA and SPACE frameworks are gaining traction, there is still no single way to measure developer team performance. As such, engineering leaders are left bearing the cognitive load of quantifying their team's output against wider company goals, with no one true approach.

Last year's *LeadDev Engineering Team Performance report* aimed to clarify how leaders go about understanding team performance. Now, one year on, can we start to define best practices for engineering managers? Which measurement methods are rising and which are falling? Is your message getting through? And how can you boost team performance when times are tighter than before?

In October, we surveyed nearly 1,000 engineering leaders to find out how organizations measure their team performance and their biggest challenges. We hope this helps you identify where you are in your team performance journey, and perhaps spark some ideas on how to drive meaningful improvements.

Thank you for reading.
Now go build epic teams!

Scott Carey
Editor in Chief, LeadDev

LeadDev

A note from O'Reilly

O'Reilly's mission is to change the world by sharing the knowledge of innovators. For over 45 years, we've inspired companies and individuals to do new things – and do things better – by providing them with the skills and understanding that are necessary for success.

By partnering with LeadDev to create this report, we continue in that tradition with you.

At the heart of our business is a unique network of experts and innovators who share their knowledge through us. The O'Reilly learning platform offers exclusive live tech and business events, AI-powered answers you can trust, interactive learning, on-demand courses, a certification experience, videos, and more, making it easier for our customers to develop the expertise they need to get ahead. And our books have been heralded for decades as the definitive place to learn about the technologies that are shaping the future.

Everything we do is to help professionals from a variety of fields learn best practices and discover emerging trends that will shape the future of the tech industry. When you and your teams are ready to learn what's next, you'll find us at oreilly.com.

O'REILLY®

Key takeaways

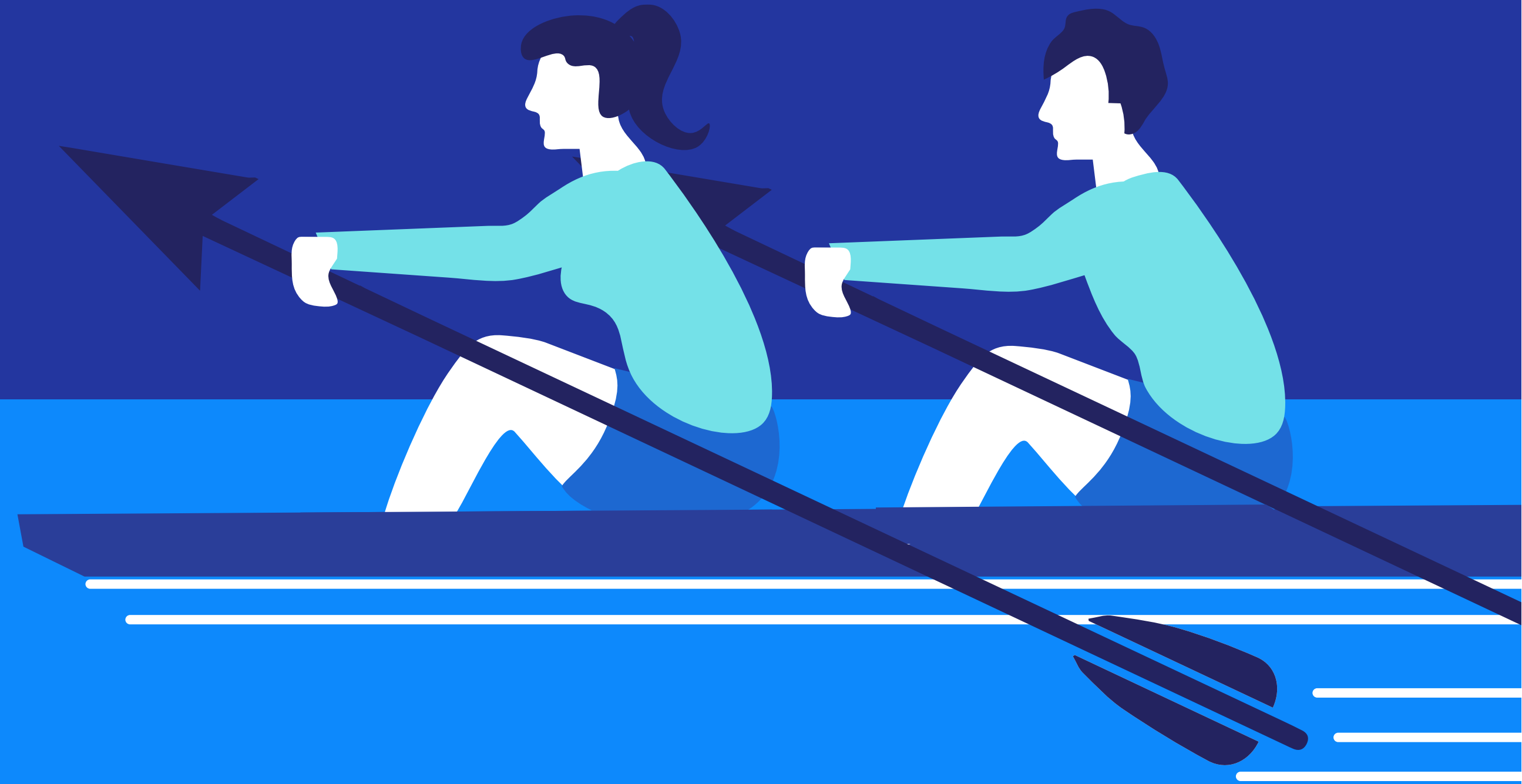**1** **Cycle time** was ranked as the most useful engineering productivity metric for the second year running

30% Are using their **own dashboards** to measure team performance

78% Want to **plug technical knowledge gaps** in their teams through upskilling

70% Actively avoid **measuring lines of code**

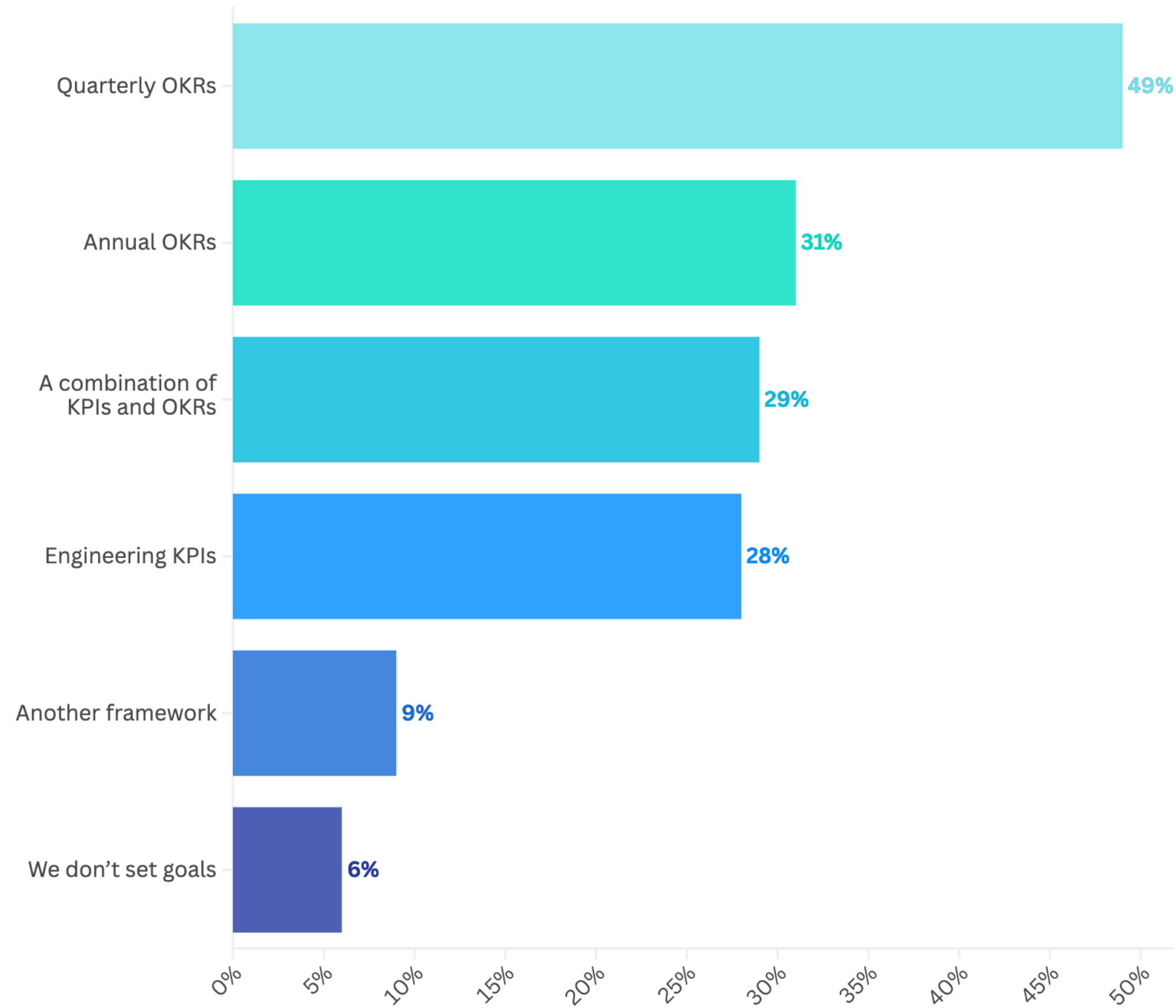Awareness and satisfaction with the **DORA & SPACE frameworks is growing**
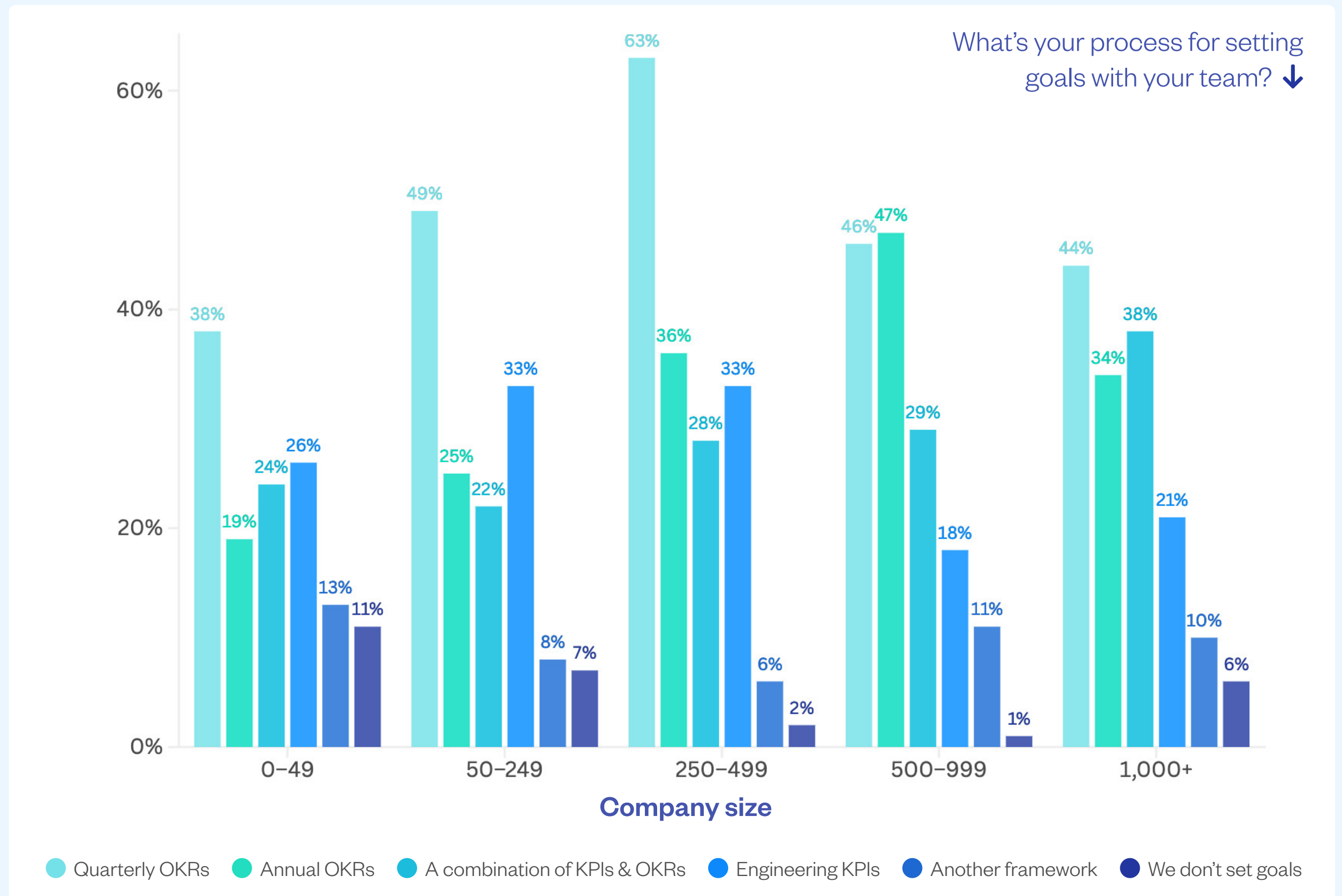
Goal setting

Goals

Goals and objectives provide the structure within which engineering teams work and measure their success. Before developers even touch a keyboard, they need to know why they are sitting at their desks in the first place.

