



State of SRE Report

2022 Edition

Research presented by Dynatrace

Foreword

Site reliability engineering (SRE) has moved center stage as organizations look to harness cloud automation to accelerate digital transformation. Most organizations, however, remain relatively immature in their adoption of SRE, which is an often-misunderstood discipline.

First and foremost, SRE is about innovation, education, and enablement. It drives greater alignment between development teams and supports the collective effort to define the best practices that will enable teams across disciplines to automate processes at scale, to meet the organization's business, security, quality, and performance goals.

It cannot be the team that has sole responsibility for automating development processes, configuring service-level objectives (SLOs), or creating fixes and workarounds to avoid overrun error budgets. In addition, it cannot be the only team analyzing vulnerabilities or building self-healing and observability into applications and infrastructure. If it does all those things, SRE will become another traditional operations or security function.

SRE is evolving into a more strategic role, focused on equipping development teams with the tools, data, and capabilities they need to drive modern development and innovation. SRE is also well-positioned to help organizations tackle new challenges, such as the growth of new technologies, languages, platforms, and tools in cloud-native delivery which have created an explosion of complexity.

There are now more than 1,000 solutions in the Cloud Native Computing Foundation (CNCF) landscape, which is far too many for any single developer or team to manage. As a result, different software development tribes are emerging, with pockets of knowledge and tooling specialties and preferences.

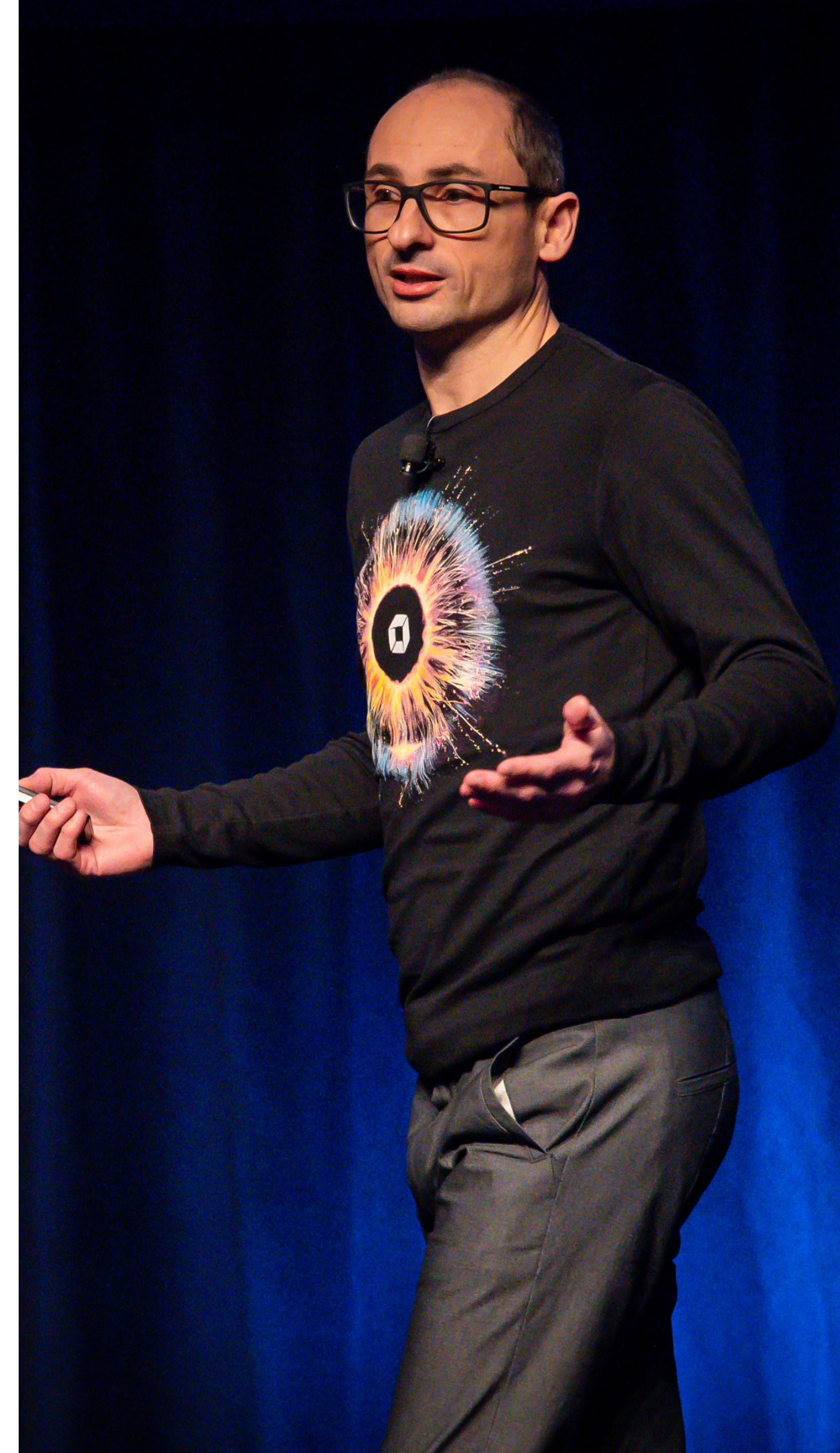
This makes it impossible to apply a standard approach to observability, automation, self-healing, and vulnerability management which is required to drive reliability across the development lifecycle. That's why it's crucial for SREs to define a "golden path" — a key set of steps development teams can take — to navigate this complexity and get where they need to go, regardless of the tools they use.

