



# THE STATE OF AVAILABILITY REPORT

2022 Annual Report

# TABLE OF CONTENTS

<b>Welcome Note</b> .....	<b>04</b>
<b>Key Insights</b> .....	<b>05</b>
<b>KPIs</b> .....	<b>06</b>
SLA responsibility grows as companies grow larger .....	<b>10</b>
Error budgets are the most commonly tracked KPI .....	<b>11</b>
Larger companies show breadth of KPIs.....	<b>14</b>
Less than 20% of teams are tracking the incident timeline today .....	<b>16</b>
Groups with higher SLAs meet them more often .....	<b>20</b>
Average incident lifecycle is well above most SLA allowance .....	<b>21</b>
Often customers are catching the incidents first before internal tools .	<b>23</b>
KPIs—Key Takeaways.....	<b>25</b>
<b>Teams</b> .....	<b>26</b>
Most teams have shifted to the “We Build it, We Own it” DevOps culture .....	<b>30</b>
Larger companies are further behind on DevOps adoption.....	<b>32</b>
Most DevOps practices remain a ‘Want to Have’ .....	<b>33</b>
Organizational leaders believe their teams are at a much higher DevOps adoption rate.....	<b>36</b>
Leaders are also unaware of how much of their teams’ time is spent on monitoring.....	<b>37</b>
The higher the SLAs, the more time spent monitoring.....	<b>39</b>
Larger organizations spend more time on incident and infrastructure management.....	<b>40</b>

# TABLE OF CONTENTS

Teams - Key Takeaways.....	41
<b>Tools</b> .....	42
The majority of services are still not in the cloud .....	45
Organizations are still at the implementation stage of DevOps tool chains .....	46
Teams are managing huge amounts of monitoring tools .....	47
Teams with higher SLAs manage more tools per category.....	48
Tools - Key Takeaways.....	49
<b>Our Guidance</b> .....	50
<b>The Steps to Success</b> .....	53
<b>Survey Demographics</b> .....	58
Roles .....	59
Geography .....	60
Industry .....	61
Organizational Size .....	62
<b>Authors</b> .....	63
<b>About Moogsoft</b> .....	64

# Welcome Note

## Hi there!

Thank you so much for taking the time to read the inaugural State of Availability Report. Our goal when starting this research was to help engineering teams and leaders uncover insights and develop good practices for availability across three topics:

1. KPIs: what are the best key performance indicators for engineering teams to track and measure availability today?
2. Teams: what is the best way to structure engineering teams and understand daily responsibilities and challenges?
3. Tools: what tools and architecture are being used by engineering teams today and how are they planning for the future?

Availability is a well-established KPI and teams are giving up a lot of time to maintain it. But is the expense worth it?

Based on research from 1,900 respondents, we say: “no”:

- Teams are not tracking 66% of their downtime and lack KPI coverage from incident detection through to resolution, with customers frequently reporting incidents before monitoring tools catch them
- Teams are spending too much time monitoring, and too much money on too many tools, and yet still not seeing good results in the availability of their systems
- All this time spent on incident management is detracting from the effort teams want to spend on making long-term improvements to stability and increased throughput (DevOps)

Read on to learn more! Each topic (KPIs, teams, tools) has its own section with findings, details, and key takeaways. We also wrap up the report with practical steps you can follow to focus and scale your team, their time, and your infrastructure.



**Happy Reading,**

**~Minami Rojas, VP of Growth and Marketing at Moogsoft**



**“Despite huge investment in monitoring, availability outcomes are not where they should be. The data points to a burning need for teams to do two things.**

**Firstly, introduce an intelligent-correlation layer—aka AIOps. The data from monitoring should be correlated into a tight, actionable incident-set. Without this, the data is expensive and arguably worthless.**

**Secondly, consolidate tool usage in this observability layer. The savings are likely greater than the investment in AIOps, and your outcomes will be better!”**

~Phil Tee, Moogsoft CEO



**“Customers see availability as a given and without it, you’re losing them. But too often teams are spending too much money, time, and energy sprinting to stand still here and it’s to the detriment of their ability to invest in the things that will set them free and assure the longevity of their organizations.”**

~ Helen Beal, Strategic Advisor

# Key Insights



**“By implementing robust and intelligent monitoring, organizations can unlock cost benefits. With that additional confidence, the error budget can become an asset that can be invested in improving services and paying down technical debt.**

**I found the numbers very telling. The fact that SLOs, and the SLIs that drive them, are far less prevalent than SLAs, suggests that organizations are ‘backing into’ SLAs. I strongly doubt this is due to a lack of monitoring coverage, but an inability to extract actionable data from the wealth of telemetry at their disposal.”**

~ Richard Whitehead, Chief Evangelist at Moogsoft



**“As techniques for measurement, reporting, and management of services have evolved, one thing remains paramount—if you are not available you are of no value.**

**Being ahead of your SLA and making the difficult decisions before it’s too late, based on in-place error budgets that signal risk, is how service owners can keep customer confidence high.”**

~ Chris Boyd, SVP Engineering at Moogsoft

The background is a dark blue gradient with several large, overlapping, wavy shapes in a lighter blue color. These shapes are composed of many thin, concentric lines, creating a ripple effect. There are also several black, rounded rectangular shapes scattered across the image, some of which are partially obscured by the wavy patterns. The overall aesthetic is modern and digital.

**KPIs**

**Key performance indicators (KPIs)** are the metrics that tell us how well we are doing our jobs. This research has baselined the leading KPIs—which engineering teams are using to track and measure availability—and explored how teams and organizations are using service level agreements (SLAs) to manage customer experience.

### **Most teams are breaching their SLAs**

Regardless of company size or the KPIs they track, teams and organizations are regularly breaking their availability promises to their customers. Teams with higher SLAs meet them more often than those with lower SLAs. This is to be expected and may be self-selecting: if teams care about SLAs, they set more exacting standards, are more likely to be doing something about them, and are better able to forecast their ability to meet them.

### **Higher availability leads to higher organizational performance**

Our data show that it's well understood that poor performance against SLAs leads to poor customer experience—customer reviews and Net Promoter Score (NPS) are the most common indicators that there is a problem. When customer experience is poor, organizational performance and employee experience both suffer. Leaders must find ways to help their teams meet their SLAs.

## **Error budgets are the leading availability KPI**

Error budgets define the maximum amount of time that a technical system can fail without the contractual consequences of an SLA. More recently, error budgets have been tightly associated with site reliability engineering (SRE) and service level objectives (SLOs).

They are internally agreed between teams to describe tolerance for user pain or frustration, which can trigger policy decisions such as “no more production releases until we stabilize”. These metrics were the most commonly used in small and medium companies and those with more aggressive SLAs. Mean time to recover/restore (MTTR) and security breaches are the next most used.

## **Teams with fewer, more meaningful KPIs and higher SLAs perform better**

It's possible there is a limit to the amount of information that is actually helpful. The law of diminishing returns suggests there is a clear point at which information overload becomes a problem and leads to ineffective decision-making. Decision fatigue can result in poor choices as individuals make mental shortcuts in their decision analysis. Both these factors suggest we need fewer, more meaningful metrics.

Furthermore, KPIs should have decisions attached to them for different outcome scenarios. When teams are working in an experimental way, they'll be creating hypotheses for the work they are doing and continually seeking feedback from both customers and systems on the impact of their work.



**Teams are not tracking 66% of their downtime**

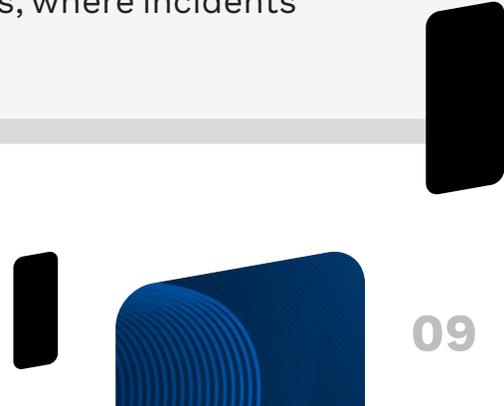
Most teams focus on MTTR, but few on mean time to discovery (MTTD). The average MTTR is thirty minutes whereas the average MTTD is sixty minutes, so teams are missing the opportunity to track 66% of their incident downtime.

They need to know these measures to track SLA performance and to justify investments that will improve availability long-term. Incidents are unplanned work that detract from innovation and improvement.

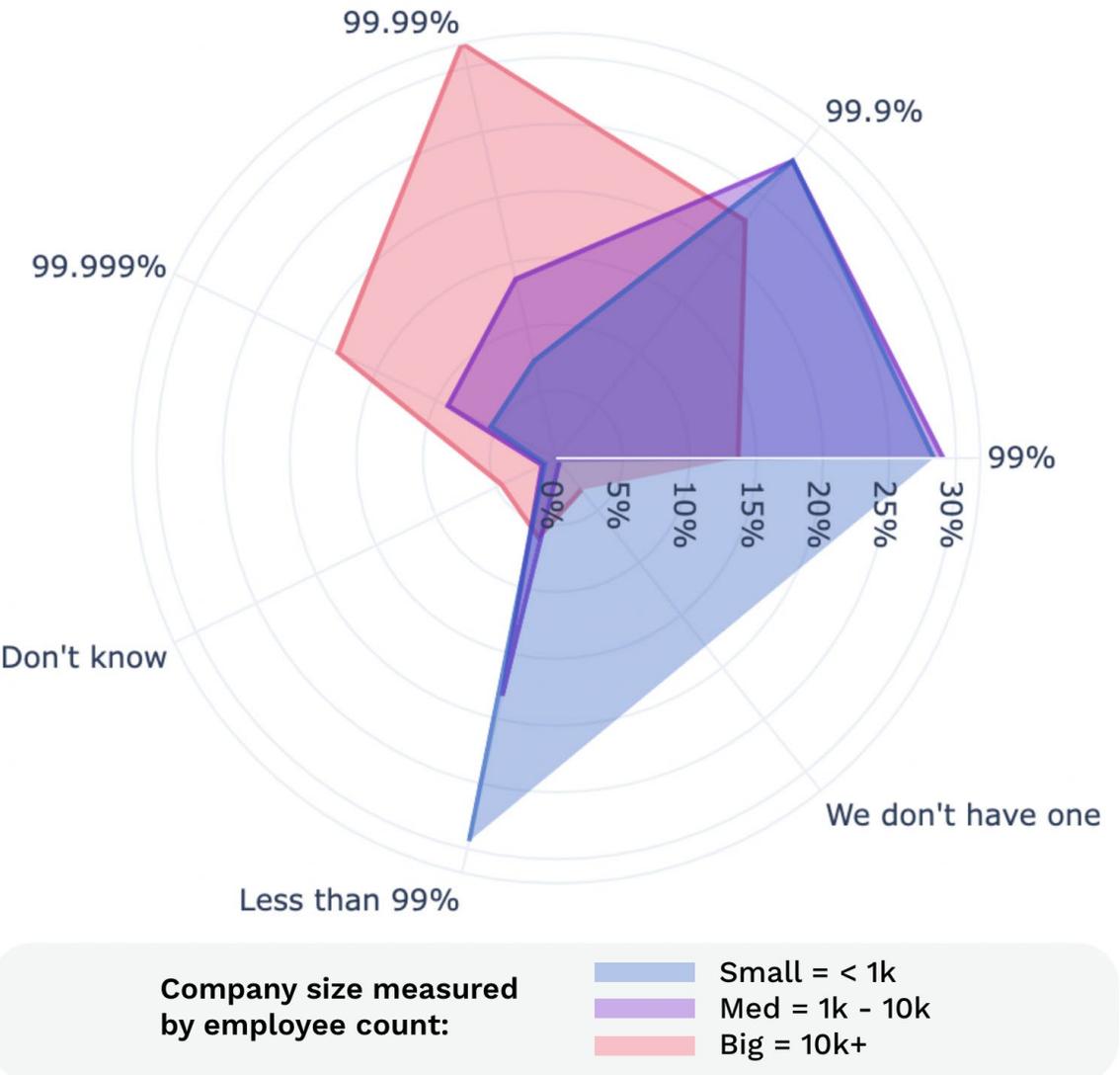
**Customers are reporting issues half of the time**

Tools are catching issues before customers are flagging them about half the time, despite huge investment in monitoring tools. Teams are spending most of their time monitoring—likely over monitoring—with no real result. This is obviously a poor customer experience and teams need to find ways to catch issues first. The problem is that having so many monitoring tools means that teams are deluged in data; more data than a human has sufficient cognitive capacity or time to handle.

In addition, many teams are transitioning to distributed architectures such as microservices, or service-oriented architectures, where incidents caused by unknown unknowns are more common.



# SLA Responsibility Grows as Companies Grow Larger



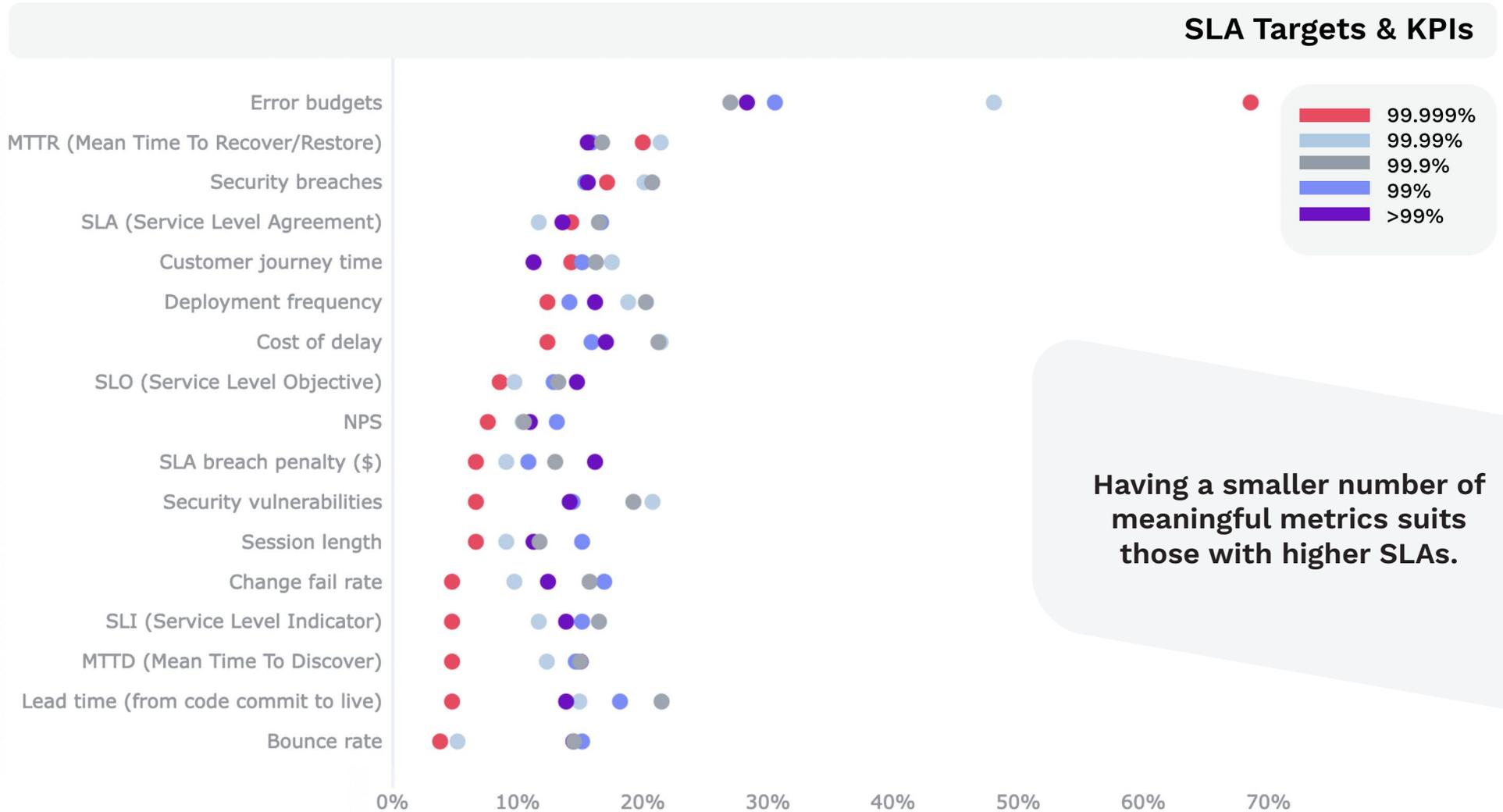
The bigger the company, the higher the SLA. You might imagine that smaller companies don't have SLAs but in our data, only 0.3% of respondents said they didn't have one.