Company-wide objectives and key results (OKRs) appear to be the way to define and communicate business goals, though the cadence varies across businesses. **Half of respondents set quarterly OKRs**, while nearly a third do so annually.

The other key framework for setting goals was key performance indicators (KPIs), which 28% of organizations rely on, up from just 14% last year. Another 29% use a combination of OKRs and KPIs, making it the third most popular approach. A surprising 6% don't set any goals at all, many of whom identified themselves as working for small start-ups or software consultancies.

What's your process for setting goals with your team? ↓

Quarterly OKRs	49%
Annual OKRs	31%
A combination of KPIs and OKRs	29%
Engineering KPIs	28%
Another framework	9%
We don't set goals	6%

What's your process for setting goals with your team? ↓

Company size

● Quarterly OKRs ● Annual OKRs ● A combination of KPIs & OKRs ● Engineering KPIs ● Another framework ● We don't set goals
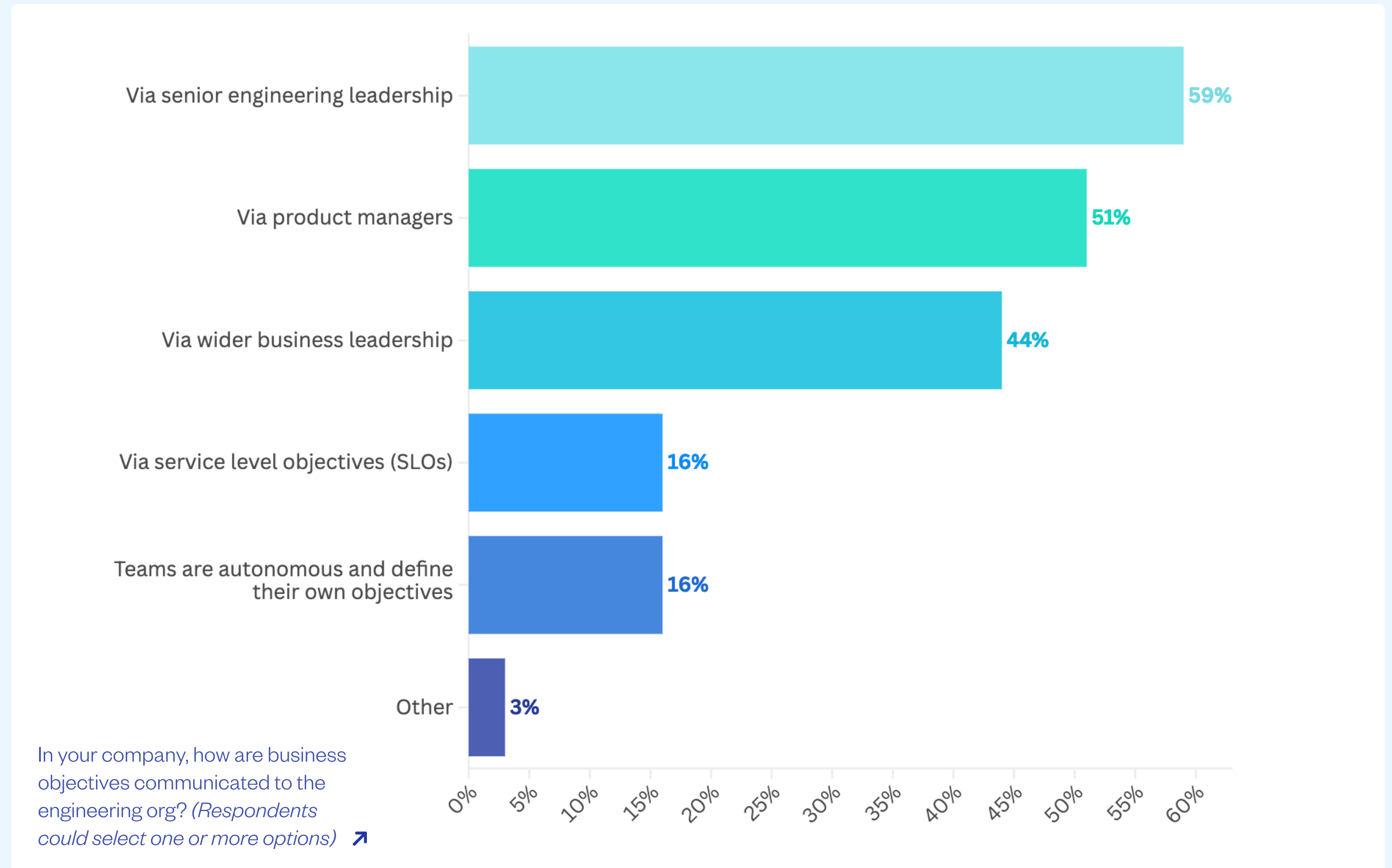
Communicating business goals

OKRs, KPIs, or a combination, are useless if they aren't communicated effectively.

While all goals must be communicated widely and often, senior leadership is responsible for communicating business objectives in 59% of respondent organizations.

Product managers are also responsible for conveying objectives at half of organizations, where wider business leadership is responsible 44% of the time. A small but not insignificant 16% of respondents have teams set their own goals and objectives autonomously.

Via senior engineering leadership — 59%
Via product managers — 51%
Via wider business leadership — 44%
Via service level objectives (SLOs) — 16%
Teams are autonomous and define their own objectives — 16%
Other — 3%

In your company, how are business objectives communicated to the engineering org? *(Respondents could select one or more options)* ↗

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

- **0%** No understanding
- **3%** Little understanding
- **3%**
- **15%** Some understanding
- **27%** Complete understanding
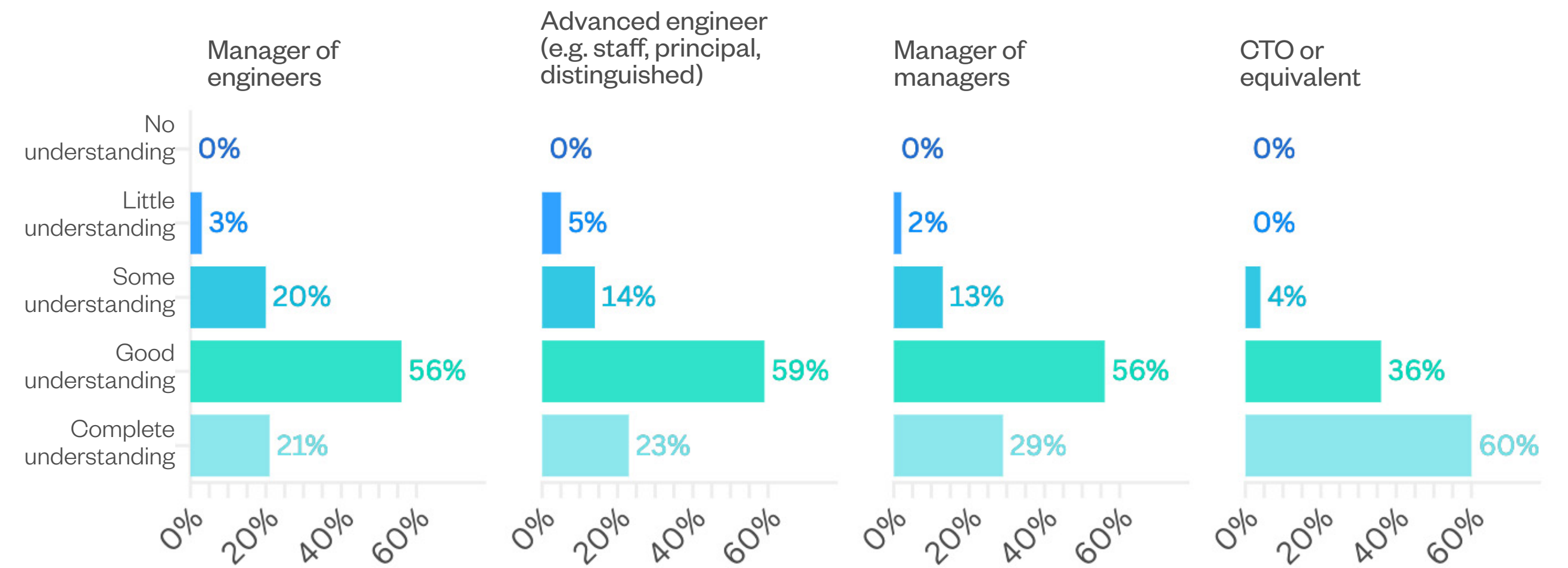- **55%** Good understanding

But is the message getting through? Just 27% of respondents say they have a "complete understanding" of their organization's strategic business goals, with almost 55% saying they have a "good understanding." That leaves a worrying 15% with just "some understanding" and 3% in the dark about their company's strategy.

There are signs that the message isn't filtering down for most organizations, too. While 60% of respondents at the CTO or equivalent seniority level have a complete understanding of strategic business goals, this drops to 29% for managers of managers, and 21% for engineering managers.

This task also becomes harder the bigger your organization is. While 50% of respondents at small companies have a complete understanding of strategic business goals, this drops to 18% at companies of 1,000+ employees.

How well do you understand your company's strategic business goals? *(By role)* ↓