The availability of self-service observability and monitoring-as-code approaches across the DevSecOps lifecycle is key, allowing development teams to build feedback loops into their applications in just a few clicks. In this way, SREs will lead the charge in going beyond basic automation to smart orchestration of customer experience and business outcomes. That will empower development teams to drive transformation faster than ever, through self-healing cloud applications that quickly scale with business needs and are reliable and secure by default.

This report digs deeper into current SRE maturity and identifies the key trends and challenges organizations are tackling amidst the complexity of cloud-native development. It reflects the input of 450 SREs from organizations around the world and draws on first-hand experience from those driving reliability best practices.

I hope you find valuable insights within these pages to help you define your own golden path and drive SRE to the next level.

—**Bernd Greifeneder**
Founder and CTO, Dynatrace



Contents

Introduction

Executive summary4

Chapter 1

Evolution of SRE.....5

Chapter 2

Role of the SLO.....16

Chapter 3

Keys to SRE success.....32

Conclusion.....44

Methodology45

Executive Summary

This report is based on a global survey of 450 SREs across a diverse range of industries and offers an unparalleled perspective into how site reliability engineering (SRE) is evolving as a discipline. The report uncovers where there are challenges to overcome, and what the future of SRE looks like in a world where the reliability, security, and resiliency of digital services is paramount for business success.

Some of the key findings we'll explore in greater depth include:



SRE practice is maturing, but not quickly enough

- Site reliability engineers (SREs) are increasingly sought after, as organizations have developed a greater understanding of their strategic value. They are, however, in short supply, so finding a way to support and enhance their efforts is crucial.
- As the maturity of SRE practices has increased, they are slowly “shifting left” as practitioners become more involved in architectural design, software development, and testing processes that fall earlier in the lifecycle. They are also driving increased adoption of DevSecOps practices to ensure security is top of mind at all stages of the development lifecycle, but these trends need to accelerate.



SLOs are becoming a staple for SRE but challenges exist in maximizing its full potential

- Despite the growing focus on service-level objectives (SLOs) as a measure of success, almost all SREs say there are significant challenges to defining and creating them. However, most of these issues are tactical problems that should be easy to overcome with the right approach.
- Evaluating SLOs is still a messy process that needs to be better defined and implemented with greater consistency across the organization, as many organizations remain unclear on who owns SLOs, and too much is left solely to SREs.



Efforts to reduce toil from SRE practices must redouble to deliver success

- There is a growing use of automation in SRE practices, but there is good and bad automation. Organizations must identify the difference and adopt strategies and solutions that make their SREs more productive.
- AIOps and unified observability solutions are becoming more important to scaling SRE practices further across the organization, but they can't just be layered on top of existing toolchains.