We also learned that higher SLAs are more prevalent in organizations where customers are primarily external and consumers (i.e. B2C not B2B organizations). We know that as companies grow and mature they develop higher-level capabilities for incident management, have dedicated IT Operations teams, and centralized platforms and services.

Having this focus on availability enables teams to support higher SLAs, and higher SLAs are required as customer numbers grow—particularly if they are external. Consumers are notoriously intolerant of slow response times or broken services.

## KPIs

# Error Budgets are the Most Commonly Tracked KPI



Having a smaller number of meaningful metrics suits those with higher SLAs.

Error budgets came up as the most popular KPIs for teams where Support Engineer was the most reported job title, regardless of SLA. And those with five nines (99.999%) rely on this KPI more than any other.

This group focuses on a handful of key metrics, whereas the others track most of them. It seems having a smaller number of meaningful metrics suits those with higher SLAs, who are also better at meeting their SLAs.

Error budgets are also the bluntest KPI. The MTTR/D next level set of incident timing breakdown is critical because knowing why targets have been missed is more important than knowing that they have.

The next two most commonly used KPIs are MTTR and “security breaches”. However at the leader/organizational level, “cost of delay” is the top metric reported, reflecting their focus on revenue/profit and strategy delivery.



*“Bill Gates said: ‘New technology has less impact in the first two years than anticipated and more in the first ten years.’ It may be that SRE has gone from being novel to mainstream to the point that it’s not even being labeled differently in some organizations. Companies need to acknowledge and act on these truths:*

- 1. You are over-monitored—you have lots of tools that could be consolidated to fewer ones and still have 100% coverage. For example, do you need a separate event and metrics tool?*
- 2. You are under-available—because you have focused on coverage you have not focused on actionability. Too much dumb data acquisition and not enough smart data analysis.”*

*~Phil Tee, Moogsoft CEO*

# A NOTE ON ERROR BUDGETS

It's important to note that error budgets are a methodology and not the metric or KPI itself.

An error budget is a threshold by which an action is taken to improve the customer experience if it's exceeded. For example, if a service level agreement (SLA) specifies that systems will function 99.99% of the time before the business has to compensate customers for the outage, that means the SLA states that systems can go down without consequences for 52 minutes and 35 seconds per year. A realistic error budget for this scenario would be 4 minutes of downtime per week, because if this is exceeded then the team knows they are on track to breach the SLA and should therefore invest in system improvements.

When the error budget is exceeded, the team will investigate each contributing downtime event, and any correlation between events, to see what can be done to improve availability. One outcome might be that performance is found to be impacted and that by improving performance, the response time service level indicator (SLI) is met again and availability is increased.

The term was popularized by the site reliability engineering (SRE) movement that began at Google. We were surprised to see this figure so heavily in the data since only 2.4% of our respondents identified their role as DevOps engineer/SRE (individual or team lead). The leading job role in this research is support engineer (38%).

The practice of error budgeting is, of course, tightly connected with SLAs, SLOs (service level objectives), and SLIs. We discovered that the majority (36%) use both SLOs and SLIs, with 13% planning to use them. We also found that 26% of respondents have only SLOs and 23% only have SLIs.

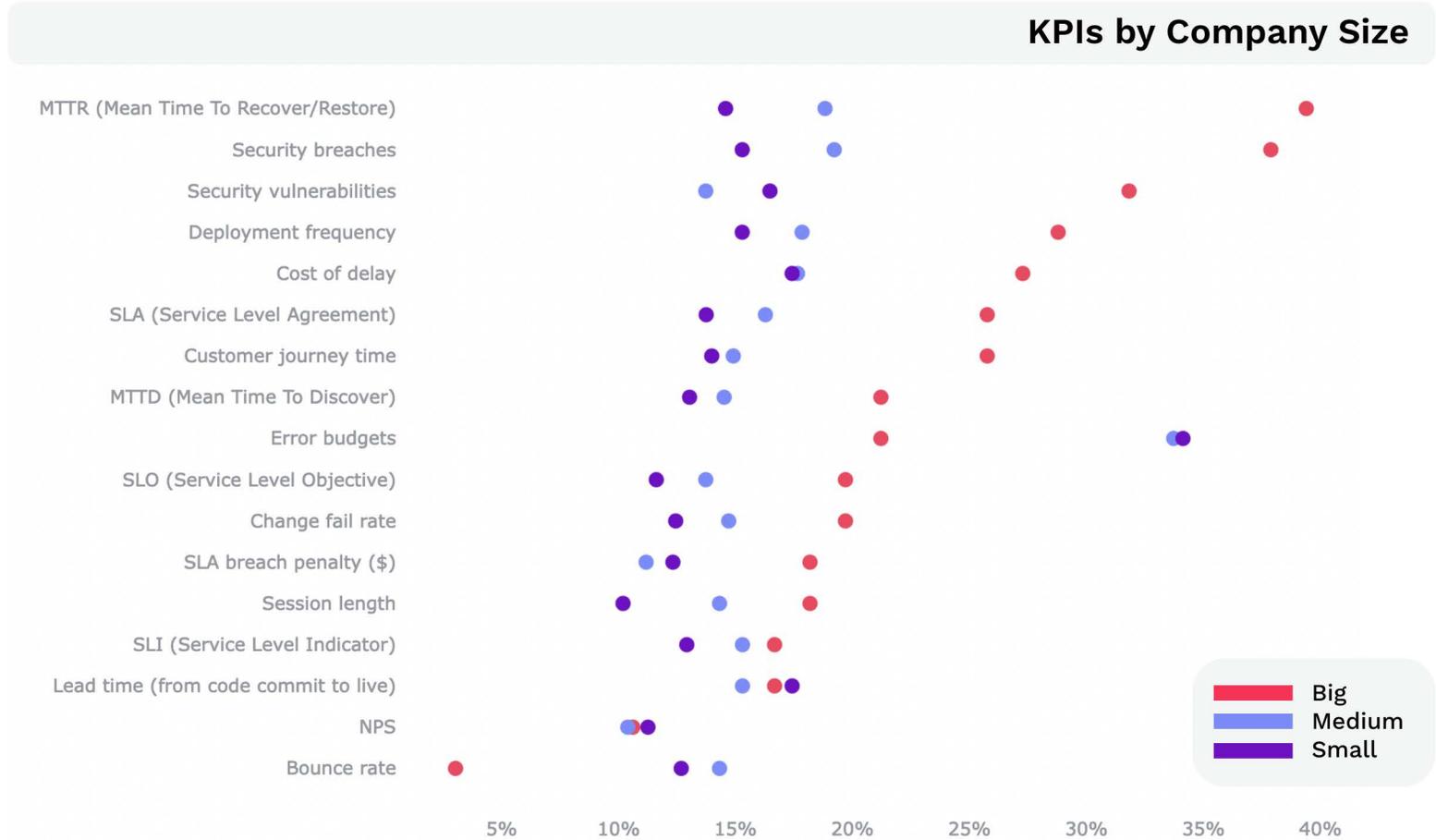
- SLA: usually contractual between two parties
- SLO: this internal goal for a team is designed to prevent the team from breaching an SLA as it's more aggressive than a contractually committed SLA; in our example scenario where the SLA error budget might be no more than 4 minutes of downtime a week, the SLO might be 3 minutes per week
- SLI: essentially an indicator of an SLO's performance; for example, a high number of downtime events indicates that the SLO for "availability" is at risk
- Error Budget: a methodology in which different budgets are tracked to know when user experience is likely to be or is already impacted, so teams can choose to act on that reality rather than shipping features or focusing on other work

A conclusion to draw from this is that SRE practices are already widespread, perhaps being used in roles that aren't using the SRE title, or at least that awareness of their usage is well recognized.

## KPIs

# Larger Companies Show Breadth of KPIs

Our research shows that larger companies are more likely to use availability KPIs and are broader in their usage. You can also see here that the small and medium companies predominantly use an error budgets KPI. The leading KPI for the larger companies is MTTR, closely followed by security breaches, then security vulnerabilities.





*“I think larger companies take longer to make any changes, especially innovative ones like using new technologies or new practices. These choices are risky and they’re willing to wait for others to prove the value and worth before investing their time getting hundreds of people to do things differently.*

*And larger companies have larger demands, meaning more volume and velocity, which leads to more ‘black swan’ events and more availability challenges. They also have more change events, which are probably the leading cause of incidents, so no matter how few or meaningful their KPIs are, I think they would still be challenged due to the sheer scale they operate at.”*

*~Eric Brousseau,  
Moogsoft VP of Product*

Perhaps these larger companies are slower in their uptake of SRE practices. Perhaps smaller companies are more commonly using the most recent progressive ways of working such as SRE. It’s easier for small companies to make changes because it takes fewer people to convince and train, and they’re incentivized to find ways to outpace larger competitors, so are prepared to take more risk.

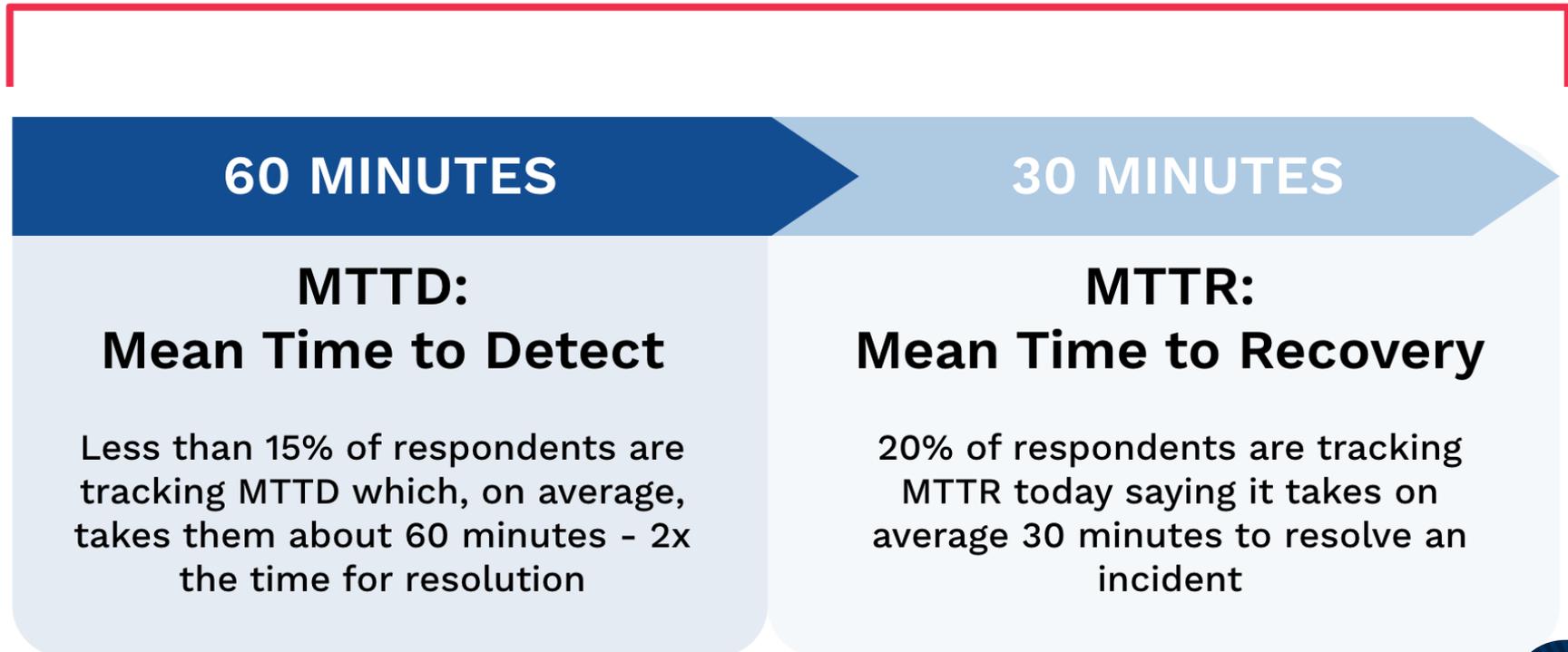
Larger companies may also have high brand awareness and higher potential negative customer impact with outages, making it critical they focus on the resolution time if an incident happens. Smaller companies are usually more proactive and watch error budgets for early signs of issues.

Generally, we can see larger companies using a smaller number of KPIs and we know that larger companies have higher SLAs, and those with higher SLAs are more likely to meet them—so it could follow that a tighter focus on a smaller number of KPIs contributes to better availability.

# Less Than 20% of Teams are Tracking the Incident Timeline

Less than 15% of respondents are tracking their mean time to detect. That adds up to an hour lost every time there's an incident.

## 90 minutes for average incident lifecycle



Discovering there's an issue takes twice as long as resolving the issue. Furthermore, 80% of respondents aren't tracking their MTTR. The data shows that the average incident lifecycle is ninety minutes and most respondents are missing their SLAs. That's a lot of unplanned work that's not visible. Peter Drucker reputedly said, "If you can't measure it, you can't improve it."

Our data suggest a smaller number of KPIs correlates with higher availability and MTTR leads. But choose KPIs carefully—adding MTTD means that teams can see the end-to-end incident lifecycle and prioritize ways to reduce it.

Reducing this unplanned work makes more time available for teams to work on: new innovations to improve customer experience; new features or platform improvements for performance and reliability (e.g. paying down technical debt, investing in modern, sustainable infrastructure); or automating toil to release even more capacity.