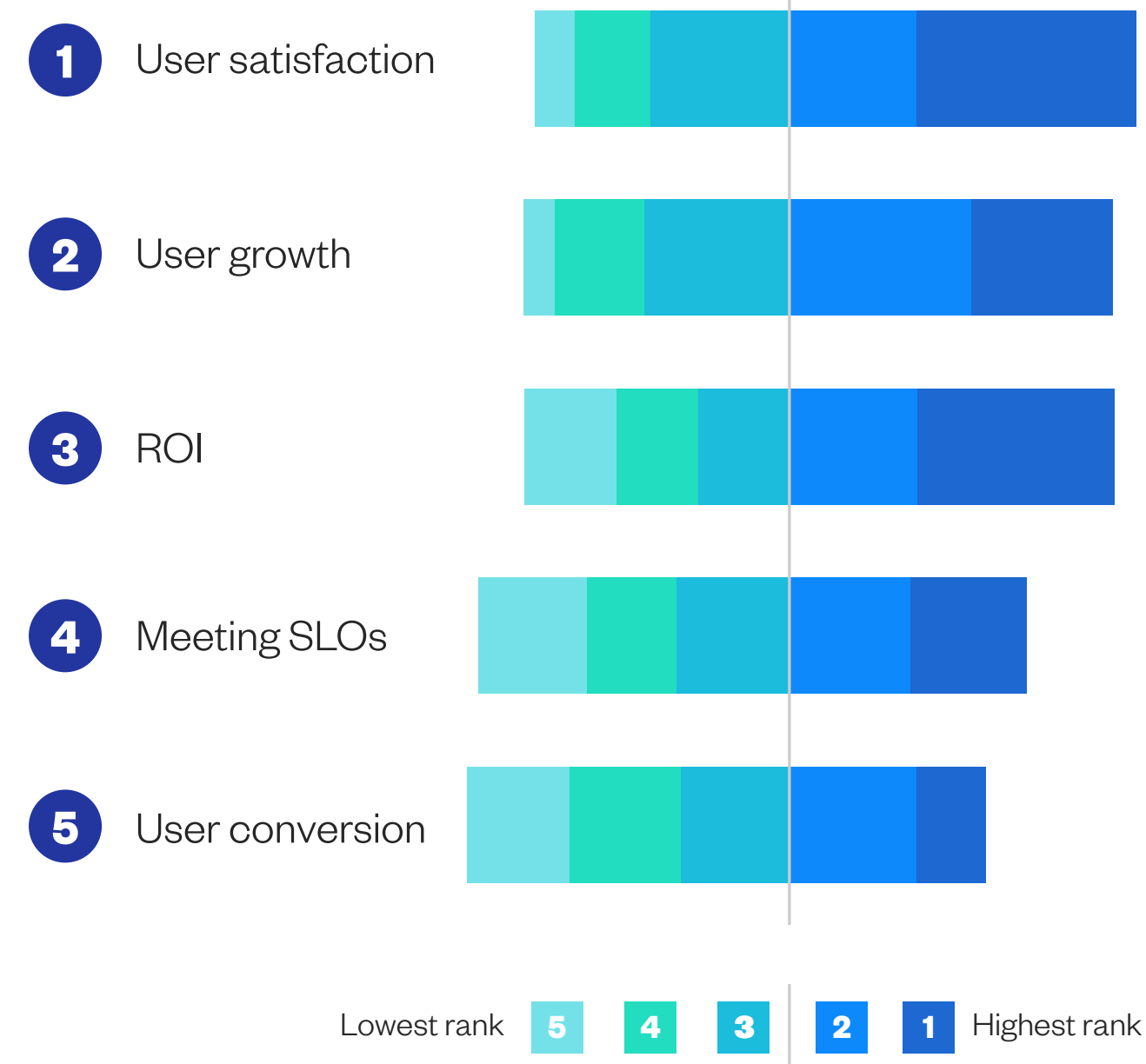|	Manager of engineers	Advanced engineer (e.g. staff, principal, distinguished)	Manager of managers	CTO or equivalent
No understanding	0%	0%	0%	0%
Little understanding	3%	5%	2%	0%
Some understanding	20%	14%	13%	4%
Good understanding	56%	59%	56%	36%
Complete understanding	21%	23%	29%	60%

Measuring progress

A big part of engineering leadership involves gauging the impact teams have on the company's bigger goals.

The first place many leaders look is customer response. The highest-ranked metric for measuring impact was **user satisfaction**, overtaking the more quantitative measure of **user growth** from last year. Next came return on investment (ROI), with meeting SLOs ranked fourth.

But those aren't the only measures teams are watching, with some choosing to measure more commercial metrics like revenue and annual recurring revenue churn, retention rates, Net Promoter Scores (NPSs), and lifetime value (LTV).

Which of the following metrics are used in your engineering org to measure impact? ↓

1 User satisfaction

2 User growth

3 ROI

4 Meeting SLOs

5 User conversion

Lowest rank 5 4 3 2 1 Highest rank

How to read this rank scale

Respondents were asked to rank their responses in order of importance, where 1 was the most important. They were asked to only rank items which were important to their engineering org.
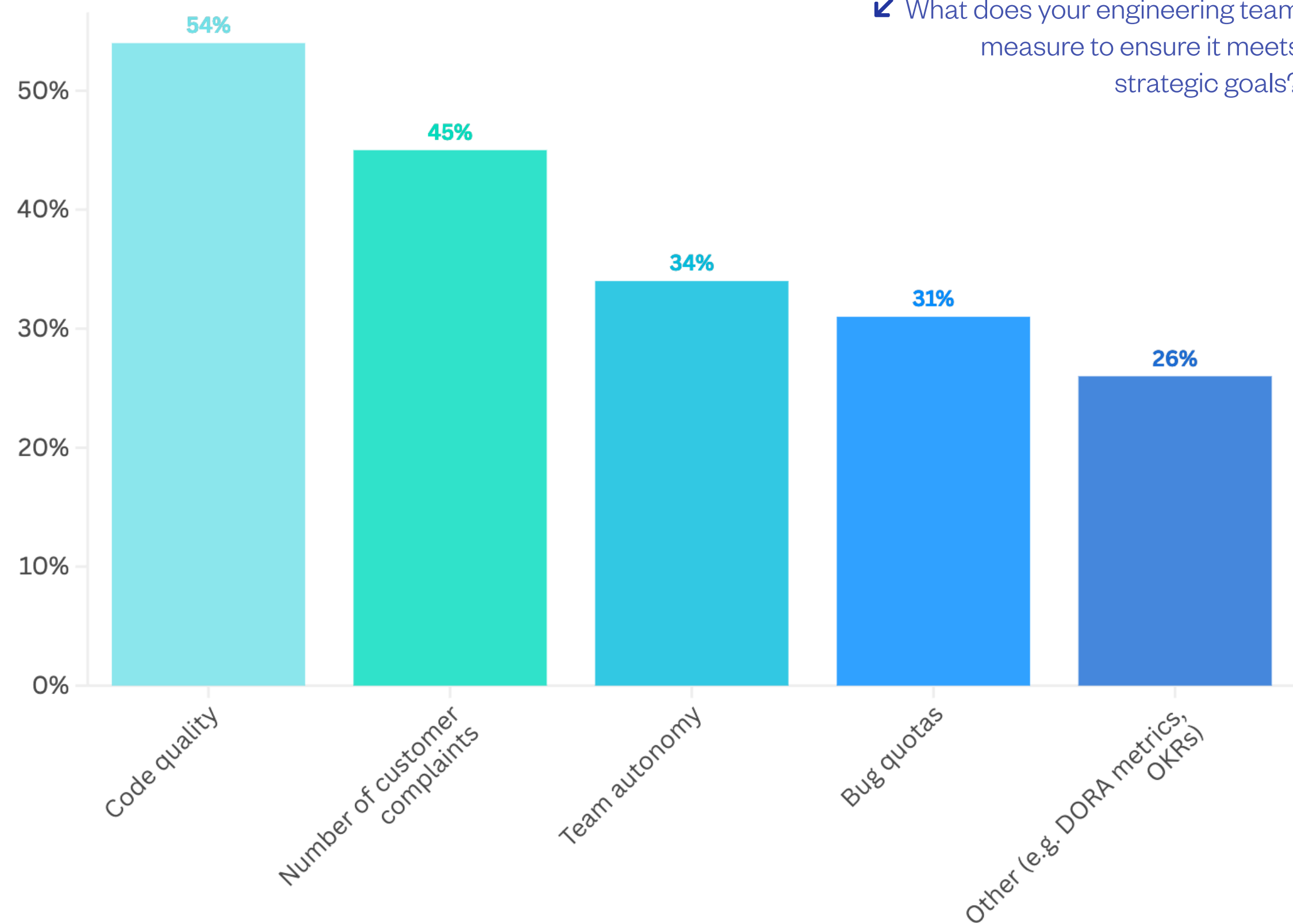
The size of each block in the scale corresponds to the number of rankings in that position.

In terms of measuring **how they are meeting broader strategic goals**, code quality is the most popular measure of success at 54%, with the number of customer complaints coming in second at 45%. Bug quotas are a useful measure for 31% of respondents. Then 34% of respondents look to measure the level of team autonomy. The numbers show most organizations rely on a combination of some of these metrics to gauge impact.

A third of respondents do not report on their team's performance at all. Of those who do, **31% favored monthly reviews**. A further 18% of respondents report on performance weekly and 17% quarterly, down from 23% last year.

↙ What does your engineering team measure to ensure it meets strategic goals?

- Code quality — 54%
- Number of customer complaints — 45%
- Team autonomy — 34%
- Bug quotas — 31%
- Other (e.g. DORA metrics, OKRs) — 26%

↙ How often are you required to report on team performance?

2024
- 33% Never
- 18% Weekly
- 31% Monthly
- 17% Quarterly
- 0% Other

2023
- 25% Never
- 19% Weekly
- 22% Monthly
- 23% Quarterly
- 11% Other
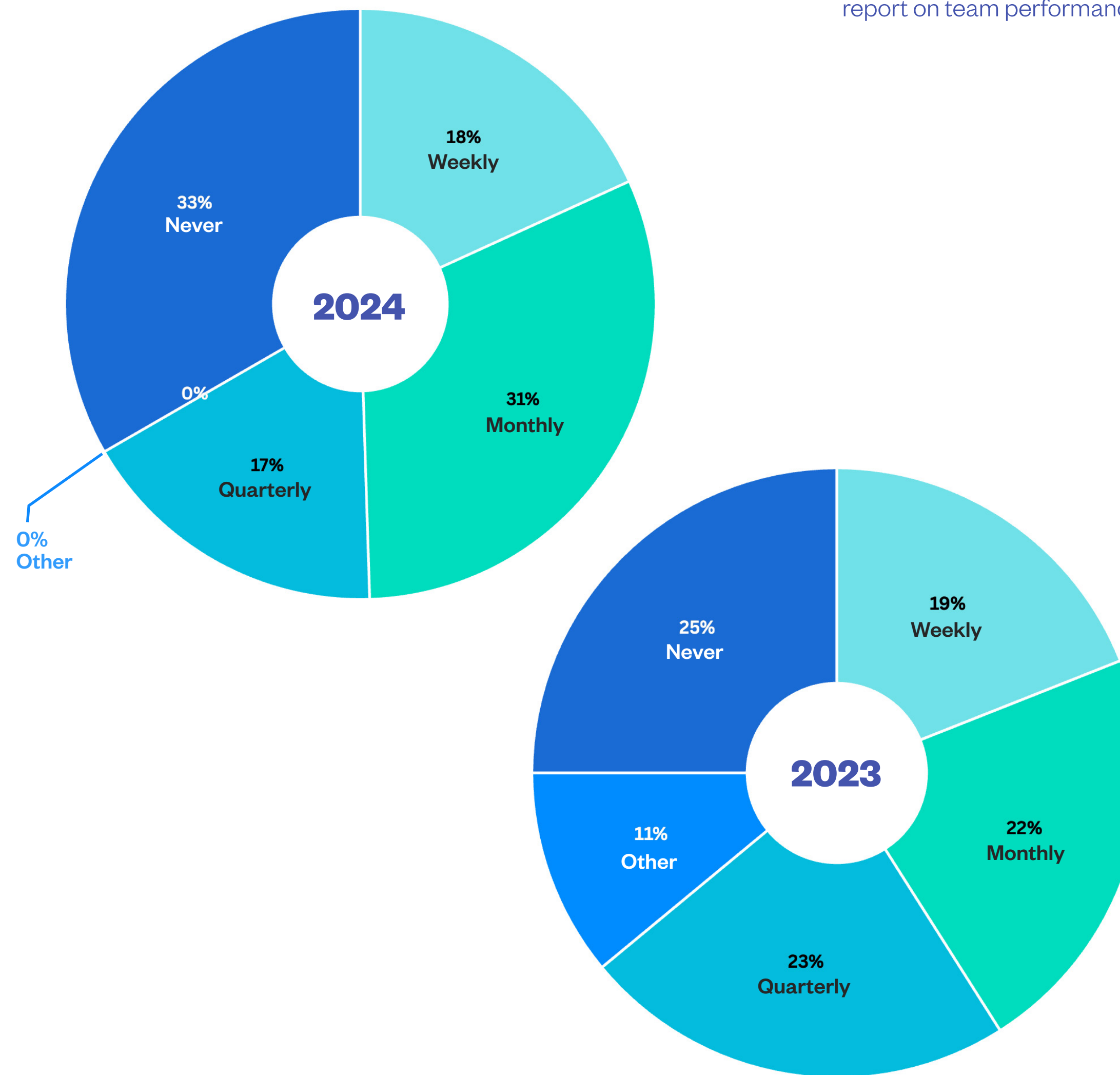
Bottlenecks

When respondents were asked to identify the number one bottleneck for engineering teams today, two key factors emerged: **lack of clarity and prioritization** and **headcount and staffing**, each being cited by 25% of respondents.

Additional bottlenecks, such as working with other teams and aligning with the rest of the business, were cited by 10% of respondents, while metrics themselves were an issue for just 2%.