CHAPTER 1

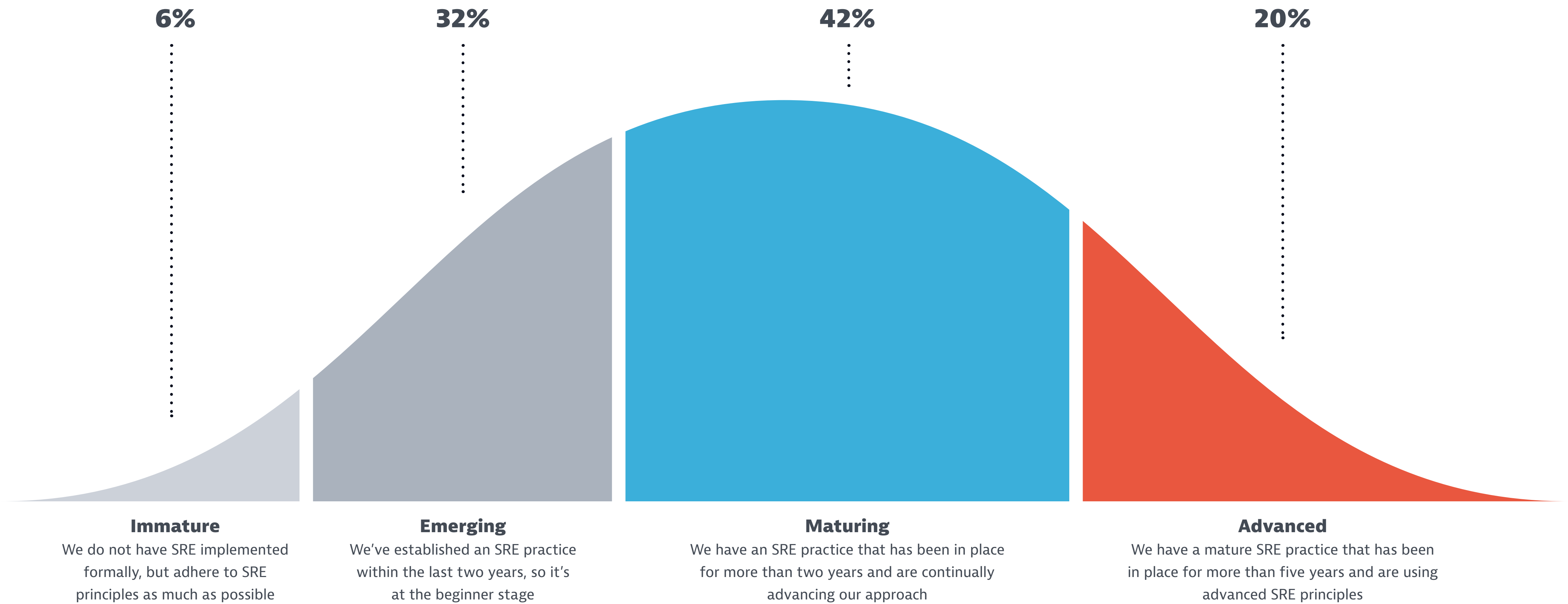
Evolution of SRE

SRE is at the early stage of the adoption curve

Site reliability engineering is gaining momentum, but gaps remain. Organizations need to evolve their approach to SRE, as only one in five (20%) claim to have a mature practice.

Additionally, 88% of SREs say there is now more recognition of the strategic importance of their role to business success than there was three years ago.

At what stage in the site reliability engineering (SRE) journey
is your organization currently?



SRE best practice:

Move away from manual, ad hoc scripts and adopt a platform-based solution with state-of-the-art automation and everything-as-code capabilities that support the full lifecycle from configuration and testing to observability and remediation.

Theme 1

MTTR reduction remains top of the list for SREs

SREs remain focused on improving the reliability of production systems, where reducing mean-time-to-repair (MTTR) is their number one priority.

However, the majority (60%) of SREs find much of their time drained by building and maintaining automation code. **While increasing automation is a key goal, the efficiencies derived will be lost if the process of enabling it is arduous and time-consuming.**

Much of the problem is rooted in how SRE teams build automation for DevOps workflows. Often, teams handle this on a case-by-case basis because their tooling doesn't come with automation built in, and don't offer everything-as-code capabilities.