*"Availability is customer experience, confidence, and trust. Communication is very important so that people understand the quality and service we are delivering. Having the right behavior means having the right tools and data. If we don't act appropriately, you can't make the best use of it."*

*~Mike McGibbney, SVP SaaS,  
Körber*



*“There are complex ways to calculate MTT(X) but we know from this research that limiting the amount of KPIs has a direct relationship with achieving higher levels of availability. We recommend then that teams focus on MTTD and MTTR—these are the most meaningful KPIs and the easiest to measure. Reducing the time spent dealing with an incident releases time to spend on improving platforms and services and reducing the volume of incidents moving forward.”*

*~Eric Brousseau, Moogsoft VP of Product*

## A NOTE ON MTTD/MTTR

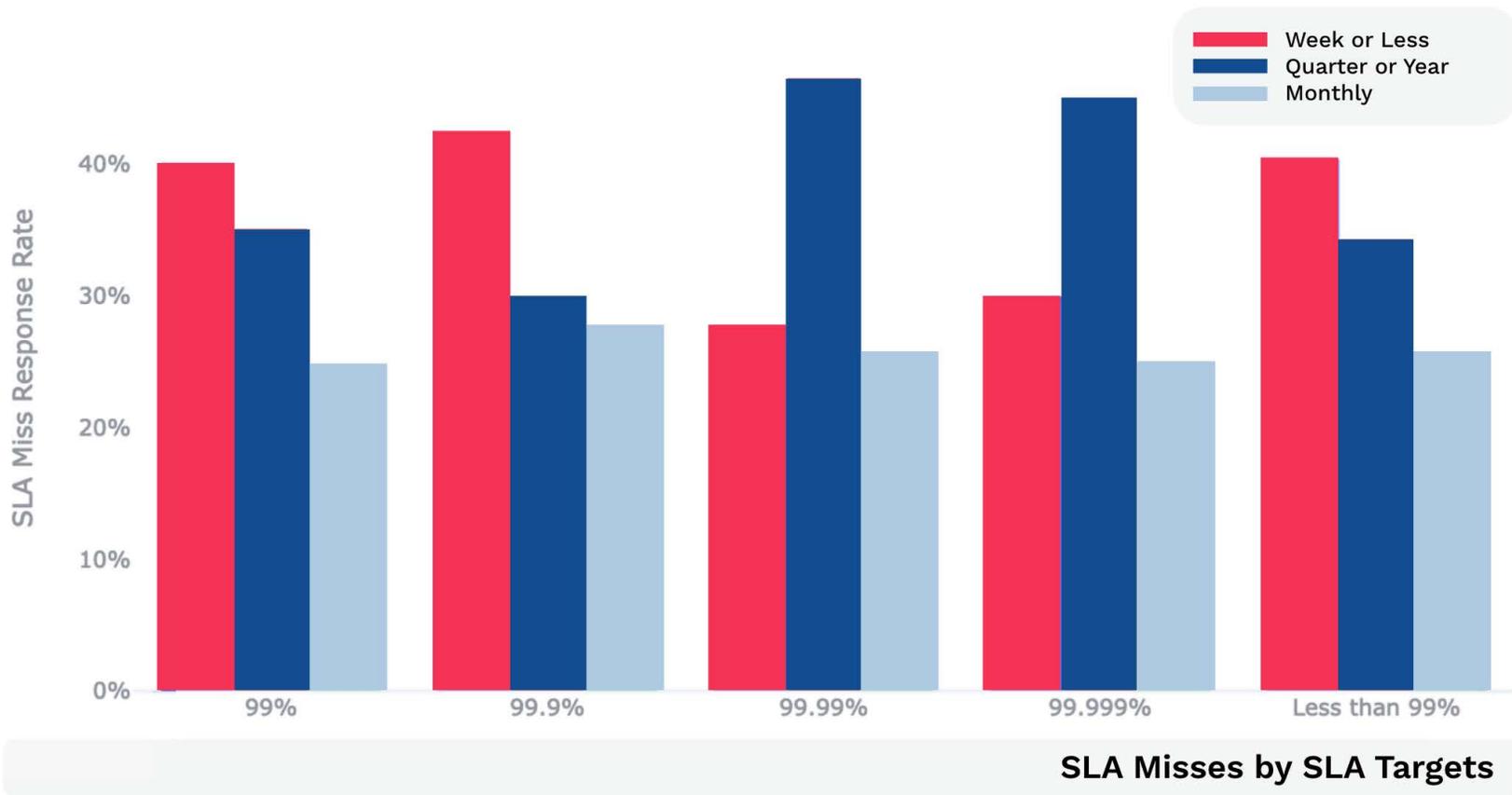
We asked about mean time to detect (MTTD) and mean time to recovery (MTTR) in this year’s research as two leading incident management KPIs strongly associated with availability. You can find a table of other “mean time” incident metrics to consider on the next page. Our top tips for KPIs are:

1. Make sure you understand the definitions of all the KPIs available to you—and that understanding is shared across your team and organization
2. Understand how the available KPIs align with business goals (short and long term)
3. Pick a small number of KPIs and focus hard on them—ensure they are instrumented so teams don’t spend time looking for them, calculating them, and reporting on them—they need to be available on at least a day-to-day basis
4. Use KPIs actively to identify and measure improvement opportunities that result in more time being made available for teams to invest long-term in customer experience
5. Look for instrumentation and tools that do more than just monitor and alert—look for tools that provide insights that are hard for a human to find on their own
6. Accept that tools need constant maintenance—they need to be correctly configured, and tweaked as conditions around them change—there is an overhead with most tools (and/or find a tool that monitors the monitoring i.e. AIOps)

Acronym	Short for	Definition
<b>MTBF</b>	<b>Mean time between failures</b>	Measures the ability of a system or component to perform its required functions under stated conditions for a set amount of time; the elapsed time between system failures during everyday operations.
<b>MTTA</b>	<b>Mean time to acknowledge</b>	The average time it takes from when an alert is triggered to when work begins on the issue.
<b>MTTD</b>	<b>Mean time to detect (discover)</b>	The time between the onset of an incident and its discovery. Or, the time spent discovering the cause of an incident, prior to starting to implement the repair.
<b>MTTF</b>	<b>Mean time to failure</b>	The average amount of time a defective system can continue running before it fails. Time starts when a serious defect in a system occurs, and it ends when the system completely fails. MTTF is used to monitor the status of non-repairable system components and analyze how long a component will perform in the field before it fails.
<b>MTTR</b>	<b>Mean time to recover (restore)</b>	The time spent getting an application back into production following a performance issue or downtime incident. This includes the full time of the outage—from the time the system or product fails to the time that it becomes fully operational again.
<b>MTTR</b>	<b>Mean time to repair</b>	The average time it takes to repair a system including both the repair time and any testing time.
<b>MTTR</b>	<b>Mean time to resolve (resolution)</b>	Mean time to resolution addresses the time required to fix a problem and to implement subsequent “cleanups” or proactive steps designed to keep the problem from recurring. Teams should address both of these tasks before they can declare an issue resolved.
<b>MTTR</b>	<b>Mean time to respond</b>	The average time it takes to recover from a product or system failure from the time of the first alert. This doesn't include any lag time in the alert system.

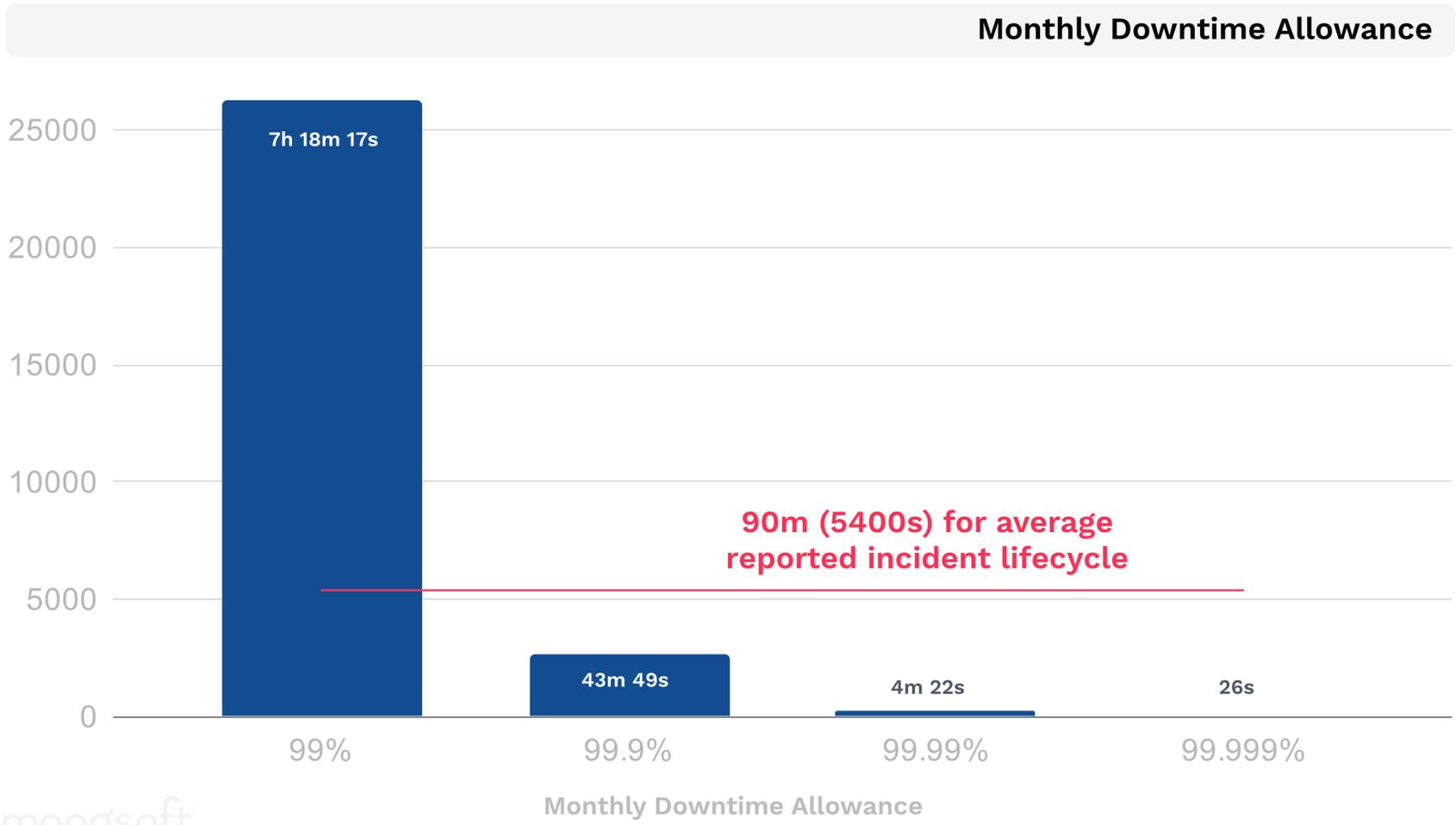
# Groups with Higher SLAs Meet Them More Often

Even though higher SLAs are harder to meet, our research shows that those teams and organizations with them (four or five nines) are missing them less. But on average 25% are missing their SLA on a monthly basis.



# Average Incident Lifecycle is Well above Most SLA Allowance

Our respondents have given us data that allows us to calculate how far they are from meeting their SLAs: how many nines, their frequency of missing them, and their MTTD and MTTR.



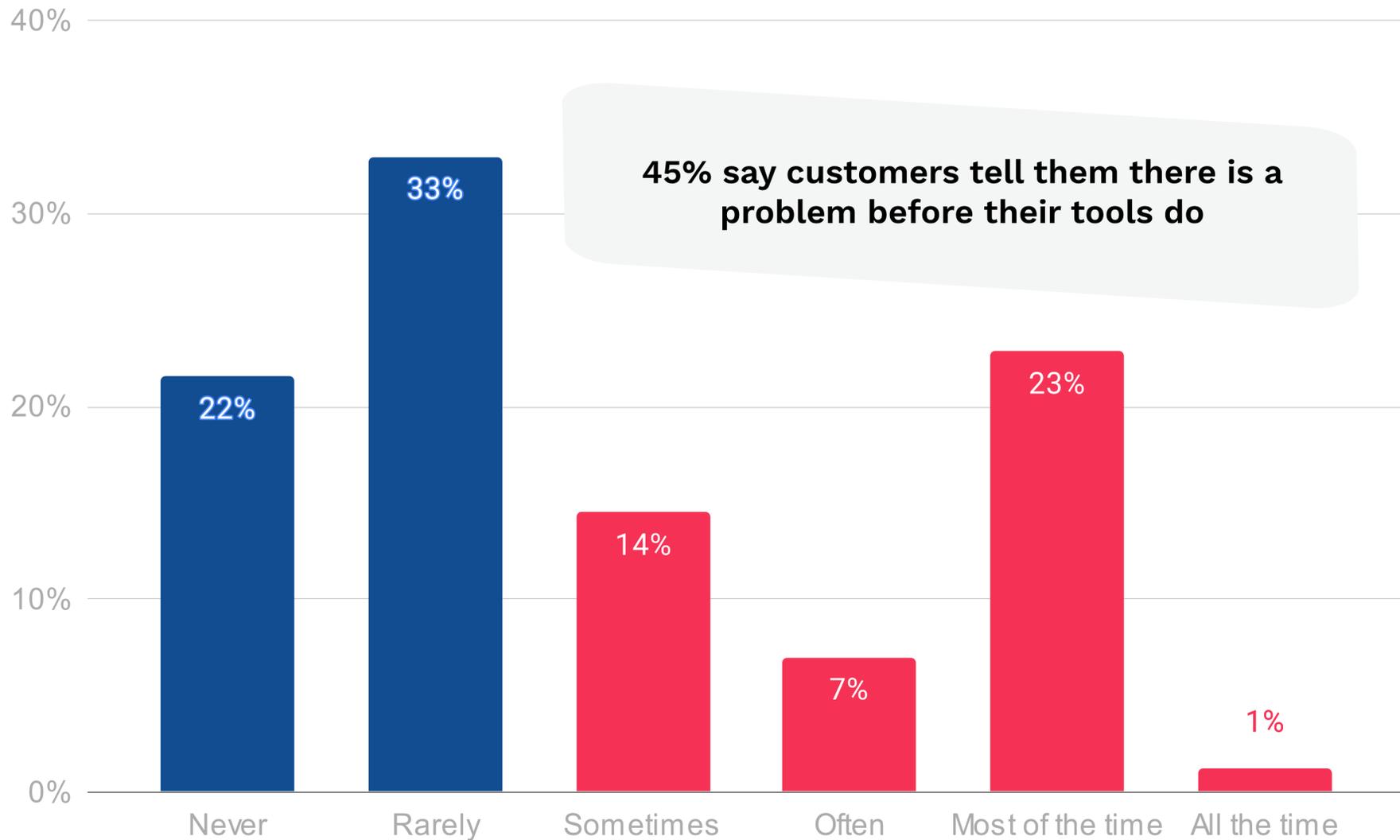
Average MTTD and MTTR combined is at ninety minutes so all of the respondents with three nines or more will be breaching their SLAs when they are having monthly incidents. This means they'll be blowing their error budgets too (that top-tracked KPI).

This leads to poor customer experience (which leads to poor reviews, referrals, and churn), and also indicates poor employee experience (which leads to poor employee retention, burnout, and lack of engagement). The combination results in poor organizational performance, and puts the ability of the organization to sustain itself at risk.

Quite simply, these teams and organizations need to find ways to fix incidents faster and find ways to reduce the number of incidents occurring in the first place.

## KPIs

# Often Customers are Catching Incidents First, before Internal Tools



45% say customers tell them there is a problem before their tools do



*“There is compelling evidence of the “over monitor/under available” syndrome here. Ultimately if customers are telling you an incident has happened it’s a failure! If customers routinely catch outages for you, your monitoring strategy has failed. It’s as simple as that. With a balanced monitoring and intelligent correlation strategy, this can be prevented and should surely be a #1 priority.”*

*~Phil Tee, Moogsoft CEO*

With 45% of our respondents reporting that their customers are telling them there’s a problem before their tools do, there is clearly work to be done to optimize customer experience.

As we’ll show in the Tools section of this report, teams have plenty of monitoring tools, so it’s puzzling that the tools apparently aren’t doing their job and alerting teams to the problem.

We asked our respondents about the consequences of poor customer experience and the ones that were most commonly cited were: poor reviews, pressure being applied by leaders, and a drop in Net Promoter Score (NPS). Organizational leaders were also concerned about a drop in revenue.

The results of poor customer experience are very clearly recognized, so leaders and teams need to find ways to make sure they catch incidents before their customers do. We suggest teams use a metric to track this, for example, “customer reported incident” (CRI) as a ratio/percentage of all incidents.

# KPIs Key Takeaways

**Availability is a serious problem for most teams and is negatively impacting customer experience**

**Fewer KPIs lead to higher performance—demonstrated by the percentage of SLAs met**

**Focusing on MTTD could reduce the cost of delay by 66% for most teams**

The background is a dark blue gradient with several large, overlapping, wavy shapes in a lighter blue color. These shapes are composed of many thin, concentric lines, creating a ripple effect. Scattered throughout the background are several black, rounded rectangular shapes of various sizes and orientations. Some of these black shapes have small, light gray rectangular tabs protruding from their edges, resembling a stylized interface or a set of floating cards.