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

- 2% Metrics
- Documentation
- Other
- Buy-in from senior management
- 3%
- 3%
- 4%
- 8% Brittle systems
- 9% Lack of skills
- 10% Working with other teams
- 11% Alignment with business
- 25% Headcount
- 25% Lack of clarity/ prioritization
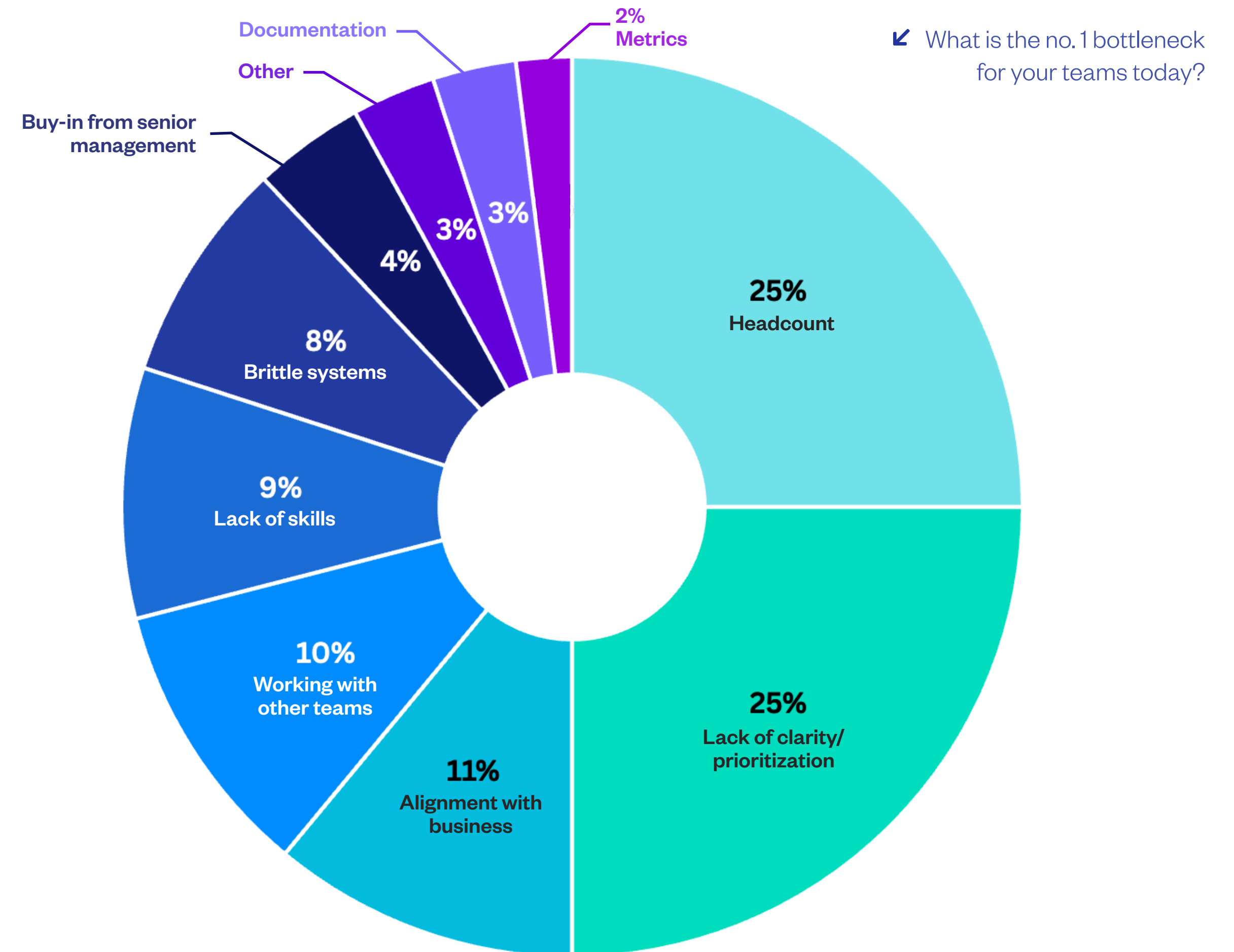
Team coordination

No team exists in complete isolation, with 47% of respondents coordinating with other teams weekly, and 28% daily. That leaves 22% doing so occasionally and just 3% doing so rarely.
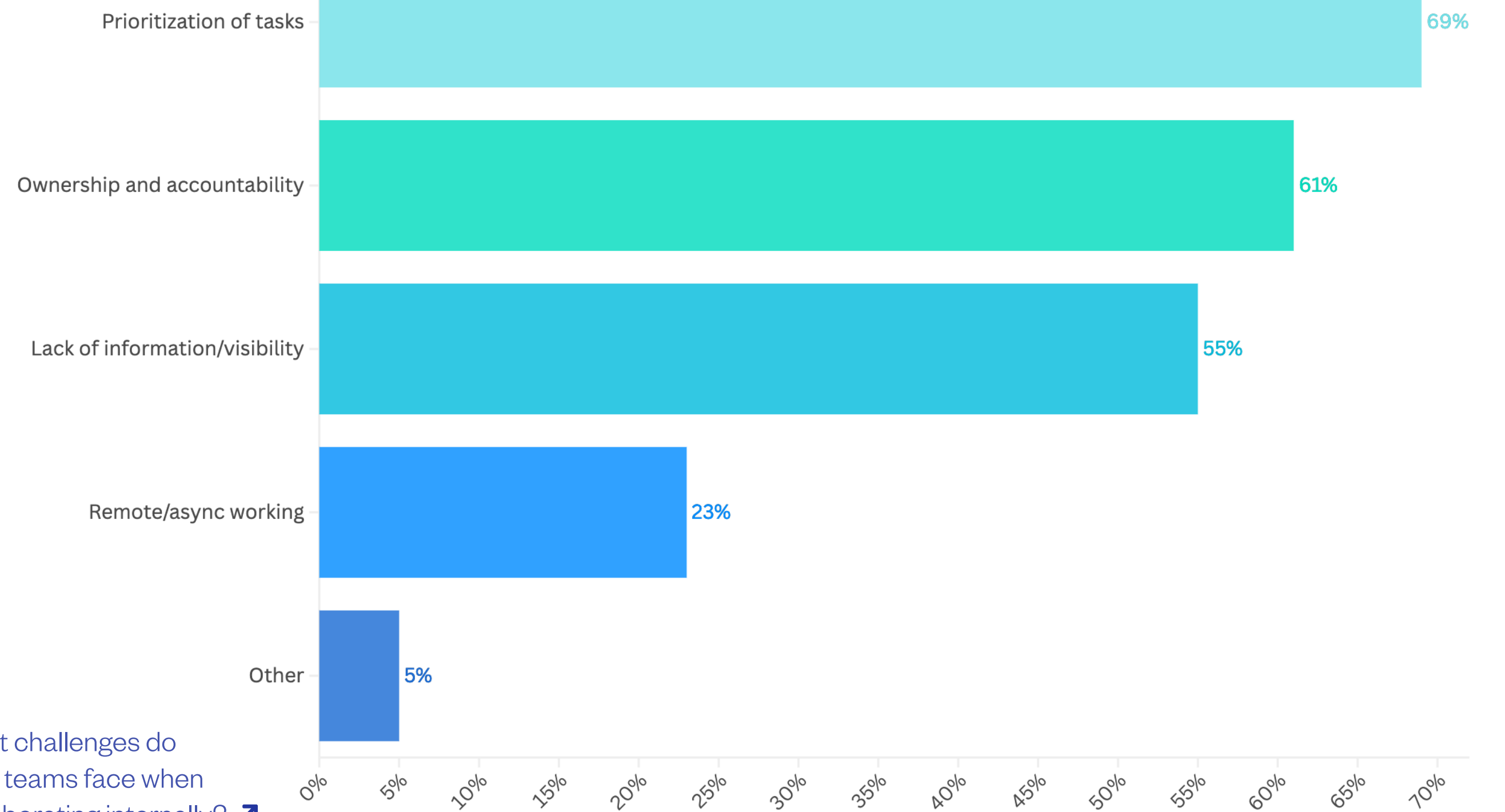
But collaboration isn't straightforward. Prioritization of tasks was cited as a key challenge to collaboration by 69% of respondents, while ownership and accountability was next at 61%.

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 23% of respondents saw this as an issue.

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

- 3% Rarely
- 1% Never
- 3%
- 28% Every day
- 22% From time to time
- 47% Weekly

What challenges do your teams face when collaborating internally? ↗

- Prioritization of tasks — 69%
- Ownership and accountability — 61%
- Lack of information/visibility — 55%
- Remote/async working — 23%
- Other — 5%
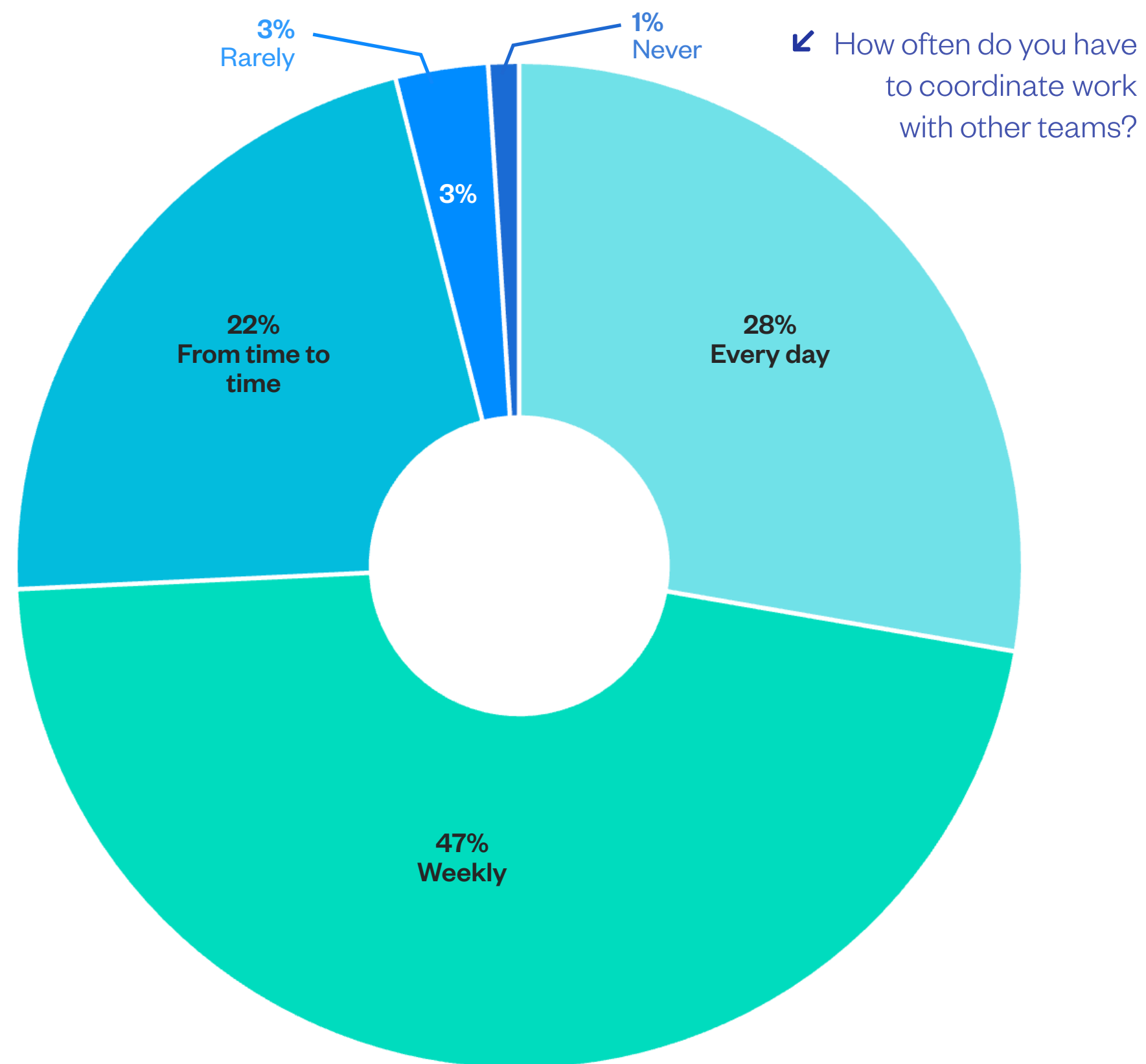
SECTION 2:

Performance metrics and measurement

Performance metrics and measurement

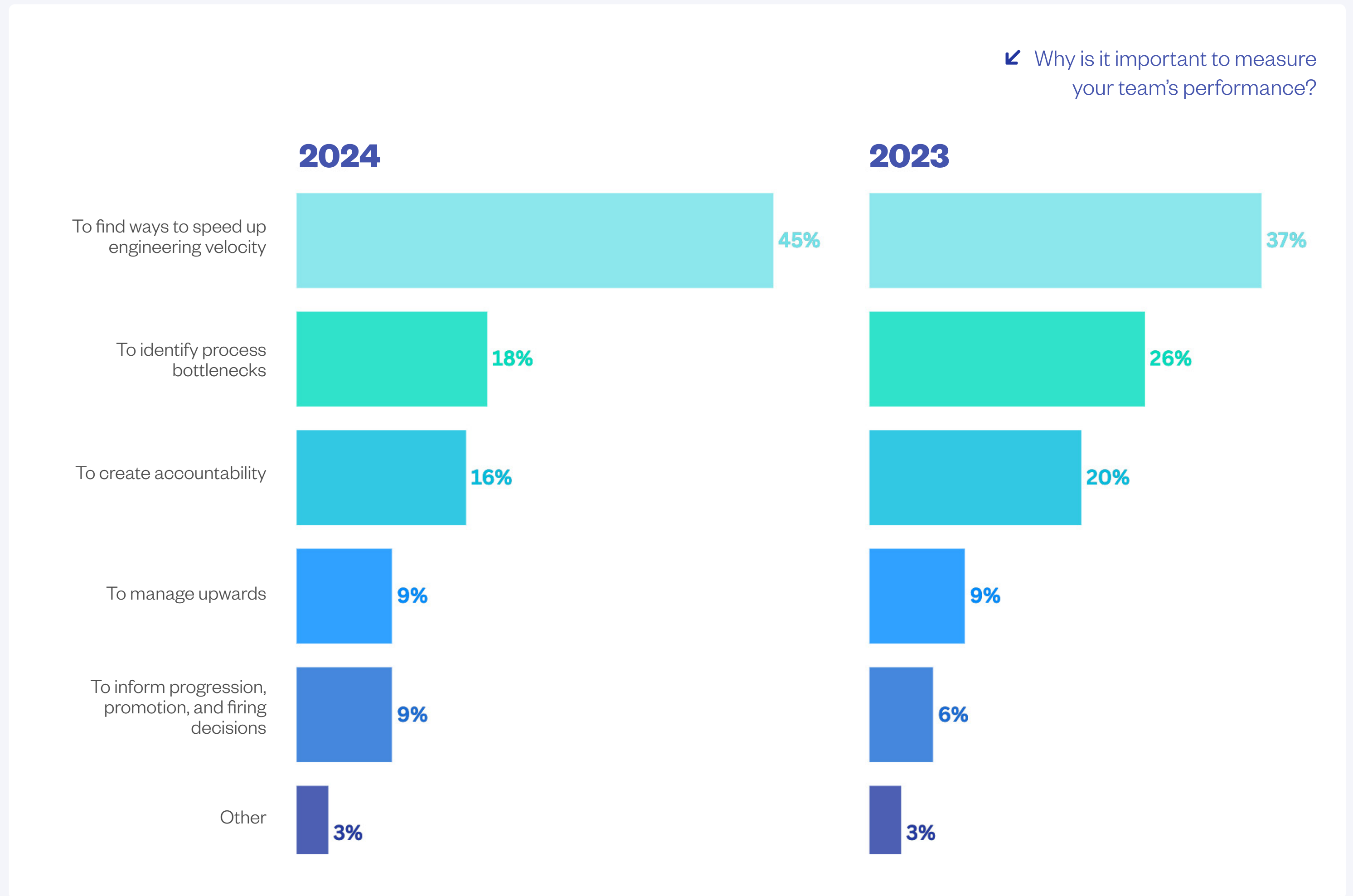Engineering leaders have multiple reasons to measure team performance, but **finding opportunities to increase velocity** is the most important to 45% of respondents, up from 37% last year.

No other reason came close. Pinpointing bottlenecks was the next most cited reason at 18% and fostering accountability at 16%, but this rises to 27% if you just asked your CTO. Just 9% saw metrics as a way to inform progression, promotion, or firing decisions.

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

2024

To find ways to speed up engineering velocity	45%
To identify process bottlenecks	18%
To create accountability	16%
To manage upwards	9%
To inform progression, promotion, and firing decisions	9%
Other	3%

2023

To find ways to speed up engineering velocity	37%
To identify process bottlenecks	26%
To create accountability	20%
To manage upwards	9%
To inform progression, promotion, and firing decisions	6%
Other	3%

For the second year running, cycle time was ranked as the most useful productivity metric. Originating with agile software development, cycle time aims to measure how long it takes to complete a certain task.

It was closely followed by lead time and deployment frequency. At the bottom of the list was work in progress (WIP) balance and the number of story points completed, as as results took priority over effort-based metrics.

Rank these productivity metrics in order of usefulness ↓

1. Cycle time
2. Lead time
3. Deploy frequency
4. Change failure rate
5. Mean time to restore
6. Number of story points completed
7. Frequency of context switching
8. WIP balance

Lowest rank 8 7 6 5 4 3 2 1 Highest rank

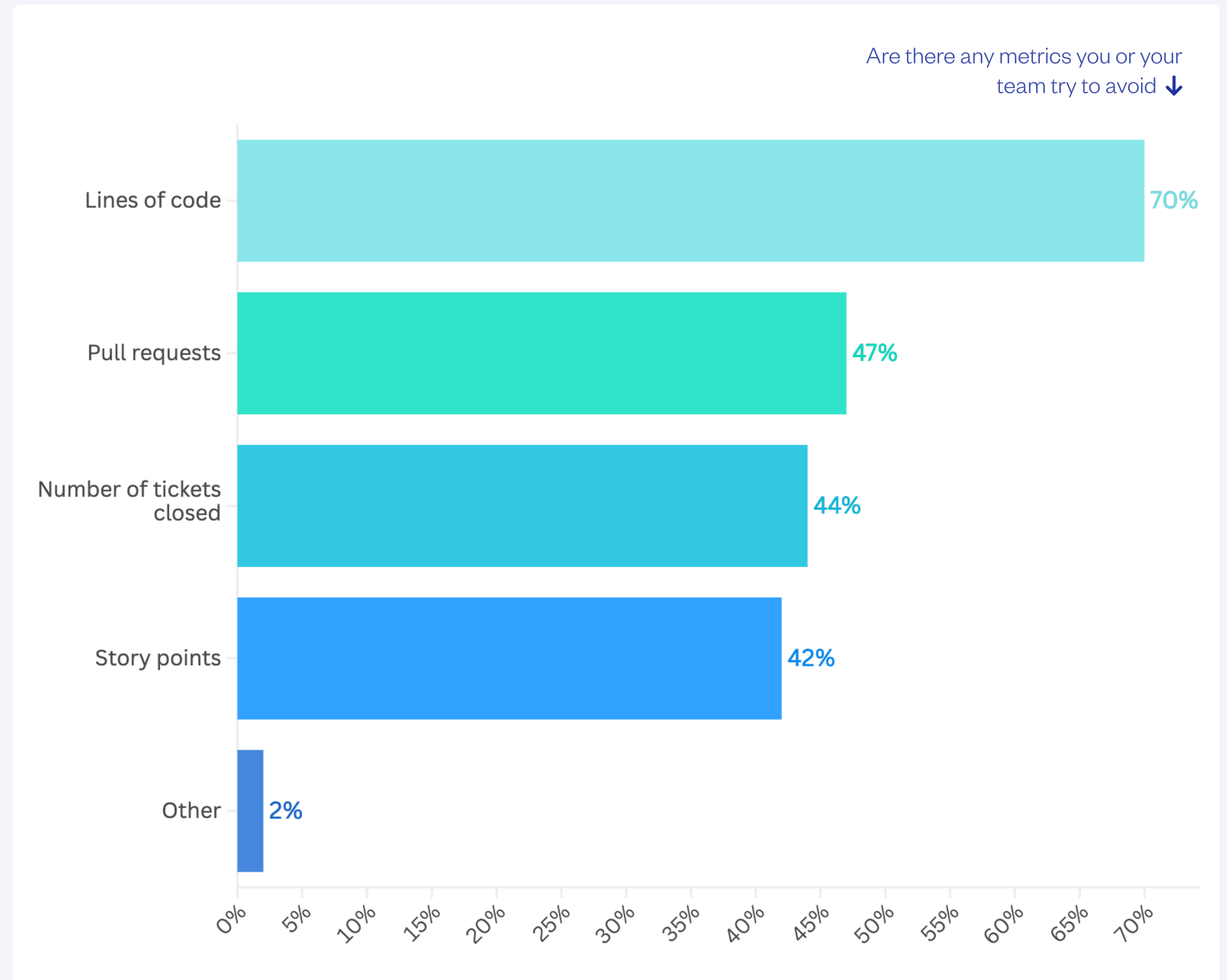**How to read this rank scale**

Respondents were asked to rank their responses in order of importance, where 1 was the most important. They were asked to only rank items which were important to their engineering org.

The size of each block in the scale corresponds to the number of rankings in that position.

When respondents were asked which **metrics they might avoid** and why, lines of code was cited by 70% of respondents. Then story points, pull requests, and the number of tickets closed all elicited similar feelings, with 42-47% not liking these measures in isolation.

When asked why they were suspicious of these metrics, their ability to be "gamed" or manipulated, inherent lack of context, and a misplaced focus on output over outcomes were all cited.

Are there any metrics you or your team try to avoid ↓

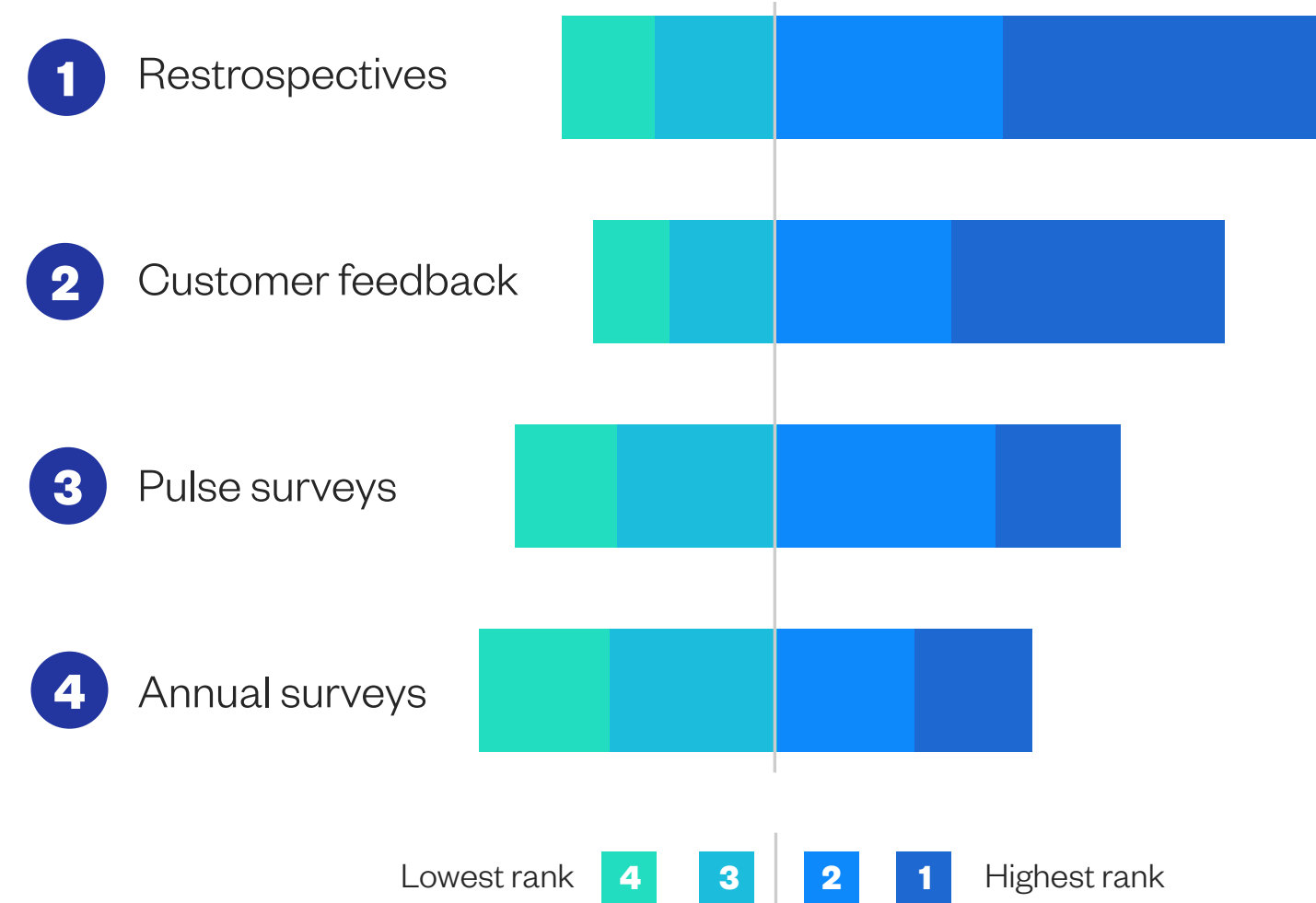| Metric	Percentage
Lines of code	70%
Pull requests	47%
Number of tickets closed	44%
Story points	42%
Other	2%