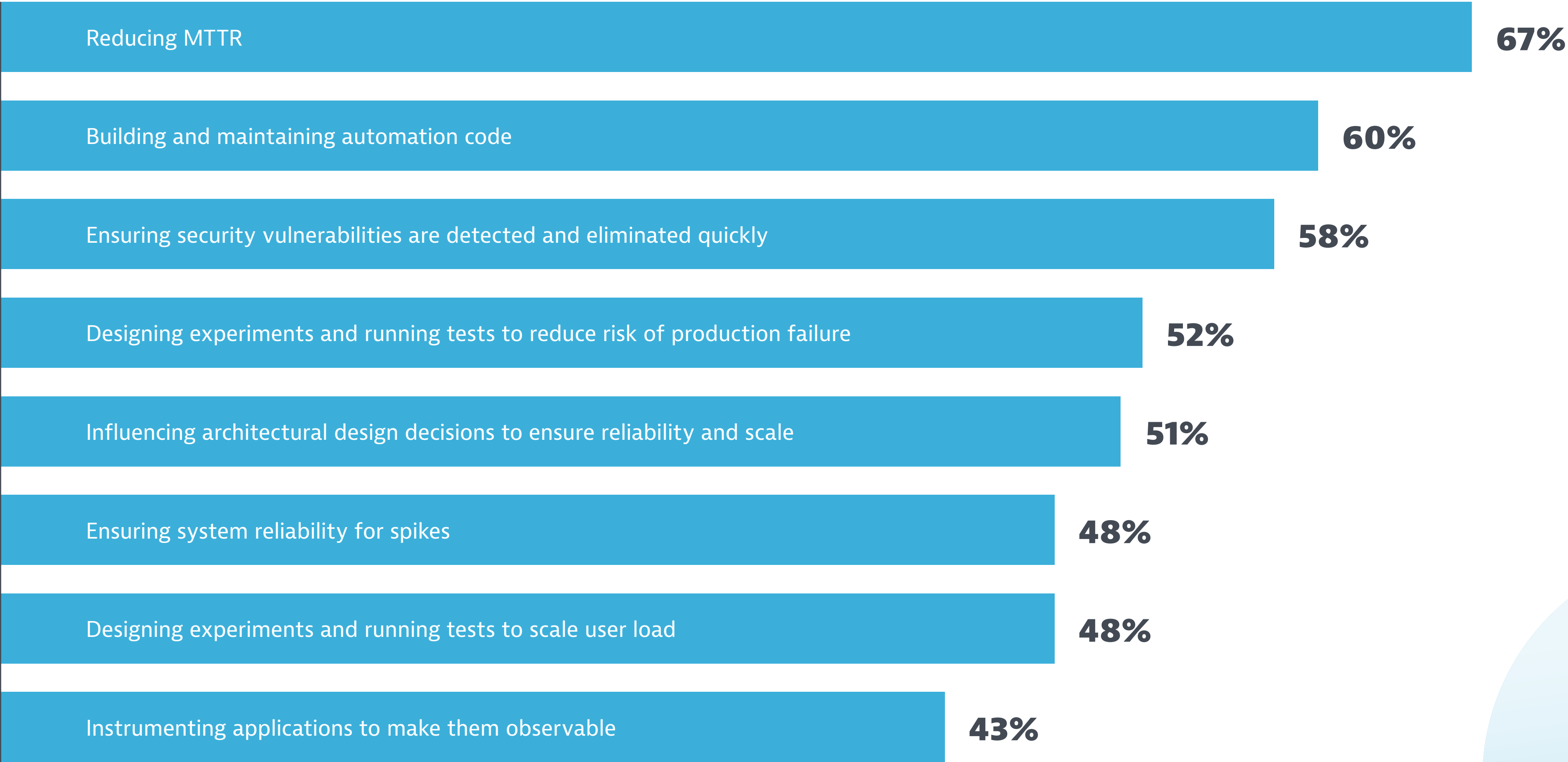
As a result, they're forced to build a layer of automation on top of their tooling. Over time, this creates a complex web of code that becomes more difficult to scale across the DevOps pipeline. SREs will undoubtedly find more of their time is drained in future if they don't identify a more efficient, longer-term approach.

This underscores the need for SREs to work with DevOps teams, developers, and architects to ensure software not only meets a business need but is resilient and automatable by default. This will enable teams to easily integrate new automation capabilities with existing tools and workflows, reducing manual effort for SREs and improving engineering practices.

88%

of SREs say there is now more understanding of the strategic importance of their role to the success of the business than there was three years ago.

Which of the following tasks do SREs in your organization dedicate the largest amount of their time to in an average week? (All responses)



Theme 2

A shift to SRE-driven engineering

More than half (51%) of SREs say they dedicate a significant amount of time to influencing architectural design decisions to improve reliability. This suggests progress on the road to SRE-driven engineering, supporting organizations’ efforts to improve reliability, resiliency, and security. Nevertheless, there is still a long way to go.

The most mature SRE practices have developers who have fought the battles and have the scars to show for it. They understand what it takes to build systems that can scale from a single user, to 1,000, or from 1 million to 10 million users. **Including these developers as part of the design process for new systems provides insight that enables architects to incorporate reliability from the beginning.**



The SRE view:

"SRE is a cultural change, which is ultimately about operating software systems better. What we're uncovering so far is unexpected at times, such as the need for simple and accessible documentation.

Documentation may not seem like an "SRE" thing, but when you build a practice around knowledge, if you don't put that knowledge down somewhere, you're going to run into all sorts of issues."