**Teams**

Organizational design is a complex beast. Assigning work to align with strategy and achieve business goals is challenging. Ways of working continually evolve to balance throughput and availability in digital products and services. This research has taken a look at how teams and leaders are working together to deliver a great customer experience (or not), and how the way we work influences uptime. We have found that:

## **Engineering teams' and organizational leaders' perspectives need to be aligned**

Our data show that leaders are more optimistic about their teams' DevOps capabilities than the teams themselves are, and that they also don't realize how much time their teams are spending on monitoring. **Optimism bias** is common in leaders; it needs to be to sustain motivation for change and delivery of vision and strategy, but leaders also need to be grounded in reality. Teams are frequently resource stretched, constrained, and hungry, and yet being asked to do more with the same, or even less. Leaders need to listen to their teams and help them find the time now to save time in the future. As it is, these differing perspectives indicate a conflict with organizational goals.

## **Engineering teams are stuck in monitoring cycles**

This leaves little time to spend on automation, paying down technical debt, and adopting DevOps practices that promise to scale availability in the future. Furthermore, spending time on monitoring the monitoring tools is unlikely to contribute positively to employee experience.

**People are still using waterfall, project-oriented ways of working (not product/agile)**

Nor have they all adopted DevOps or migrated to the cloud yet, but teams are reporting high levels of autonomy. **Success in digital transformation** is reliant on progressive ways of working such as agile, DevOps, SRE, and cloud adoption. While teams haven't yet fully absorbed DevOps practices, the "we build it, we own it" approach is a start that will pay dividends later. As long as the teams are also given the autonomy to decide where to invest their time.

**Engineering teams are the last in an organization to access quantifiable measures**

Logging and error budgets are lagging indicators. Teams generally lack leading indicators to help understand how they will do (e.g. SLIs), and they don't have the capability to see or report on the type of work that they are doing (i.e. unplanned work vs new features). Because there are no standard KPIs tracking what teams spend time on, it is hard to properly communicate how much of the team's time is taken up by monitoring and unplanned work.

This makes it challenging to set expectations for how much work the team can take on (apart from SLAs). It also makes it hard for leaders to measure improvements as organizations progress through digital transformation journeys. Value stream management (VSM), and flow metrics would provide leading indicators of what work is happening.



*"To get to an optimal service model, you need to design the organization. Good behaviors are driven by accountability. The enterprise architecture function is not there for creating documents, but to focus on the realization of value. Each part of the organization needs to be able to operate on its own, but there are horizontal things driven across all, and there are things that each function has to take ownership of and be accountable for."*

*~Mike McGibney  
SVP SaaS, Körber*

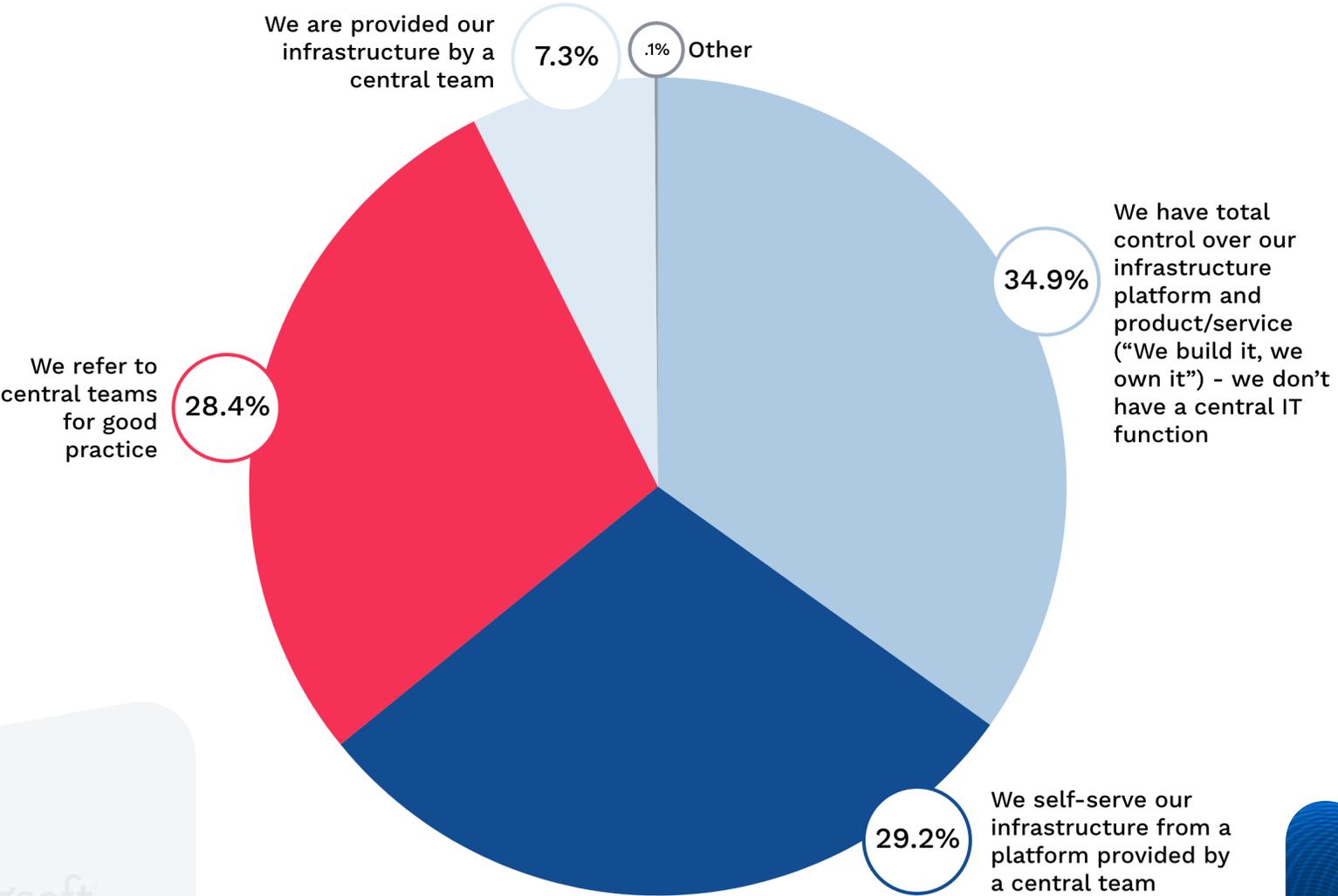
# A NOTE ON VALUE STREAM MANAGEMENT

Value Stream Management (VSM) has been around conceptually since the emergence of lean at Toyota in the 1950s but is currently undergoing a renaissance thanks to advanced tooling that has emerged from the DevOps toolchain era. VSM's goal is the optimization of value—the flow of work, and realization of value outcomes. It puts customer experience at the center of all that it does, and demands that teams are not negatively impacted by waste in many forms. This waste includes the unplanned work that occurs when availability is compromised.

Flow metrics help teams understand the health of their value stream as well as the impact their work is making with their customers. In particular, work distribution helps teams to see the proportions of time they are spending on work and adjust investments accordingly. Additionally, monitoring tools and AIOps can provide deep insights into customer experience at the leading indicator level (as opposed to the lagging indicators that are typically used, such as profits and revenues—which are business metrics acting as proxy metrics for customer experience). VSM balances efficiency and effectiveness.

# Teams Have Shifted to the “We Made It, We Own It” DevOps Culture

## Team Autonomy





*““Centralized teams often provide development teams with tools and guidelines but allow each individual team to do their own thing. There are certain tasks to do and measures to hit, but letting them “run their own business” helps to encourage people to bring their ideas to the team. It’s not that the team is “burned out” necessarily—teams often become accustomed to the amount of load/pressure they are under—but it’s good to minimize it so they can focus on delivering differentiating features. Measuring and minimizing unplanned work is part of that.”*

*~Helen Beal, Strategic Advisor & Analyst*

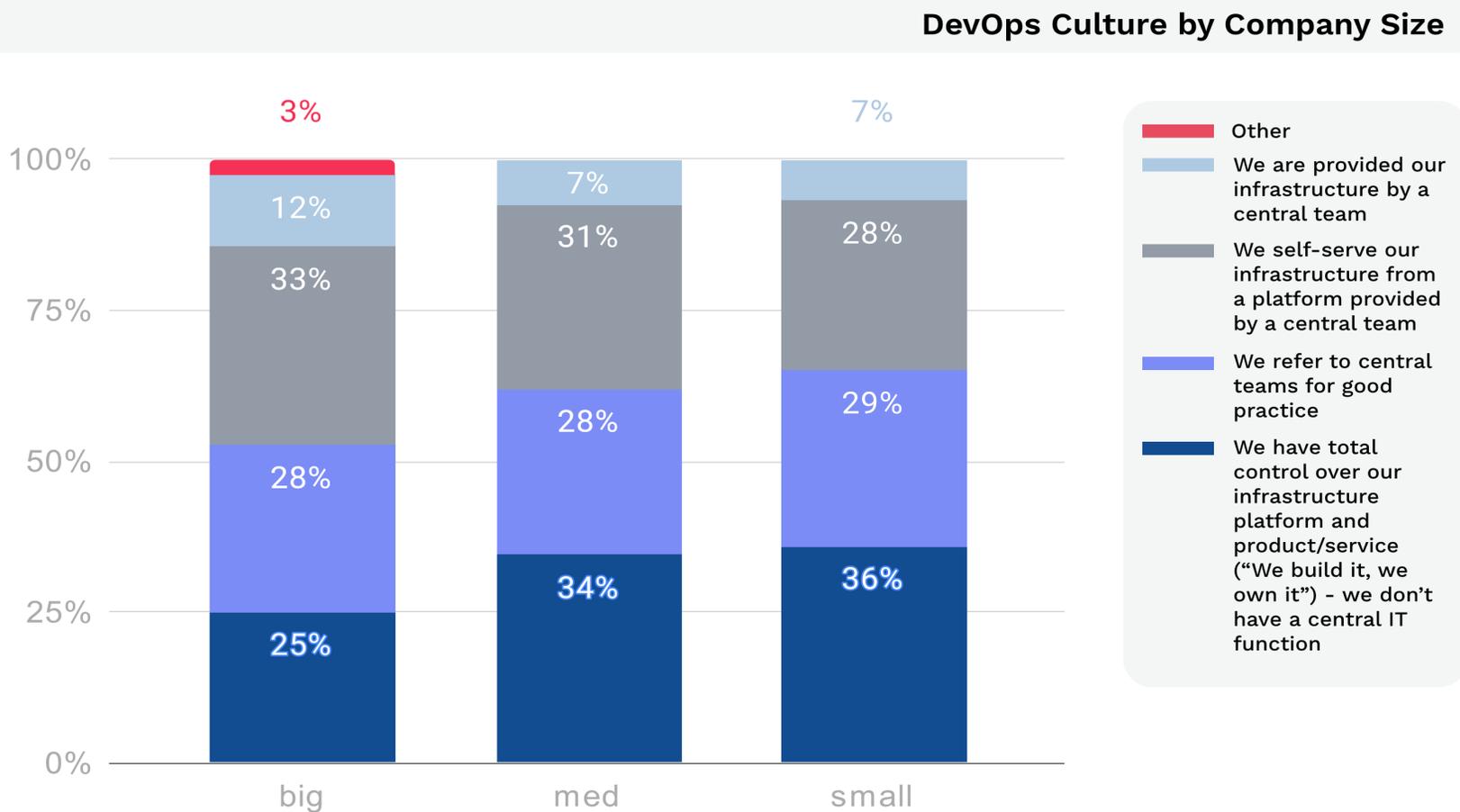
Amazon’s CTO, Werner Vogels, famously told his engineering teams “you build it, you run it”. Using the pronoun “we” helps teams feel accountability and autonomy. And running systems, services and platforms requires end-to-end ownership including ensuring feedback flows from the customer back into planning and decision making (not simply “running” it).

Most of our respondents (34.9%) said that they follow this mantra and have total control over their infrastructure platform and product/service and don’t have a central IT function. 29% self-serve their infrastructure from a platform provided by a central team, and a further 28% refer to central teams for guidance. Only 7% are provided their infrastructure by a central team.

This shows a solid adoption of the team and organizational level design patterns that research such as the [State of DevOps Report 2021](#) reveal correlate with success (e.g. platform teams). But this raises the question, why are these teams then consistently still struggling with meeting their SLAs? They may have control over their infrastructure, but do they have control over their work and their time?

# Teams Larger Companies are Further Behind on DevOps Adoption

Our data show that teams in larger companies have less autonomy—with only 25% of those respondents saying they build and own their infrastructure.



Summary Company Size

## Teams Most DevOps Practices Remain a “Want to Have”

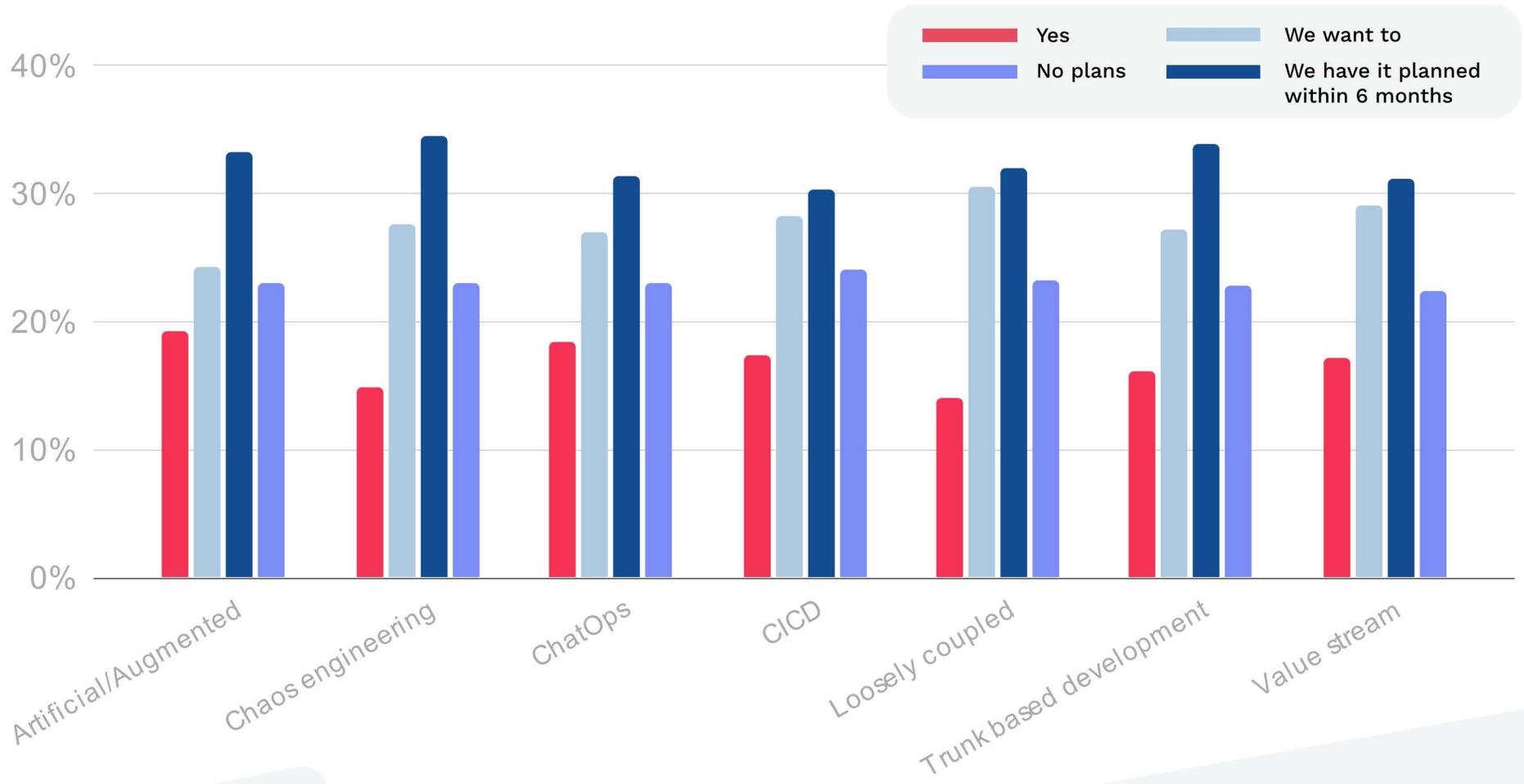
We asked our respondents about their adoption of a range of DevOps capabilities and discovered that, at the team level, less than 20% had implemented any of them. Artificial and augmented intelligence was the top reported category (this isn't necessarily AIOps—respondents could interpret this as AI included anywhere in the DevOps toolchain, e.g. at the developer or testing level).