Qualitative metrics

In terms of methods to measure team performance, the trusty retrospective was the most popular, far outpacing customer feedback, pulse surveys, and cumbersome annual surveys.
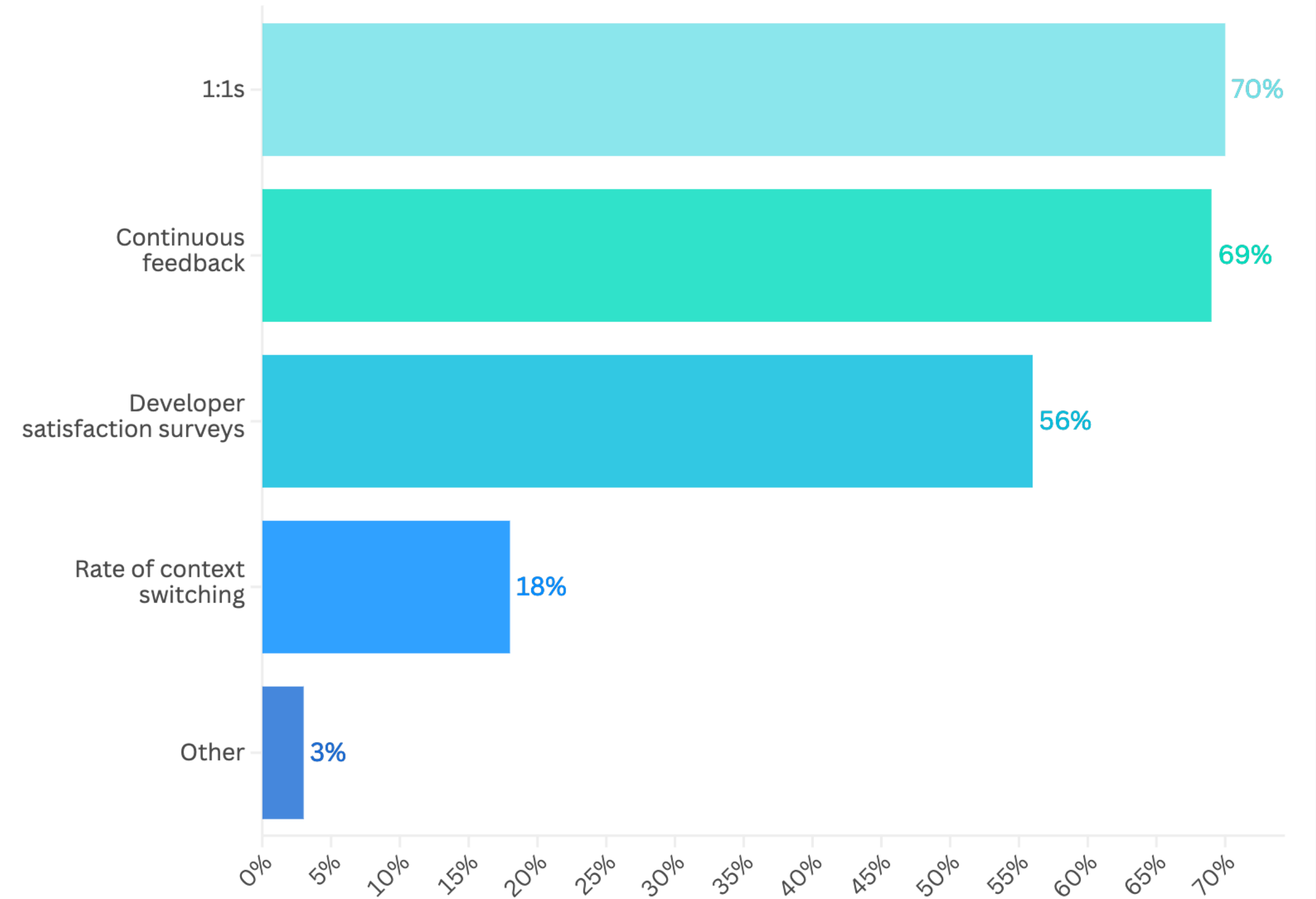
When it comes to the most useful methods to gauge **developer experience**, a combination of 1:1 meetings and a continuous feedback mechanism was favored by 70% of respondents. Developer satisfaction surveys were also cited as useful by 56% of respondents.

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

1 Restrospectives

2 Customer feedback

3 Pulse surveys

4 Annual surveys

Lowest rank 4 3 | 2 1 Highest rank

How to read this rank scale

Respondents were asked to rank their responses in order of importance, where 1 was the most important. They were asked to only rank items which were important to their engineering org.

The size of each block in the scale corresponds to the number of rankings in that position.

↙ Which methods have you found the most useful to measure developer experience?

Method	%
1:1s	70%
Continuous feedback	69%
Developer satisfaction surveys	56%
Rate of context switching	18%
Other	3%

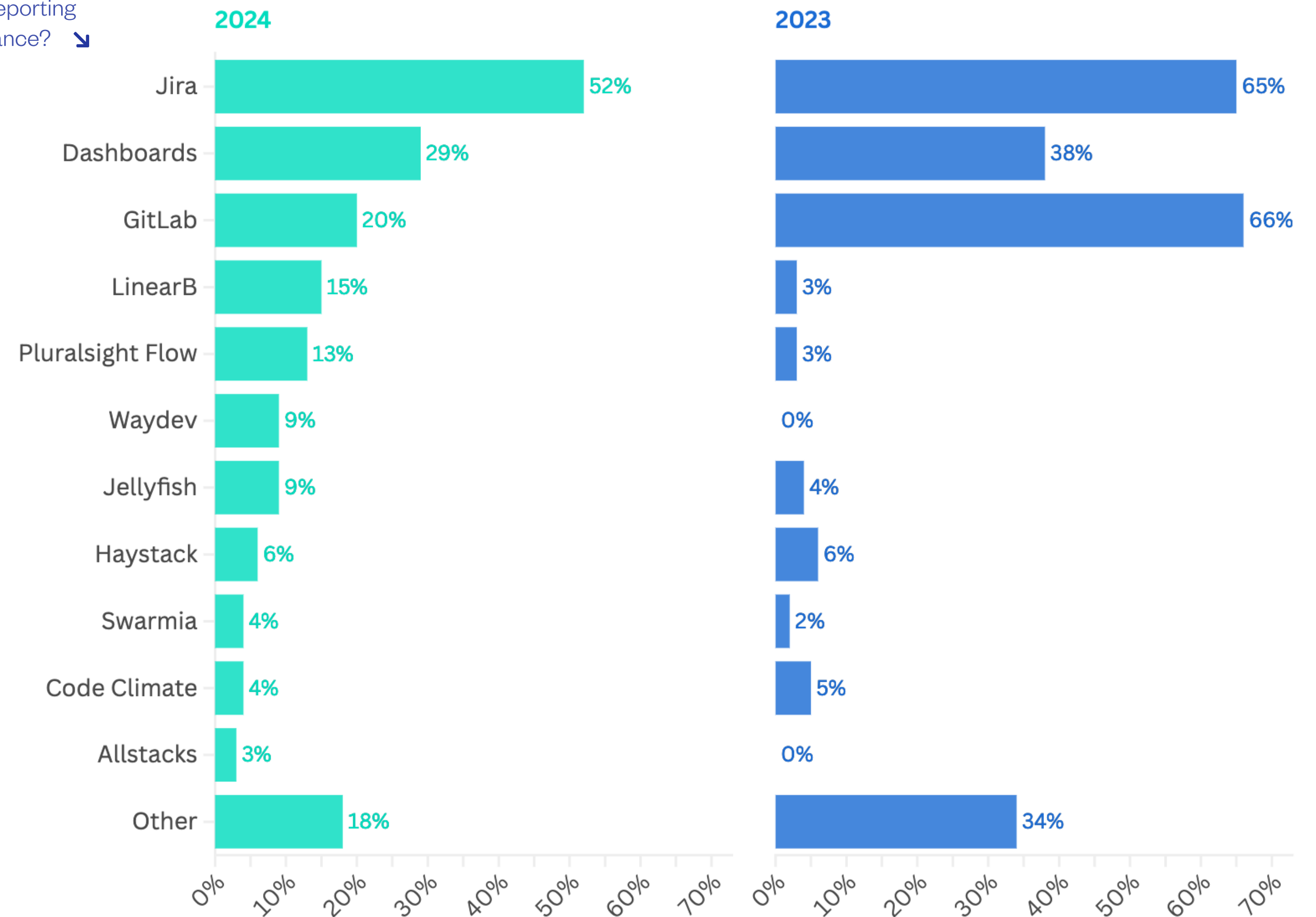0% 5% 10% 15% 20% 25% 30% 35% 40% 45% 50% 55% 60% 65% 70%

Popular tools and metrics

When it comes to reporting tools, there is plenty of choice for engineering managers. Despite a wave of new entrants to this market, there are still two clear standouts: JIRA (52%) and the humble dashboard (29%).

That being said, we saw a huge shift in reporting tool usage between 2023 and 2024. GitLab dropped from 66% usage to 20% this year, with a slew of best-of-breed tools filling the gap. LinearB rose from 3% to 15%, Pluralsight Flow from 3% to 13%, and Jellyfish jumped from 4% to 9%.

Waydev was a new entry to our list this year and is already being used by 9% of respondents to report on team performance. Of the 18% that use a different tool, reporting within GitHub, Plandek, Asana, and Sleuth all received honorable mentions.