Stephen Townshend, SRE

iag

SRE best practice

Don't make reliability and resiliency an afterthought. Build a strong case for SRE principles to be incorporated into the design process (i.e., SRE-driven engineering).

Theme 3

Security is a core pillar of reliability

SREs are also making progress in extending DevSecOps approaches across their organization to ensure systems are restored quickly following the discovery of a vulnerability. More than two-thirds (68%) of SREs say they expect their role in security to become even more central in the future as organizations continue using third-party libraries for cloud-native application development. As we saw with the discovery of the Log4j vulnerability in December 2021, third-party code libraries can contain significant security risks, and SRE teams have a key role in ensuring those flaws are identified and eliminated quickly to protect their organization.

68%

of SREs say they expect their role in security to become even more central in the future.

Theme 4

SREs must be free to experiment

While more than half (52%) of SREs dedicate a large amount of time to designing experiments and tests to reduce the risk of production failure, only one in ten highlights this as their top priority.

Given the importance of experimentation to SRE, teams still need to make progress to ensure they have more time available for these tasks. For SREs to mature and deliver more strategic business value, engineers must streamline tasks that involve intensive manual effort.



Rising expectations and demands on SREs stretches their time increasingly thinly

Which of the following tasks do SREs in your organization dedicate the largest amount of their time to in an average week?