Chaos engineering was the least adopted but most planned within six months, along with trunk-based development. Both these practices have very close ties with availability (see call outs on page 35).

While the smallest group are those already using these practices, the largest by an overwhelming majority are those who intend to implement these practices. Something is creating friction in the adoption process. Most likely it is all that time teams are spending monitoring and dealing with incidents/unplanned work.

**<20% of  
respondents have  
adopted DevOps  
practices, but AI is  
leading the charge**

## DevOps Adoption vs Roadmap



## A NOTE ON CHAOS ENGINEERING

Netflix developed Chaos Monkey in 2011 to test the resilience of its infrastructure—and by making it available to the community, chaos engineering was born. It's the discipline of devising and executing experiments on a digital product or service with the explicit intention of learning about the system's capability to withstand turbulent conditions and make improvements for improved availability. It requires relatively high levels of stability and the ability to assign time to the improvement practice. Most teams start with experiments on their pre-production environments and graduate to production as they build confidence.

## A NOTE ON TRUNK-BASED DEVELOPMENT

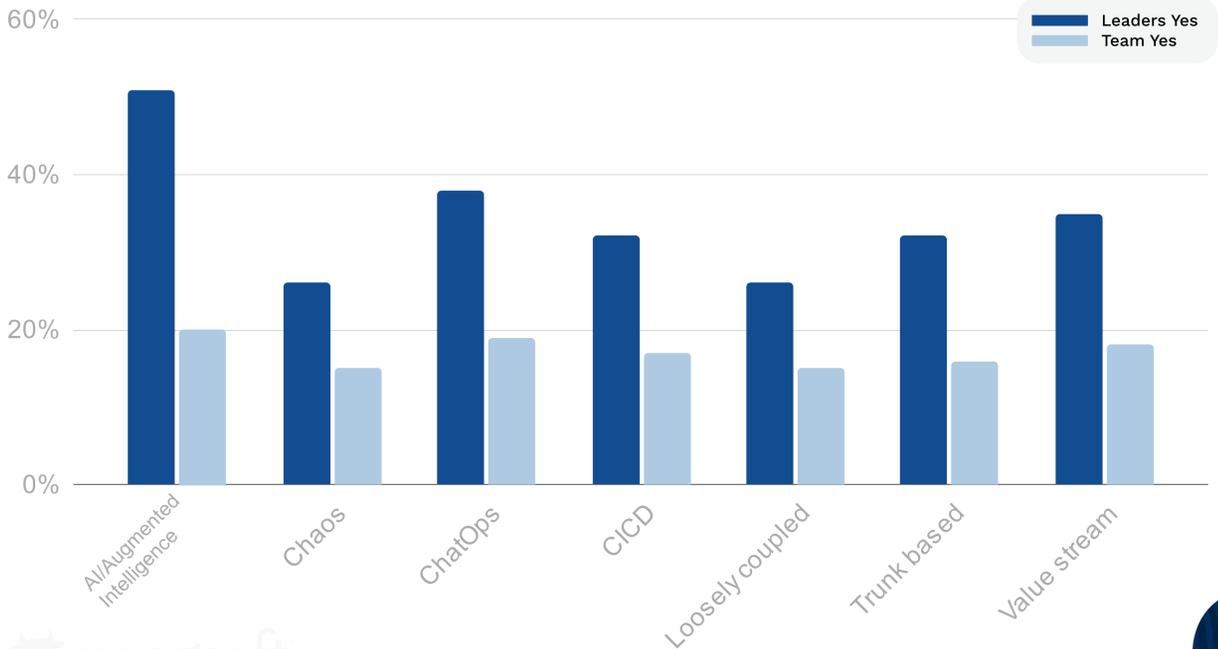
When multiple developers work on a single product or service it can cause painful merge events when their code bases are integrated. In trunk-based development, developers' own feature branches are short-lived (usually less than a day) thus driving frequent and small integration events to minimize risk and continually test quality. It's one of the key characteristics of Continuous Integration (CI)—all developers commit at least daily to trunk along with version control, and automated unit, integration, and user acceptance tests.

This practice aims to build-in quality early and avoid “integration hell.” Ultimately, availability is positively impacted as issues are caught and resolved early on in the pipeline so they don't cause incidents and unplanned work later. CI is the practice that enables software to always be in a releasable state that enables teams to practice continuous delivery. Together, continuous integration and continuous delivery are referred to as CICD.

# Organizational Leaders Believe Their Teams Are at a Much Higher DevOps Adoption Rate

There is a leadership bias toward positively reporting capabilities—management believes they have far more advanced DevOps capabilities than teams are reporting across all categories, but especially AI, ChatOps, CICD, and VSM. As we have learned, teams are very keen to adopt these capabilities, but they likely don't have the time. There's a consistent difference in perspective between the work that teams think they are doing, and what management believes they are working on.

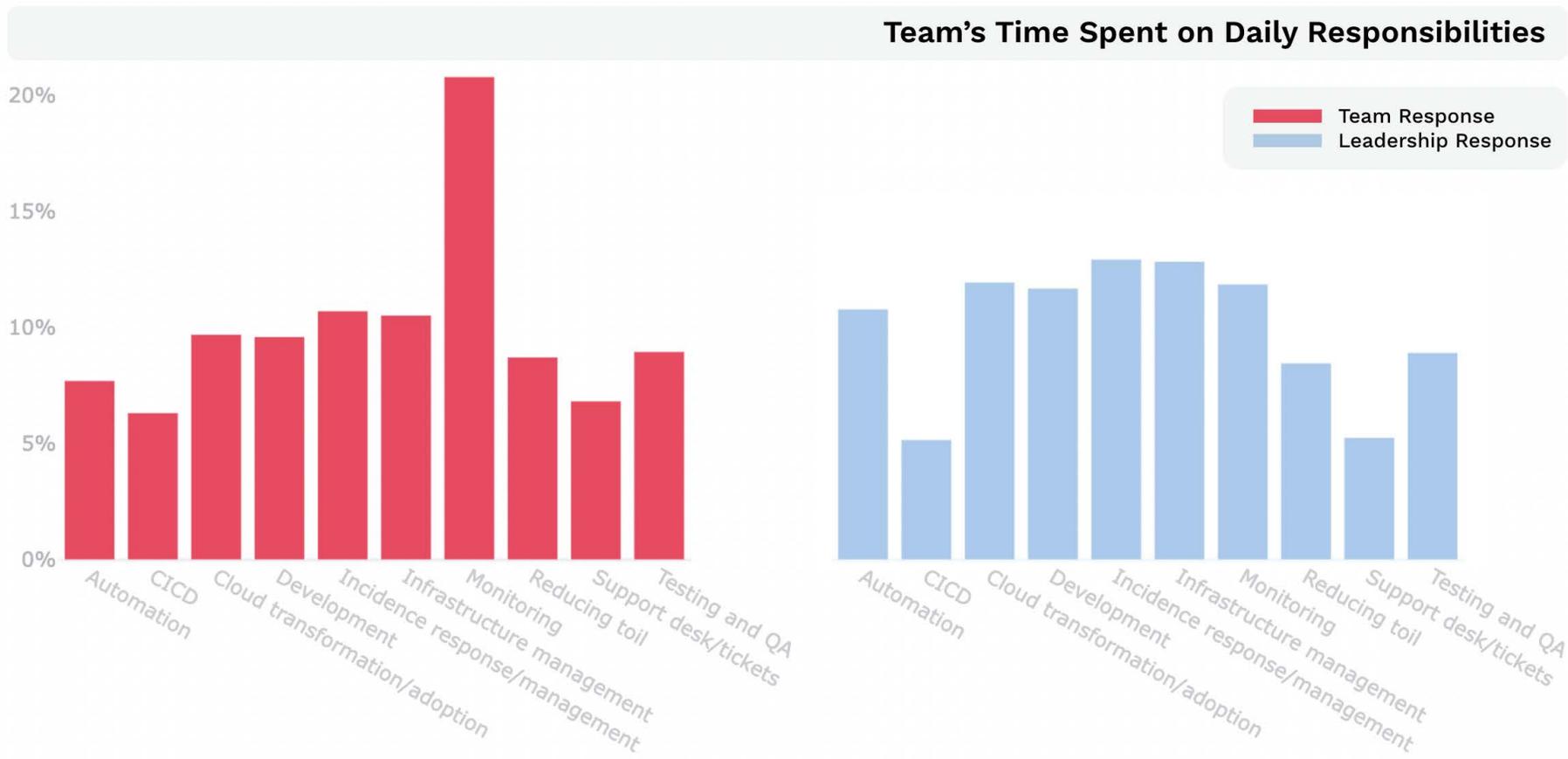
It's up to the teams to make their work visible and management to enable them to do so. Then leaders need to help their teams to discover improvements and ensure they are able to find the time to invest in the future—even if this means throttling change work coming into the system.



*“Manage up/manage down is the single most dangerous attitude in business. Transparency between leadership and execution teams in business is vital.”*

*~Phil Tee, Moogsoft CEO*

# Leaders Are Also Unaware of How Much of Their Teams' Time is Spent on Monitoring



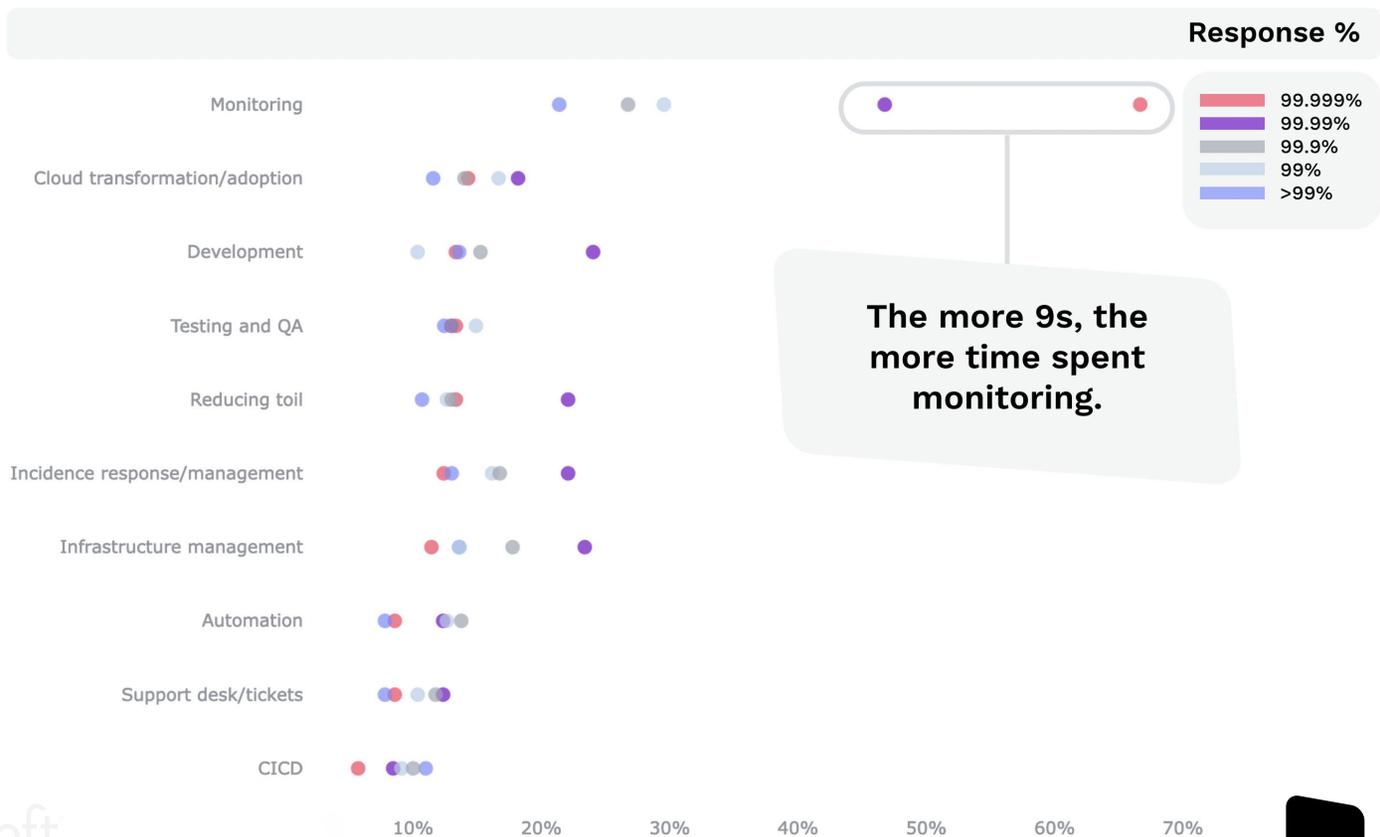


Teams spend by far the most time monitoring over anything else. Yet management believes they are spending time fairly equally across the board, with less time most notably on CI/CD, support desk/tickets (the same for the teams), reducing toil and testing, and QA.

Looking at the data this way showed us the leaders' optimism bias again. Management thinks the teams are spending time investing in the future, whereas the teams lean more heavily towards keeping today's systems alive. This should be a wake-up call to leaders everywhere—if you want to invest in the work that enables digital transformation, and inject capacity into teams, you need to help them now to find time and find ways to create more time in the future. It is commonly difficult to carve out time to spend on experimentation and innovation.