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

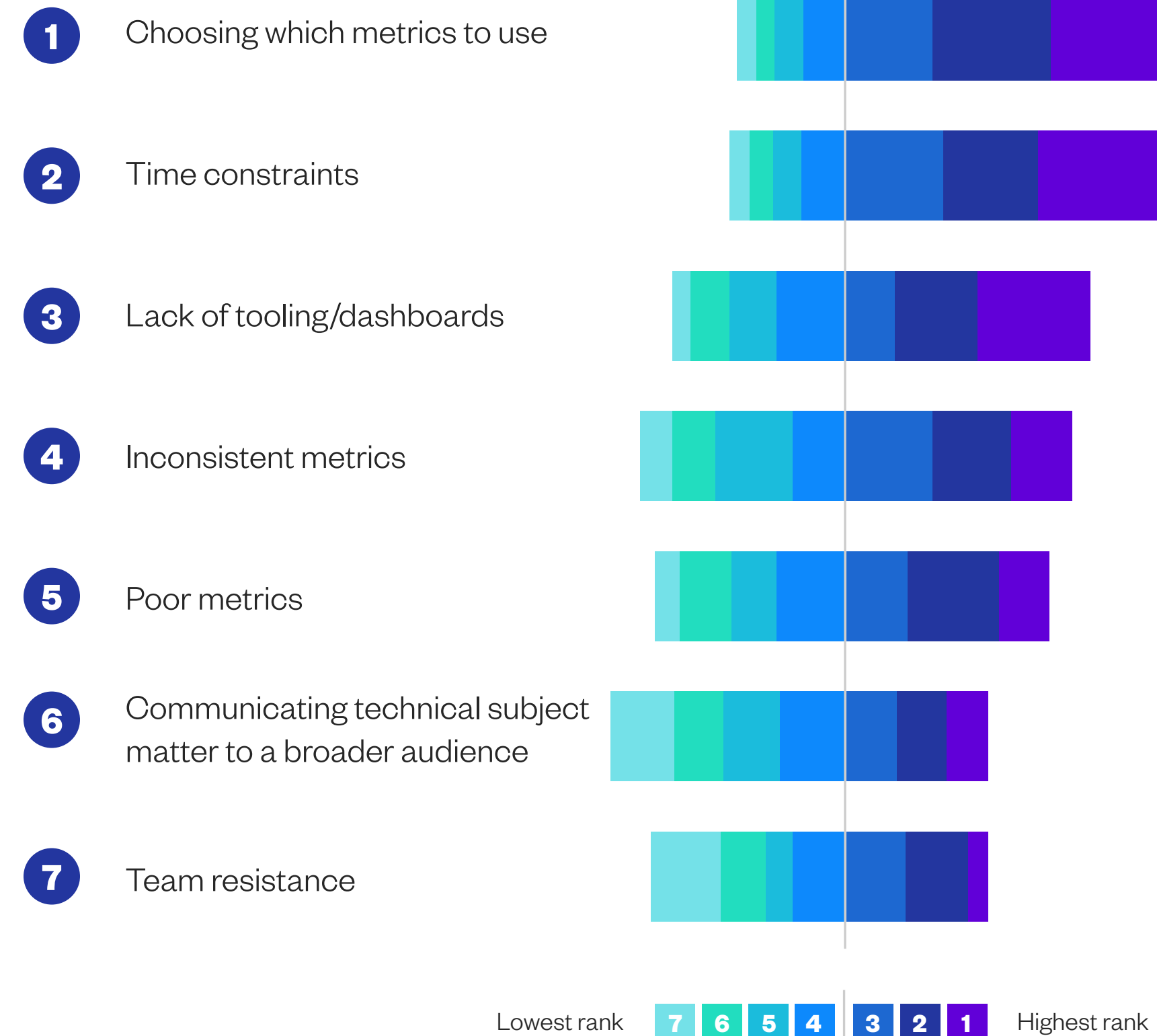| Tool	2024	2023
Jira	52%	65%
Dashboards	29%	38%
GitLab	20%	66%
LinearB	15%	3%
Pluralsight Flow	13%	3%
Waydev	9%	0%
Jellyfish	9%	4%
Haystack	6%	6%
Swarmia	4%	2%
Code Climate	4%	5%
Allstacks	3%	0%
Other	18%	34%

Choosing the precise metrics to measure team performance was the biggest challenge cited among respondents. This was closely followed by time constraints.

Interestingly, very few engineering leaders identified "team resistance" to being measured or "communicating technical subject matter to a broader audience" as particularly challenging.

The main challenge with reporting team performance is **choosing which metrics to use.**

What primary challenges do you encounter with reporting on team performance, if any? ↓

1 Choosing which metrics to use

2 Time constraints

3 Lack of tooling/dashboards

4 Inconsistent metrics

5 Poor metrics

6 Communicating technical subject matter to a broader audience

7 Team resistance

Lowest rank 7 6 5 4 3 2 1 Highest rank

How to read this rank scale

Respondents were asked to rank their responses in order of importance, where 1 was the most important. They were asked to only rank items which were important to their engineering org.

The size of each block in the scale corresponds to the number of rankings in that position.

The rise of DORA and SPACE

Many of these performance measures will look familiar to anyone who has come across DevOps Research and Assessment (DORA) metrics. First released in 2014, these metrics track how well organizations unite their developers and operations teams to deliver better software, faster.

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 to provide a more holistic picture of team performance.
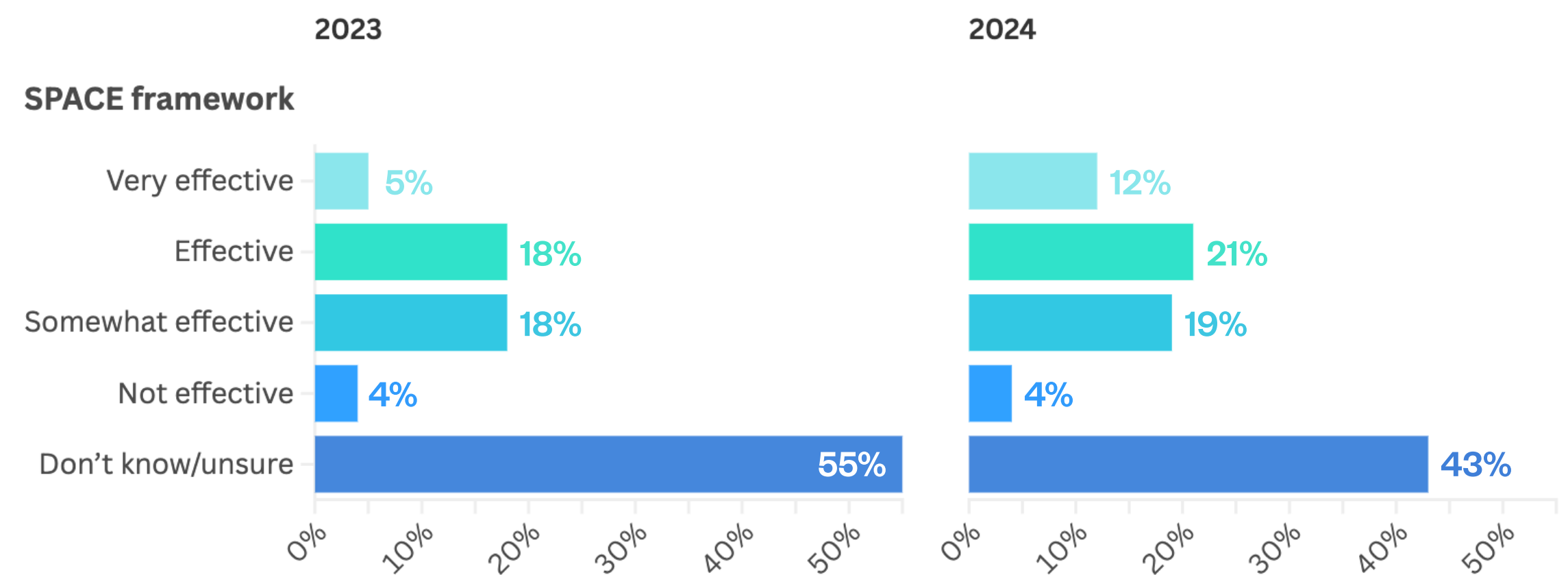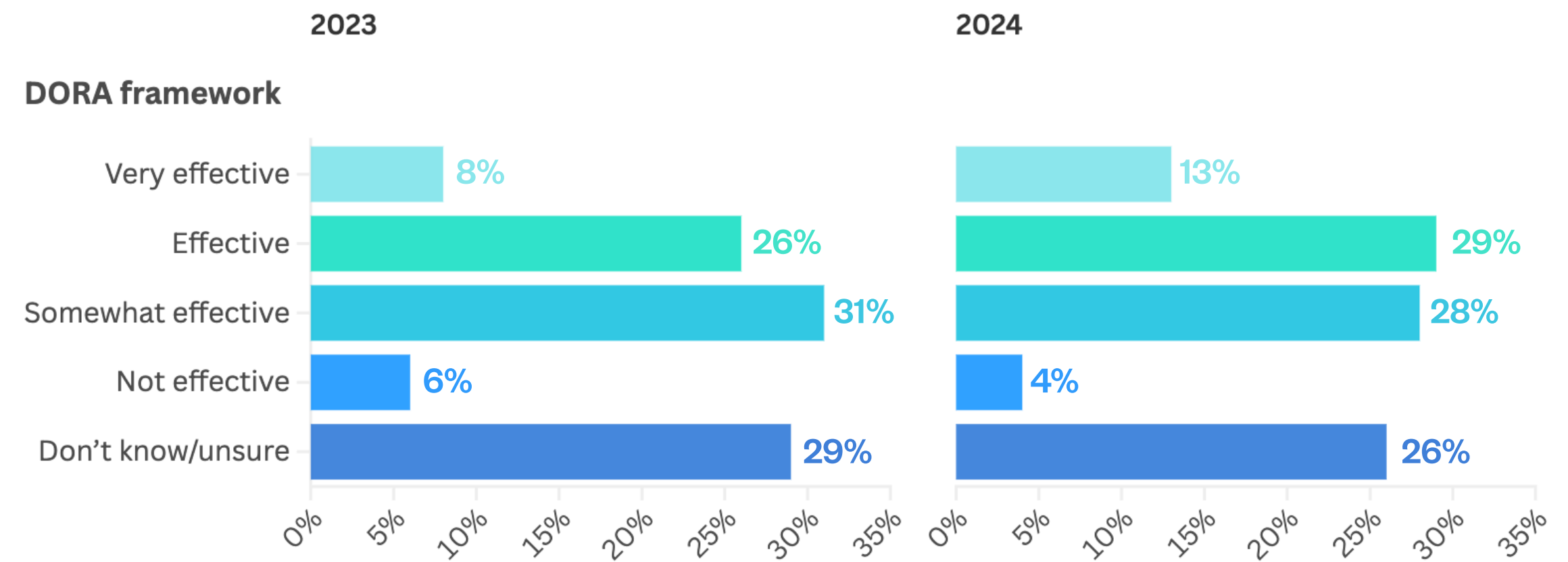
When asked about the effectiveness of DORA metrics in measuring performance, 42% said it was a very effective or effective framework, up from 34% last year. Still, 26% of respondents said they didn't know or were unsure about its efficacy. This is likely because they hadn't heard of or used the framework themselves.

Then, for the newer SPACE framework, 33% saw it as a very effective or effective framework, up from 23% last year. Still, 43% were unable to give an opinion on its efficacy, down from over half last year.

Just 4% of respondents saw one or both frameworks as ineffective.

How effective are the following at measuring team performance? ↓

DORA framework

2023

Very effective	8%
Effective	26%
Somewhat effective	31%
Not effective	6%
Don't know/unsure	29%

2024

Very effective	13%
Effective	29%
Somewhat effective	28%
Not effective	4%
Don't know/unsure	26%

SPACE framework

2023

Very effective	5%
Effective	18%
Somewhat effective	18%
Not effective	4%
Don't know/unsure	55%

2024

Very effective	12%
Effective	21%
Somewhat effective	19%
Not effective	4%
Don't know/unsure	43%