23%

Reducing MTTR

11%

Designing experiments and running tests to reduce risk of production failure

10%

Ensuring system reliability for spikes

12%

Influencing architectural design decisions to ensure reliability and scale

11%

Ensuring security vulnerabilities are detected and eliminated quickly

9%

Instrumenting applications to make them observable

12%

Building and maintaining automation code

10%

Designing experiments and running tests to scale user load



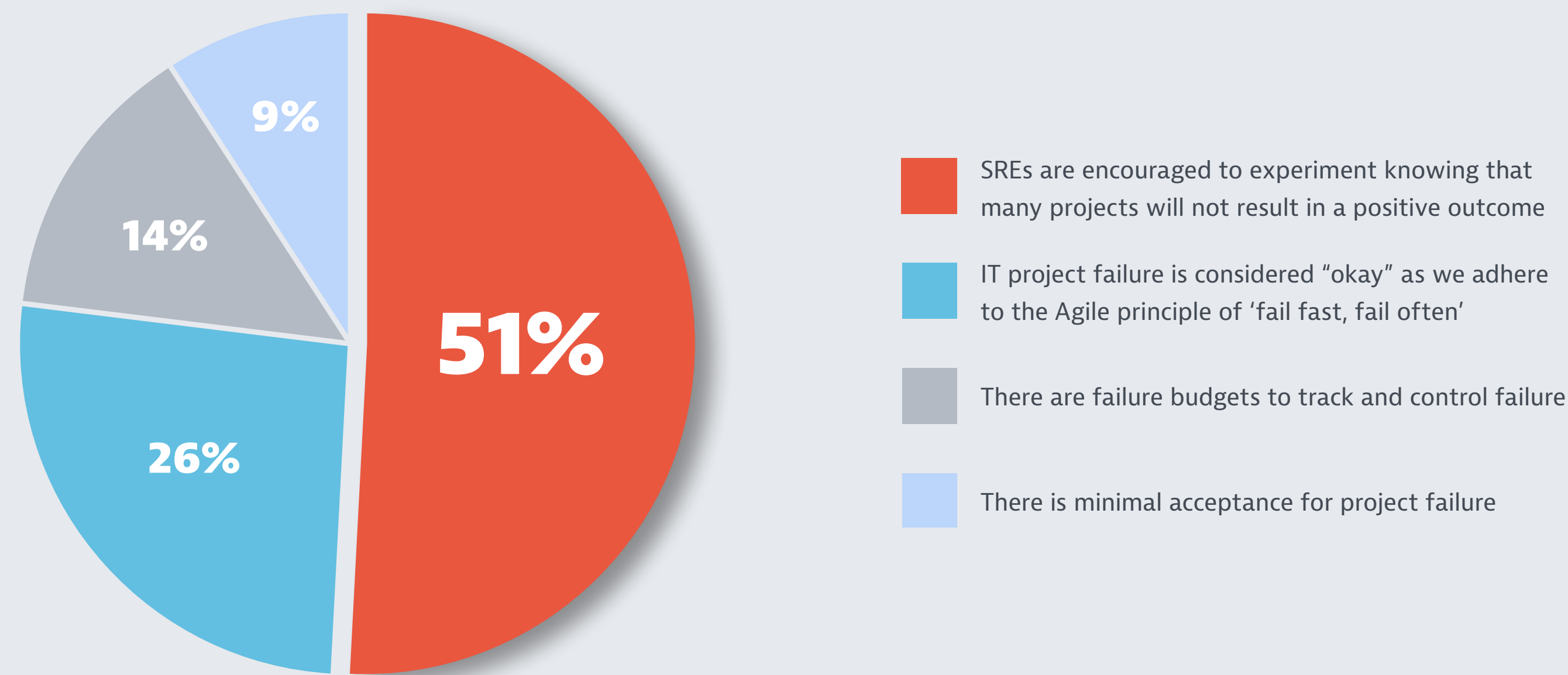
Theme 5

SREs need more license to prioritize strategic work

Despite it falling relatively low on the list of tasks they prioritize, 51% of SREs say they're encouraged to experiment, and project failure is seen as okay in a quarter (26%) of organizations. This reinforces that it's likely to be other pressures that distract SREs from focusing as much time as they'd like on experimentation. Organizations must, therefore, look to new strategies and solutions that reduce the need for SRE teams to perform less strategic tasks.

Organizational leaders also need to foster a culture that accepts failure and understands that "fail fast, fail often" provides the greatest competitive edge. To enable this, they need to unshackle SRE teams from the traditional goals that view IT as a cost-center .

How is project failure for SREs treated in your IT organization?



Theme 6

Reliability is recognized and rewarded

SREs must be free to challenge accepted norms and set new benchmarks for innovation-led design and engineering practices. Many organizations are making strides in this direction and have methods for rewarding the success of SRE teams. Nearly a third (31%) use hackathons to devise new ways of improving reliability, which offer prizes to winning SRE teams. These kinds of approaches will be key to encouraging a culture of experimentation that promotes the strategic value of SRE for the business.

How does your organization recognize and reward reliability?



76% have **specific bonuses/rewards** for hitting KPIs around reliability



44% give **special recognition** for the positive impact engineers have on the business beyond firefighting



31% organize **hackathons** to improve reliability and award prizes





CHAPTER 2

Role of the SLO

SLOs have become a guiding light for SREs

Organizations are realizing the value of going beyond basic measurements for service levels and setting goals that are based on meaningful metrics for the business. In addition to the focus on SLOs, more than half (58%) of SREs use [DevOps Research and Assessment \(DORA\)](#) metrics, which have emerged as an industry standard for identifying where improvements are needed in software development and delivery.

SREs are metrics driven

As SRE maturity progresses, teams need to focus on identifying gaps in the way they measure success; particularly when it comes to optimizing critical user journeys.

This will increase the importance of observability platforms that offer detailed insight into real-user experiences, so SREs can see beyond back-end performance monitoring data and understand what influences users' behavior to drive business success. Similarly, it can help them to identify and understand exactly what is draining an error budget and the rate at which it is doing so, as well as qualifying the overall impact those issues could have on a service.

4 Key Metrics

SRE is metrics-driven and its success depends on reliable metrics.

According to DORA these include (but are not limited to):



Deployment Frequency

How often an organization successfully releases to production



Lead Time for Changes

The amount of time it takes a commit to get into production



Change Failure Rate

The percentage of deployments causing a failure in production



Time to Restore Service

How long it takes an organization to recover from a failure in production

How does your organization evaluate service levels for its applications and infrastructure?

81% We set **objectives and key results (OKRs) and key performance indicators (KPIs)**

75% We set **service level objectives (SLOs)**

65% We use **service level agreements (SLAs)** from service providers

58% We use **DevOps Research and Assessment (DORA) metrics** (e.g., time to recovery and deployment frequency)

38% We use **simple monitoring tools** that are available

The SRE view:

"You can't have SRE without the SLO, it's that simple. **SLOs are the measure of reliability, the system, and the customer.** Understanding them is the quickest path to SRE maturity.

When the SLO becomes the method of measuring success, there's more parity between teams, and more feeling that we are all trying to achieve the same goal."