# The Higher the SLAs, the More Time Spent Monitoring

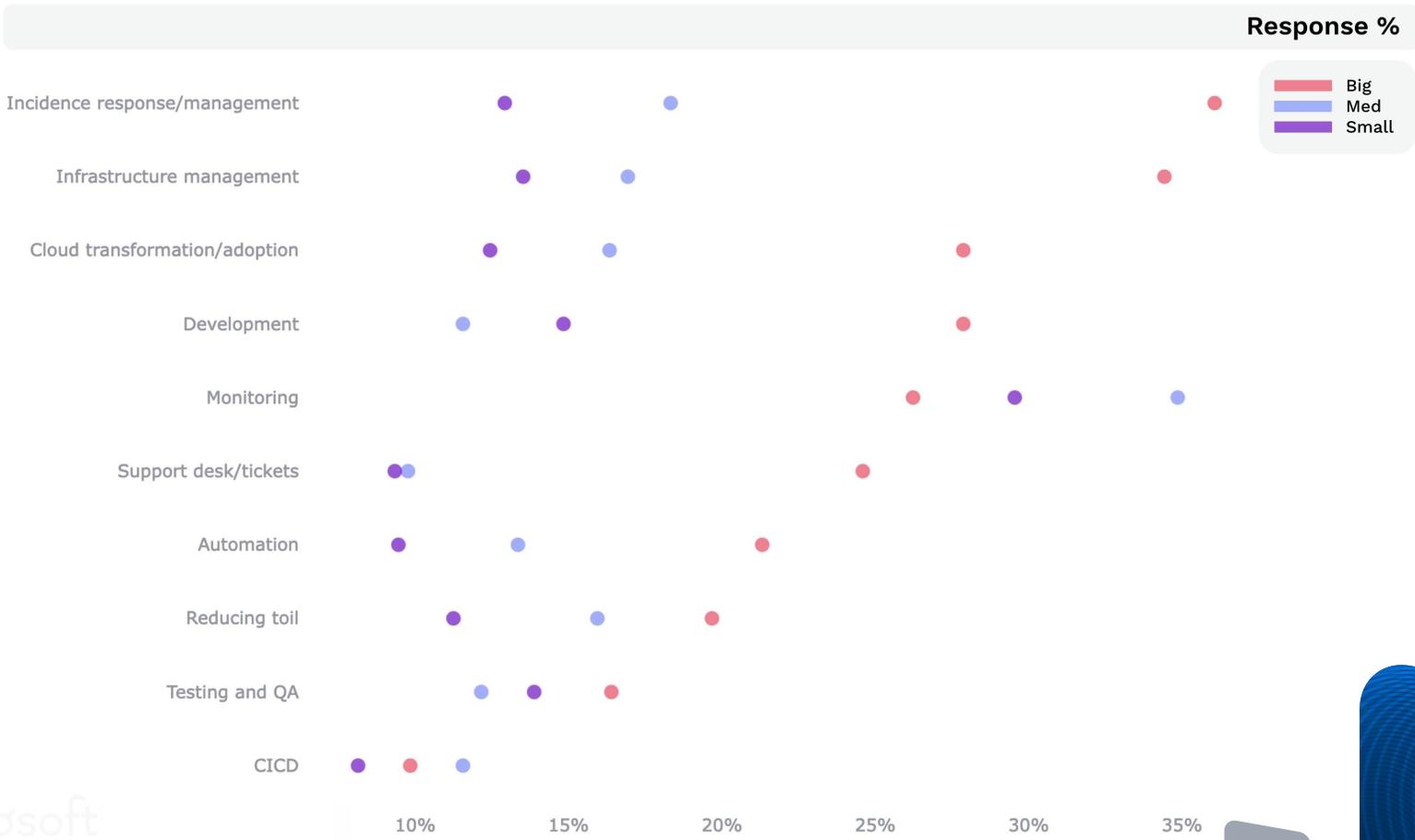
The more nines in an SLA, the more time is spent monitoring—and larger companies have higher SLAs. We also discovered that the larger the organization, the more time is spent on incident and infrastructure management (see below). None of these activities (monitoring, incident, and infrastructure management) is value-adding to the customer. And they are often stressful, onerous, and demotivating. They eat up time that teams could be investing in DevOps capabilities that will buy them more capacity in the long term to spend on improving customer experience with new features and faster platforms.



The more 9s, the more time spent monitoring.

# Larger Organizations Spend More Time on Incident and Infrastructure Management

Since time equals money, we can also surmise that larger organizations spend more money on these activities. These talented and expensive engineers may be in centralized IT Operations teams and operations centers, or embedded in multifunctional, cross-skilled teams. Either way, larger organizations are burning higher proportions of their IT budgets keeping their products and services running—not on making them better.



# Teams Key Takeaways

**DevOps adoption and the move from project to product is still very much underway**

**Engineering teams are stuck in monitoring cycles—and it's not working (and not fun)**

**Organizations are lacking the metrics that align teams and leaders, or provide insights to improve**

**Finding and replacing SREs and developers with DevOps skills is very expensive—investing in tools that support DevOps ideals is a cost-effective alternative**

The background is a dark blue gradient with several large, overlapping, wavy shapes in a lighter blue color. These shapes are composed of many thin, concentric lines, creating a ripple effect. Scattered throughout the background are several black, rounded rectangular shapes of various sizes and orientations. Some of these black shapes have small, light gray rectangular tabs protruding from their edges, resembling components of a puzzle or a modular design.

**Tools**

People and processes may come first but tools matter—they have enabled so much of the digital transformation we have seen in recent years. The internet is the tool at the bedrock of these transformations, and software really has eaten the world. Monitoring was one of the first tool categories to appear, initially in the 1990s in the operating system. A lot has changed since then. Let's take a look at how:

## **Cloud migration is still happening**

While it can often feel that cloud has happened, the reality is that most organizations are still in the adoption and migration phase with only about 50% of infrastructure moved to cloud so far. Cloud promises higher availability through modern distributed architectures and elasticity. It is also a prerequisite for many of the optimizations offered by DevOps. While it may be true that not everyone wants all of their workloads in the cloud, most organizations do want most.

## **Everyone has a LOT of monitoring tooling**

On average respondents said they have 16 monitoring tools, and in some cases, they have up to 40 tools. Single domain monitoring tools are proliferating in the tool stack, demanding more time from teams to monitor, not less. Higher SLA teams have a higher average number of tools per category. That's a lot of tools and goes a long way to explaining why teams are spending so much time on monitoring. It's likely that leaders know they have invested in all of these tools, but underestimate how long teams have to spend managing and maintaining them. And looking for answers when a problem occurs.

## AIOps is already part of the key technology stack

Monitoring tool types are all about equal with around 80% of respondents reporting using them, whether network, web performance, integration/API, storage, system, RUM, EUM, APM, logging, dependency, CICD, social sentiment/brand monitoring. Almost everyone is using everything but it's apparently not working as teams are still missing their SLAs, and nearly half the time their customers are telling them about problems before these tools.

## DevOps toolchain adoption is lagging

Only a third of respondents have DevOps toolchain and automation capabilities today but almost everyone wants them. Once more, it seems that the time spent on managing the existing investments is disabling teams from optimizing for the future.



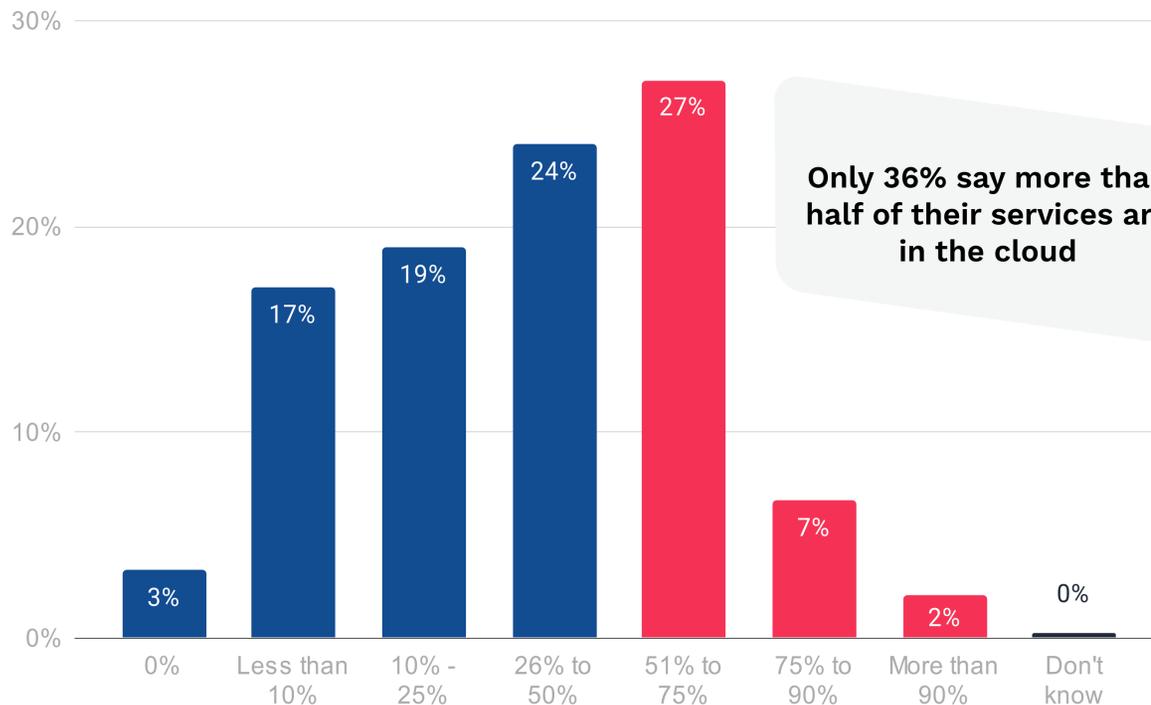
*“Perhaps the first phase of the cloud transition in SRE/DevOps has been characterized by the provisioning of observability to gather data. The success of the second phase will be the rebalancing of monitoring towards AIOps, and advanced correlation, to ensure availability is actively managed rather than heroically pursued.”*

*~Phil Tee, Moogsoft CEO*

## Tools

# The Majority of Services Are Still Not in the Cloud

Only 36% of respondents report that more than half of their services are in the cloud. The key benefits of cloud computing are scalability, reliability, and availability.



And, along with DevOps, cloud is a key enabler for digital transformation.

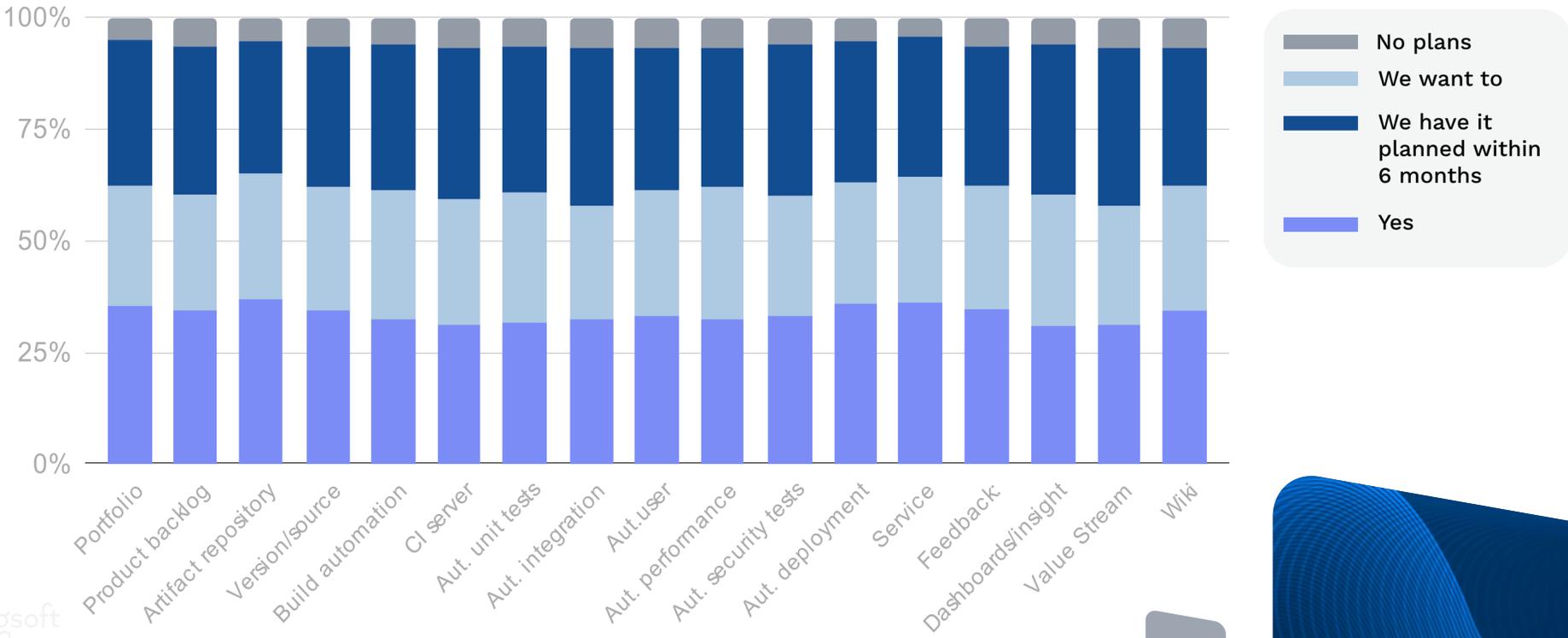
In our research, we saw respondents struggling to move from project to product, adopt DevOps practices and toolchains, and with cloud adoption more generally.

We also discovered teams are overwhelmed with monitoring, incident, and infrastructure management. This reinforces the message that teams need methods to get themselves out of this quagmire so that they can spend their time on innovation and optimization.

# Organizations are Still at the Implementation Stage of DevOps Toolchains

This data quite closely reflects what we saw in the Teams section of this report looking at the adoption of DevOps practices. Adoption is far outstripped by those who are planning for these toolchain components or who want them, but haven't started planning. With DevOps toolchains though, over 25% of respondents—versus less than 20% of respondents relating to practices—have already adopted these components. And only around 5% at most have no plans to use.

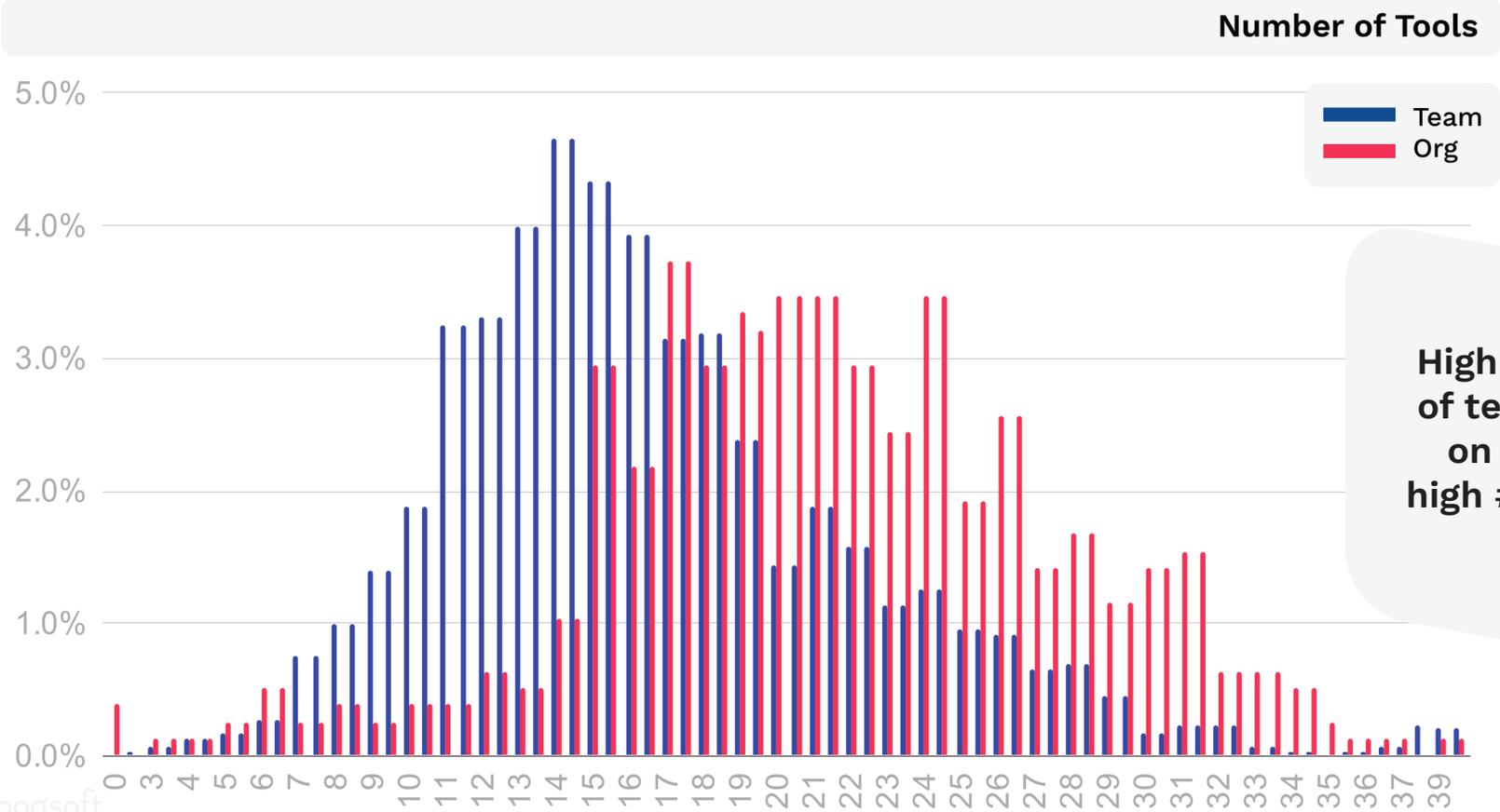
The highest interest is in build automation, CI server, automated unit tests, and value stream management (VSM).