Learning and development
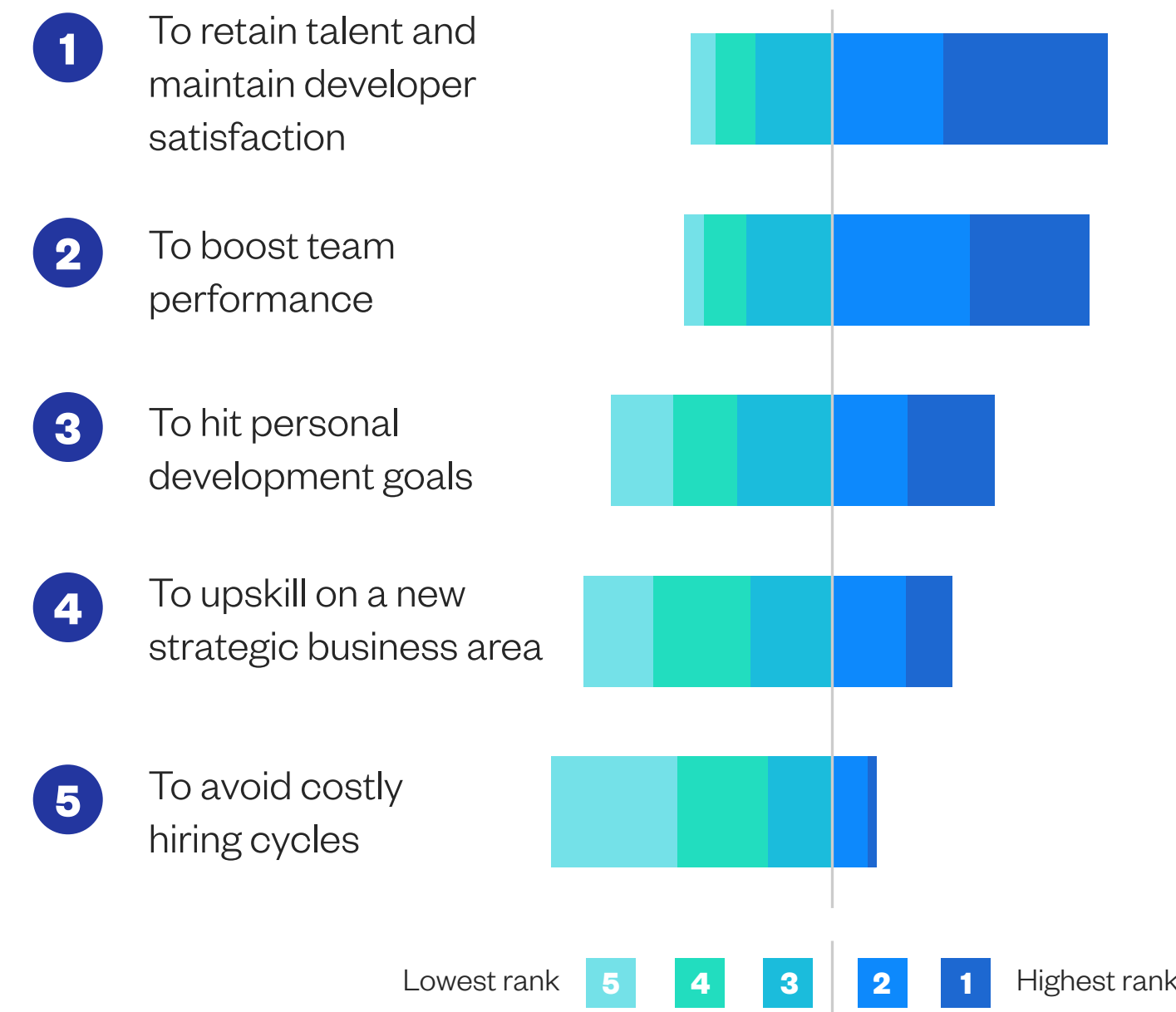
Learning and development

70%

of respondents say that **learning and development** for engineers is a priority for their organizations.

Learning and development becomes key as engineering leaders look to remove bottlenecks and increase velocity in challenging times.

70% of respondents said that learning and development is a current priority, mainly to retain talent and maintain developer satisfaction, closely followed by improving team performance.

Technical expertise is the most common development area focused on, with 78% looking to plug technical knowledge gaps by upskilling their teams. Developing leadership and strategy skills was a focus area for 41% of respondents, similar to communication and interpersonal skills, at 39%.

What are the reasons behind investing in learning and development? ↓

1. To retain talent and maintain developer satisfaction

2. To boost team performance

3. To hit personal development goals

4. To upskill on a new strategic business area

5. To avoid costly hiring cycles

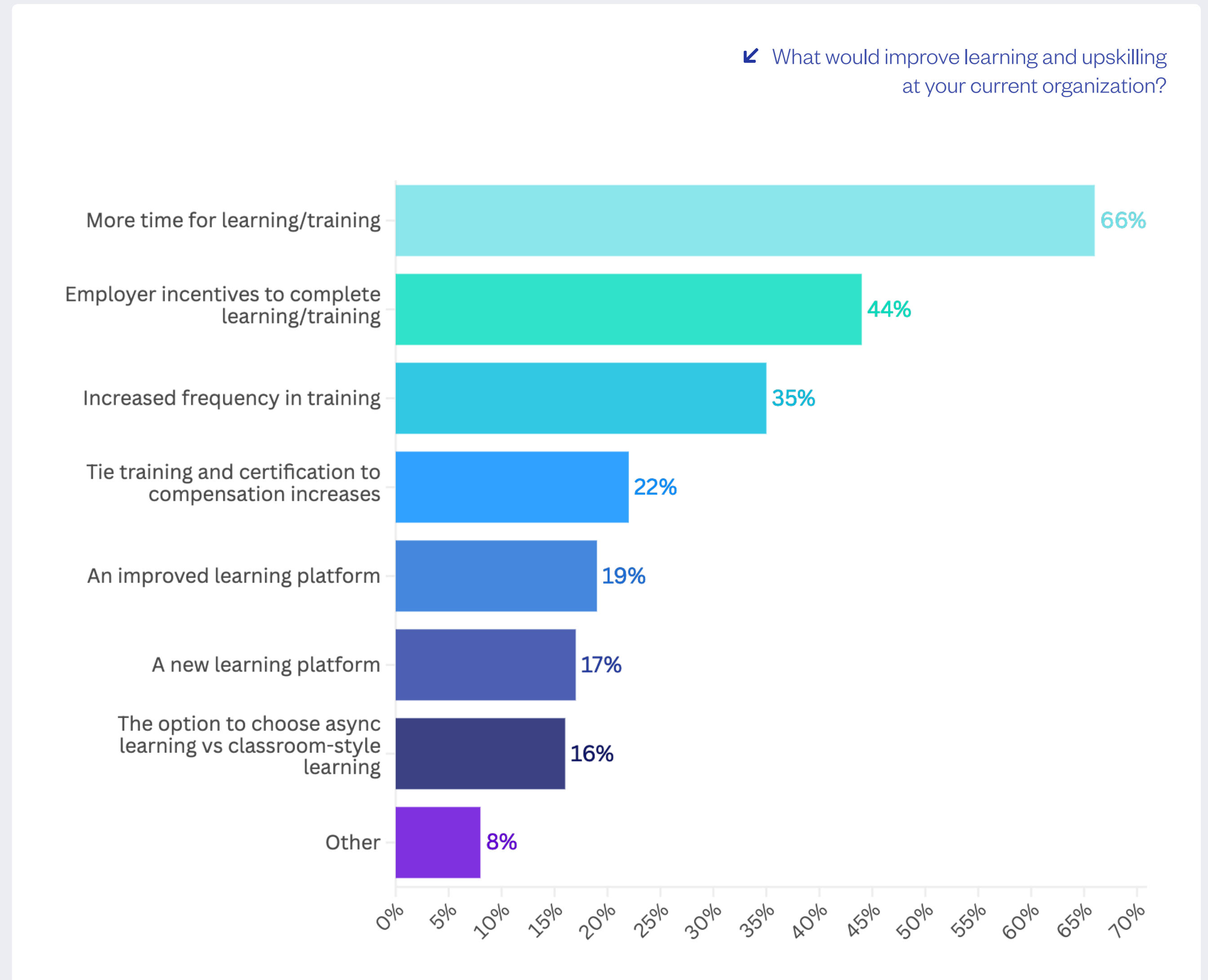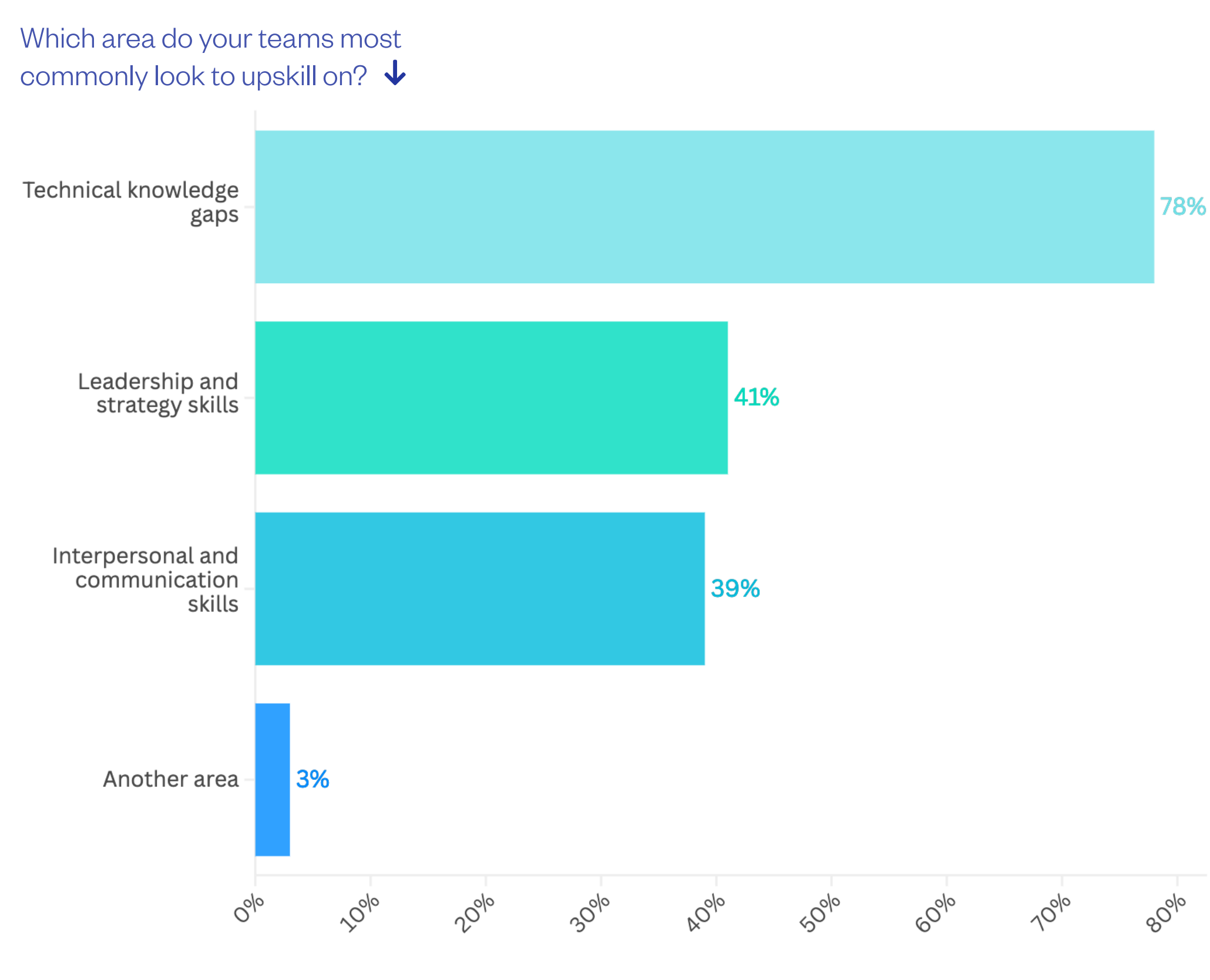Lowest rank 5 4 3 2 1 Highest rank

How to read this rank scale

Respondents were asked to rank their responses in order of importance, where 1 was the most important. They were asked to only rank items which were important to their engineering org.

The size of each block in the scale corresponds to the number of rankings in that position.

We all know that finding the time and resources to focus on learning and development can be difficult. 66% of respondents want more time to focus on learning and training, and 44% want their employer to incentivize such efforts, with 22% wanting it to be directly tied to their compensation. 35% crave more training opportunities, and 17% would like a better or new learning platform.

Which area do your teams most commonly look to upskill on? ↓

Area	Percentage
Technical knowledge gaps	78%
Leadership and strategy skills	41%
Interpersonal and communication skills	39%
Another area	3%

↙ What would improve learning and upskilling at your current organization?

Improvement	Percentage
More time for learning/training	66%
Employer incentives to complete learning/training	44%
Increased frequency in training	35%
Tie training and certification to compensation increases	22%
An improved learning platform	19%
A new learning platform	17%
The option to choose async learning vs classroom-style learning	16%
Other	8%

Methodology

This survey was conducted between October 18 – November 3, 2024, and we received 978 completed responses.

By job function

- 32% Manager of engineers
- 26% Advanced engineer (e.g. staff, principal, distinguished etc)
- 25% Manager of managers
- 9% CTO or equivalent
- 0% Currently not working
- 7% Other

By region

- 43% USA
- 29% Europe (excluding UK)
- 13% UK
- 5% Canada
- 4% Asia
- 3% Central and South America
- 2% Oceania
- 1% Africa

By company size

- 11% 0–49
- 31% 50–249
- 20% 250–499
- 8% 500–999
- 30% 1,000+