Michael Cabrera, SRE Leader

 **vivint**

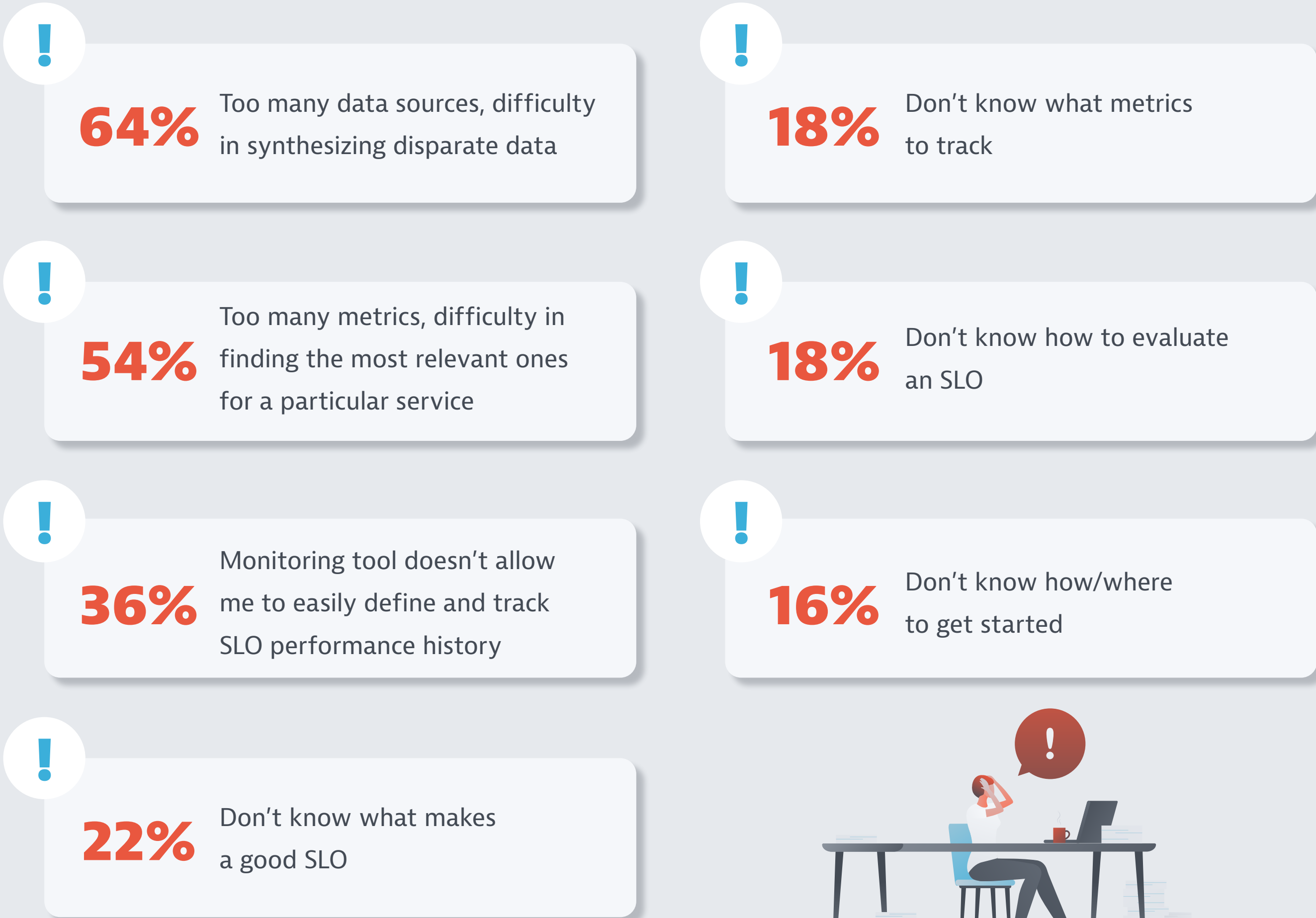
Data overload stands in the way of setting SLOs

Despite the growing use of SLOs 99% of SREs say there are challenges to defining and creating them. However, these challenges are mostly tactical, and therefore are relatively easy to solve with the right solutions in place.

For their more strategic challenges, SREs should invest time in keeping up to date with industry best practices through sources such as Google’s [SRE Handbook](#). Continually reviewing what competitors and peers are using as their benchmarks can help to develop a deeper understanding of SLOs.

99% of SREs say they encounter challenges when defining and creating SLOs.

What are the biggest challenges your teams experience to **define and create SLOs**?



Siloed teams and rising complexity make it difficult to manage SLOs

When it comes to defining and creating SLOs, SREs struggle with data overload. This typically results from the multitude of metrics and monitoring solutions teams use to manage applications and infrastructure, and the limited capabilities they provide to help SREs establish SLOs. It's not just setting SLOs that's the problem either — SREs also experience significant challenges in managing and evaluating SLOs once they've been defined. The use of multiple tools is a core source of frustration, alongside team silos, the prevalence of blind spots, and the need to correlate performance metrics with user-experience data. Manual evaluation of SLOs also leads to precious time being wasted and holds teams back from focusing more of their energies on innovation.