# Teams are Managing Huge Amounts of Monitoring Tools

It's no wonder teams are reporting that they are spending huge amounts of their time monitoring; on average, teams are monitoring 16 tools—if not more.

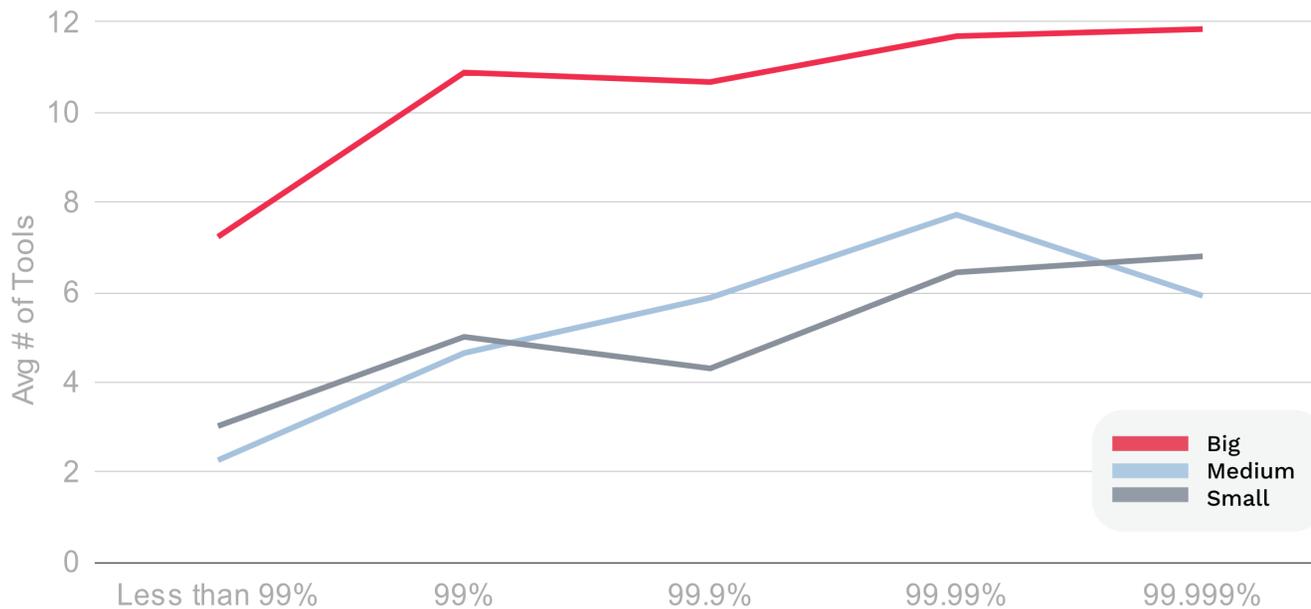
Leaders report thinking of even more tools. This could be because they are concerned with the whole organization (i.e. multiple teams), or because they don't understand which tools are actually being used.



High SLAs = high % of team time spent on monitoring + high # of monitoring tools

# Teams with Higher SLAs Manage More Tools Per Category

Average Number of Tools by SLA & Company Size



It seems that to reach those higher SLAs, more tools are required across every category, peaking at four nines. And more time is spent monitoring.

Tools incur license fees and management and maintenance overheads.

Availability costs money.

Some of this investment is paying off as the teams with the higher SLAs are more likely to meet them.

No doubt availability would suffer if the tools and the people were taken away, but for most organizations, availability isn't where it should be. SLAs are regularly being missed—customers are frequently reporting issues before the tools, and teams are spending huge amounts of time on monitoring which is not creating customer value.

# Tools

## Key Takeaways

**Use AIOps to reduce MTTD and MTTR to improve customer experience and release time for improvements**

**Consolidate monitoring tools to reduce license, management, and maintenance overheads**

**Focus on employee experience—which is intrinsically linked to customer experience—to improve organizational performance**

The background is a dark blue gradient with several abstract elements. There are large, overlapping shapes with concentric white lines, resembling ripples or sound waves. Scattered throughout are various geometric shapes: white rounded rectangles, black rounded rectangles, and black rounded squares. The overall aesthetic is modern and digital.

# Our Guidance

The data show that teams and organizations must improve availability—they are missing SLAs far too frequently and should not be alerted to a problem by their customers. Our research shows that teams and organizations must not only set a KPI for availability—anything not measured will atrophy—but the actual goal is staying ahead of customer sentiment. Regardless of actual availability being industry standard or exceeding an SLA, customers want a partner that is proactively managing their experience. To do this, difficult decisions have to be made in the face of violated error budgets before the violation occurs or in response to an event.

Availability comes at a high cost, and leaders need to buy back time for their teams so that they can invest in technical stability. All the investment made in monitoring is causing teams to spend more time monitoring. Tools and noise have proliferated as a result of throwing tools at flaky systems. Engineering teams currently lack standard KPIs to communicate their time spent on maintenance, versus building and automating, causing a mismatch in leaders' and teams' perspectives on capabilities and work profiles.

By reducing the time teams spend on monitoring and incident management, the more time they can spend creating value that will improve customer experience—be that new features or platform improvements. Less unplanned work means more time paying down technical debt and automating toil and improvements in employee experience. And it's more time teams can spend on mastery and learning.

As Andrew Clay Shafer put it, “You’re either building a learning organization, or you’re losing out to someone who is.”

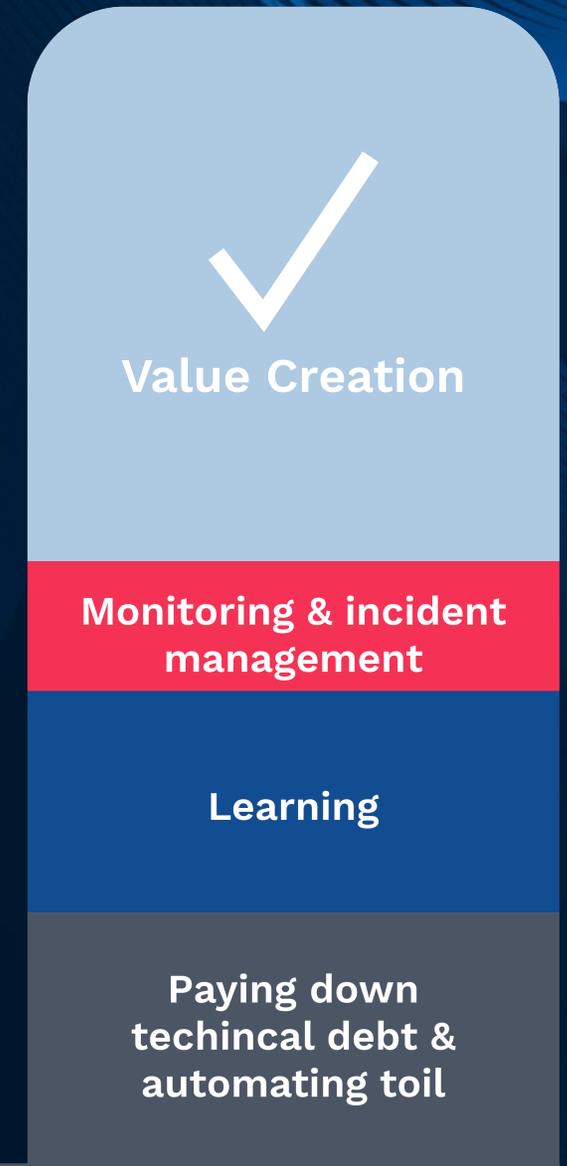
And mastery also contributes to employee experience, along with autonomy and purpose, according to Dan Pink’s research in *Drive*. Organizations are maintaining availability, but they aren’t investing in stability, and they are running out of runway. You must transition to technical stability to scale and sustain your team and your infrastructure.

So how do you do that?

**What teams spend their time doing.**



**Without Moogsoft**



With **moogsoft**

The background is a dark blue gradient with several large, overlapping circles of concentric, lighter blue lines. Scattered throughout are several black rectangular shapes of varying sizes and orientations, some with small white rectangular cutouts.

# The Steps to Success

We've put together our six "Steps to Success" to help guide you and your organization towards scaling and sustaining your team and infrastructure.

1

## Baseline Your Current State

- Document and agree on your business goals in the context of availability
- Identify and classify business-critical apps, services, and infrastructure
- Discover and document the tools you have, their usage, and their costs
- Identify your current KPIs, SLAs, SLOs, and SLIs if you have them

### Output & Outcomes

The results of your analysis will show you which of your technology assets matter most to your availability goals and show you where you need to invest and divest. This is the foundation for what follows.

2

## Choose Your KPIs

- Unlock the capability to monitor from MTTD to MTTR and set objectives for reducing both
- Set SLOs, SLIs and error budgets if you haven't already and track how frequently a customer catches an issue before you do
- Make the work distribution in your team visible (add tags in your ticketing tools) for unplanned work, improvements, technical debt, and new features

### Output & Outcomes

The results of your analysis will show you which of your technology assets matter most to your availability goals and show you where you need to invest and divest. This is the foundation for what follows.

# 3

## Consolidate Your Tools

- Prioritize your existing monitoring tools by usage and value
- Focus on where you can reduce your monitoring tools' footprint, and correlation gap
- If you are not actively using a particular dataset, remove it

### Output & Outcomes

Reducing the number of monitoring tools you have will not just reduce your total cost of ownership (TCO)—it'll also start to dampen the noise you're dealing with and lessen alert fatigue, potentially also reducing MTTD and MTTR.

# 4

## Target Noise Reduction

- Adopt AIOps to rapidly reduce noise
- Give leadership and the teams a single view of all the data and insights
- Measure the impact on the volume of unplanned work your team has to deal with

### Output & Outcomes

Using AIOps to monitor your monitoring will produce an immediate reduction in ticket volume and shrink MTTD and MTTR by identifying the source of problems sooner, releasing time and capacity for you to move onto the next step.

5

## Pay Down Technical Debt

- Prioritize where the technical debt hurts the most using insights from Moogsoft to shine light onto problem areas
- Automate toil away to release even more time
- Use chaos engineering experiments to further improve availability

### Output & Outcomes

Initially you'll see more tickets tagged as technical debt—then fewer as it's paid down and you benefit from increased system stability presenting itself as reduced mean time between Incidents and less unplanned work. You'll see an increase in tickets allocated to new features and improvements.

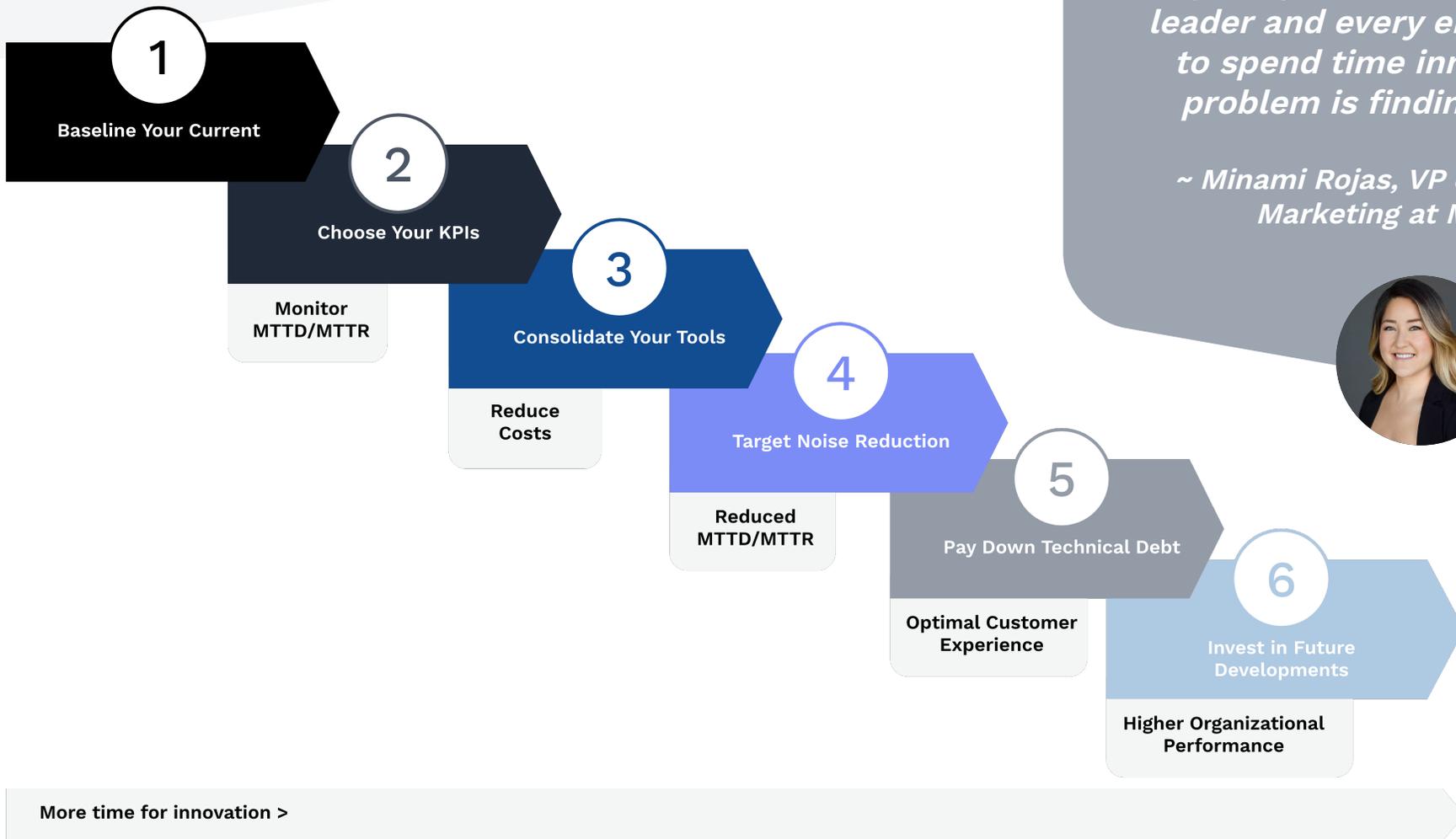
6

## Invest in Future Developments

- Reprioritize on innovating for customer experience, and away from maintaining customer experience
- Remeasure how frequently you are catching problems before your customers, and report on the improvement
- Refocus efforts on learning and implementing DevOps practices to further nurture both throughput and availability

### Output & Outcomes

By now, not only will you be seeing a reduction in MTTD, MTTR, and MTBI, you'll also be meeting SLAs more frequently so you can adopt higher SLAs. You'll be seeing more user stories completed, higher levels of customer satisfaction reported and higher levels of DevOps capability adoption reported.



*“Figuring out how to optimize a complex infrastructure and team environment can get quickly overwhelming. Every leader and every engineer wants to spend time innovating, the problem is finding the time.”*

*~ Minami Rojas, VP of Growth and Marketing at Moogsoft*



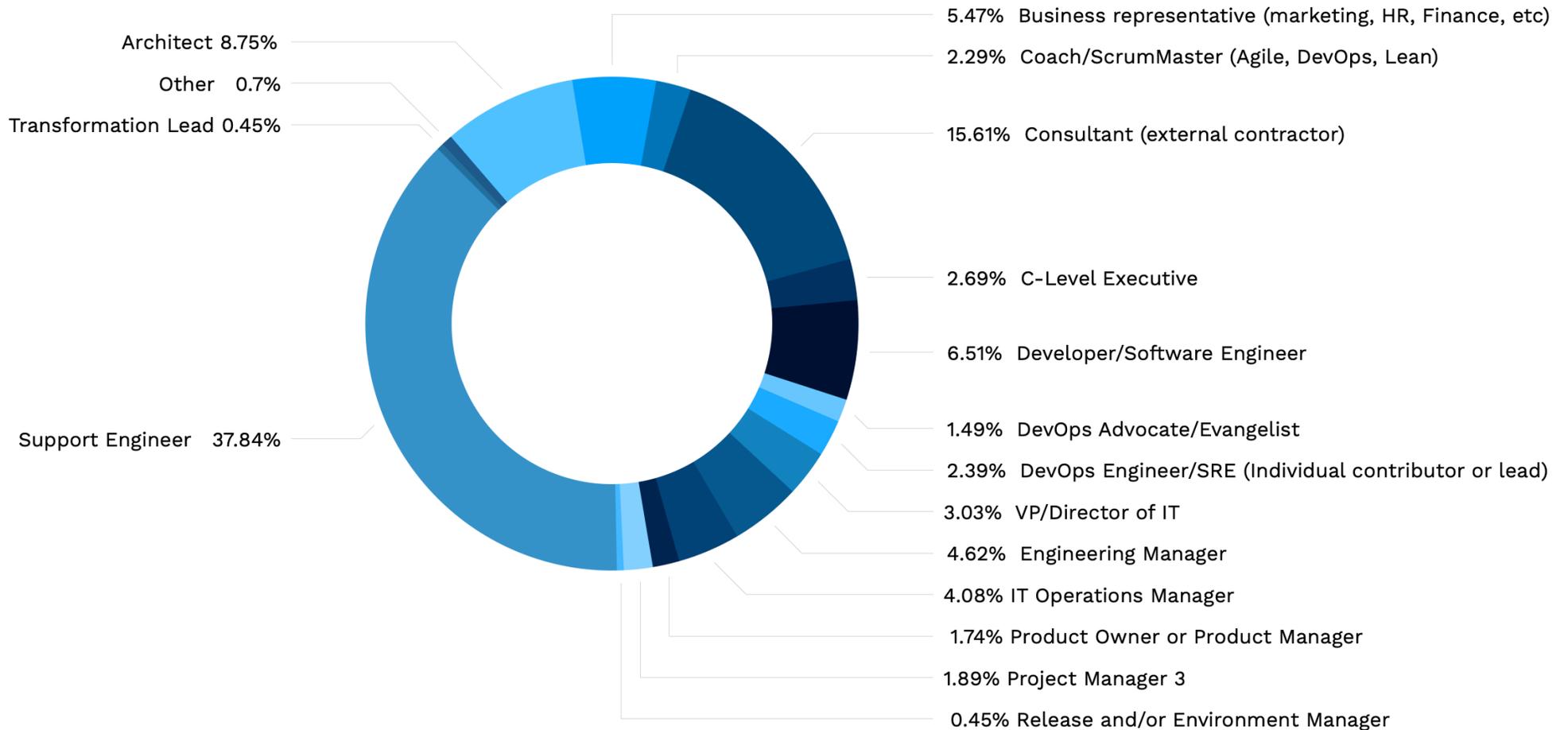
More time for innovation >



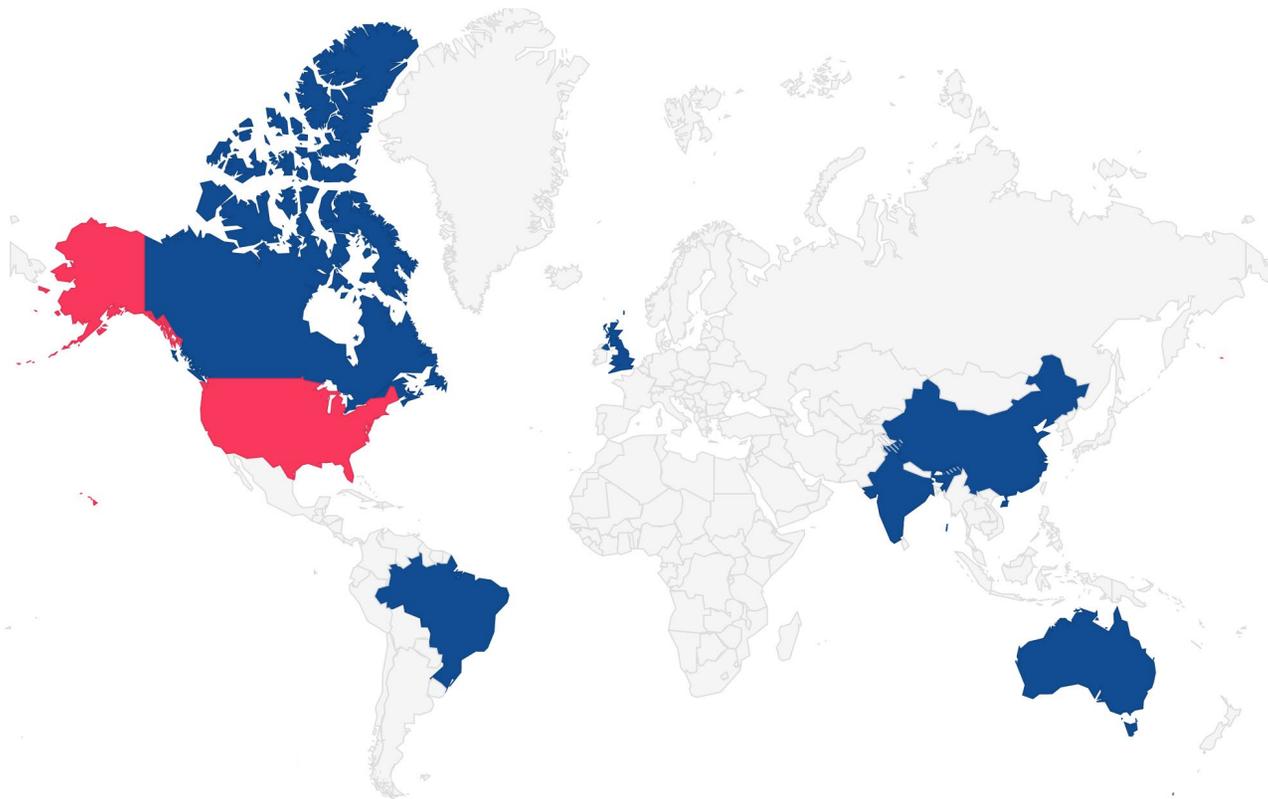
# Survey Demographics

# Demographics Roles

n=1899 with a 92% completion rate.

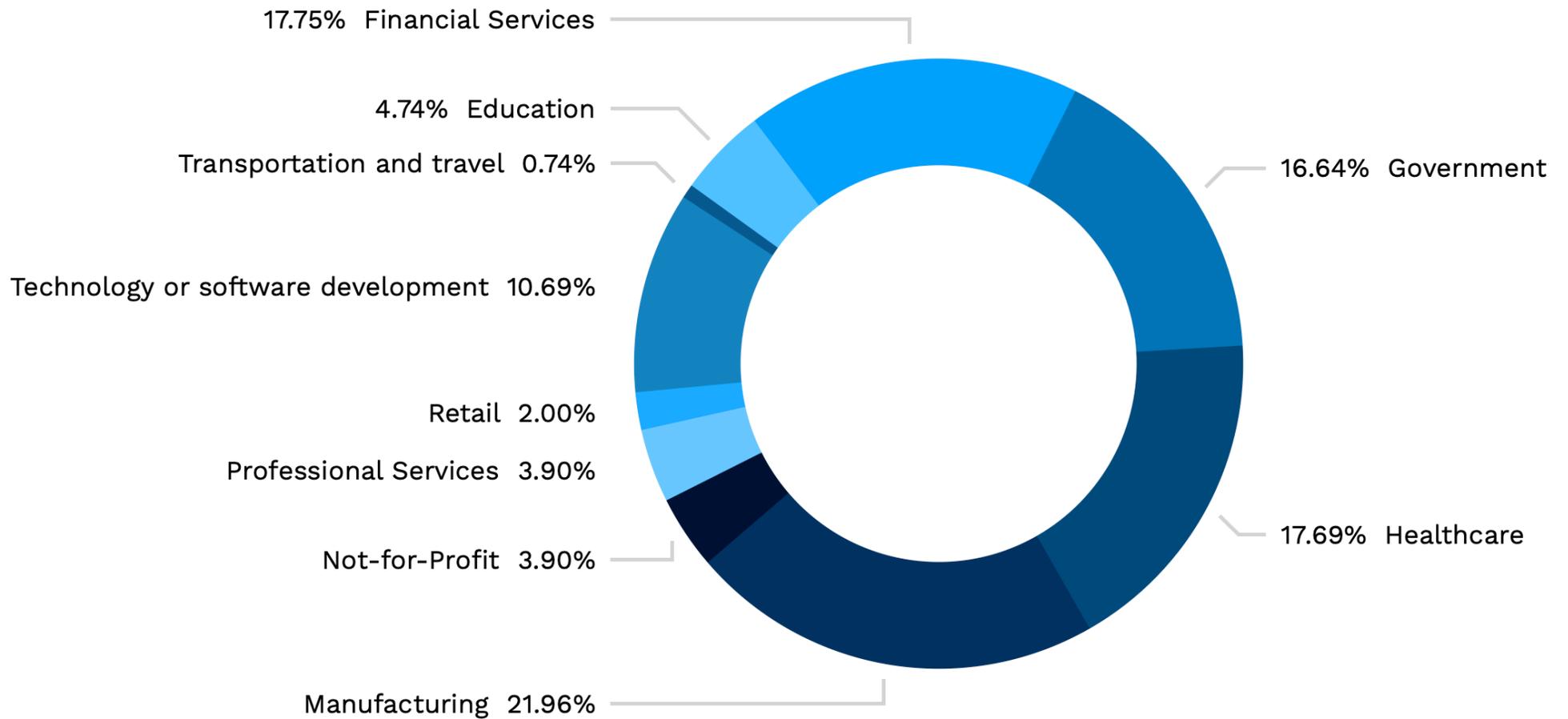


# Demographics Geography

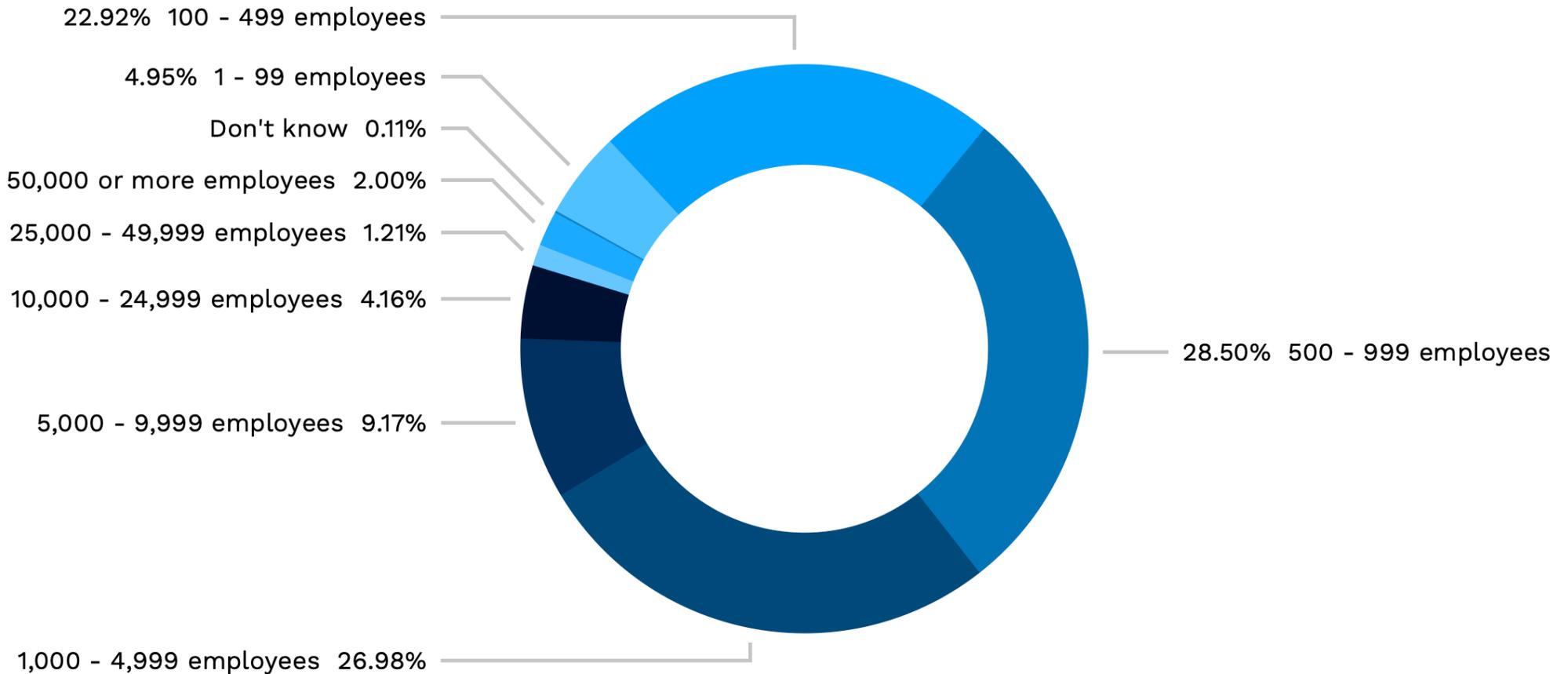


<b>US 86.69%</b>	<b>CN 1.25%</b>	<b>GB 0.48%</b>	<b>AU 0.24%</b>	<b>UNK 0.19%</b>
<b>IN 7.47%</b>	<b>CA 1.06%</b>	<b>HK 0.39%</b>	<b>BR 0.24%</b>	

# Demographics **Industry**



## Demographics **Organizational Size**



# Authors

## Lead Analysts:

Helen Beal  
Minami Rojas

## Contributors:

Kai Wang  
Melissa Hill  
Mike McGibbney

## Moogsoft Team:

Phil Tee  
Richard Whitehead  
John Haley  
Eric Brousseau  
Mike Cabot  
Chris Boyd

## Editing and Proofreading:

Rupert Field

## Design:

Alison Shutterly



## Our Mission

As the world grows ever more complex, we stick to our belief that simplicity is the key to greater achievement. And we never forget that AI is built on human designs, dreams, and desires. That's why everything Moogsoft makes, helps people and machines work more harmoniously, to create clarity from chaos, and expedite innovation.

Moogsoft provides an AIOps solution for frontline engineers faced with availability pressure by detecting problems before they become critical, identifying who should respond, and understanding patterns to prevent similar issues in the future.

**Learn More at [www.moogsoft.com](http://www.moogsoft.com)**