If they fail to address these issues, teams will continue to work in silos, which wastes time as they play “the blame game,” when error budgets are exhausted and SLOs are violated. It also makes it more difficult to set SLOs that are meaningful and viable, and implement an effective process to monitor, alert, and respond to violations. As a result, the core principles of SRE can end up being abandoned as resolution times increase, and it becomes more difficult to implement remediation plans before users are affected.

What are the biggest challenges that your teams experience to **manage and evaluate SLOs**?



SRE best practice

Implement continuous release validation where code quality is automatically and continuously evaluated against key SLOs as it moves through the delivery pipeline to prevent violations. This stops bad code in its tracks and allows developers to fix issues before it reaches production, reducing the need for manual intervention and remediation effort.

SREs need to unite teams around a single version of “the truth”

To overcome the challenges they face in defining, creating, managing, and evaluating SLOs, organizations should consolidate everything in a single observability platform that meets the needs of all key stakeholders, rather than using multiple monitoring tools. If this platform also has native SLO capabilities, organizations can avoid the dreaded scenario of adding yet another tool to their already bloated toolchains. That enables SREs to create a single source

of truth so they can easily monitor and track their error budgets, while managing their SLOs with greater accuracy and less manual effort.

It’s also important to ensure that SLO dashboards, error budgets, remediation plans, and alerting mechanisms are agreed upon, tested, and implemented in advance, to minimize the risk of a breakdown in collaboration when violations occur.



Choosing the right SLOs — getting started

When it comes to implementing SLOs, the biggest hurdle SREs face is figuring out where to start, and then identifying the metrics they should focus on. It's easy to go down a rabbit hole trying to identify the best approach, and it's vital to remember there's no one-size-fits-all methodology.

The most common pitfall is the temptation to take the path of least resistance, by just creating SLOs based on the service-level indicators (SLIs) already being captured. This approach is the simplest, but it's also highly ineffective.

A better path is to identify the business objectives and service-level agreements (SLAs) that SLOs need to meet, by asking what matters most to the business?

Four popular SLOs that organizations can adopt to get started include:

- **Availability** — Whether the service is available to users
- **User satisfaction (Apdex)** — The satisfaction level, based on the performance of the service
- **Error rate** — The ratio of failed requests to total requests
- **Crash rate (mobile)** — The crash rate across all supported devices

For organizations looking to mature their established practices, there are several other common SLOs to consider. It is, however, important to remember not all of these will be relevant for every organization, so SREs need to implement them on a case-by-case basis, with a clear understanding of how they will support the business. When it comes to SLOs, remember — less is more.



Common SLOs to consider

Business SLOs (End User Centric)



Availability

Is the service available for users?



Engagement

How engaged are the users?



Conversion

What is the rate of users reaching my business goals?



User Satisfaction (ApDex)

What is the satisfaction level on the performance of my app from 0 - 1

Performance SLOs



Utilization

Average time resources are busy servicing work



Success Rate

Ratio of success vs total requests



Response Time

Time it takes to service a request



Saturation

Resources that are most constrained



Traffic

Measure of how much demand is being placed on your system

Recommended SLOs for mobile apps

Let's look at an example for getting started with SLOs for mobile apps. SREs should combine a mix of business and performance SLOs to ensure they get the balance right and are measuring the things that matter most to the success of their app and its outcomes for the business.



App Adoption

Ratio of daily users vs total users



Availability

Rate of requests with a valid response



App Rating

Ratings based on Android or iOS Store



Response Time

Rate of login requests faster than 100ms



Crashes

Crash rate on officially supported devices



Success Rate

Rate of successful requests with HTTP 500

Don't "guesstimate" your SLOs

SREs use a range of approaches to identify the targets for their SLOs, with no clear accepted "norm" or established best practice. **Half of SREs noted their organizations have little methodology for how they set targets for their SLOs.** The most common approach is no more scientific than estimating the right target based on the requirements for end-user experience.

It is very difficult for most organizations to set SLO targets that have a tangible impact on the business. **Setting thresholds too high makes it unlikely they will ever be achieved, but setting them too low makes them meaningless as they fail to offer teams any incentive to improve service levels.**

It's essential that SRE teams adopt a more precise method to defining their SLO targets. For example, they could look for an advanced monitoring solution that guides them toward the right SLO thresholds based on historical data and industry standards. However, less than one in four (24%) organizations have embraced this approach. Clearly there is significant progress to make in taking SLOs in this direction. It's also important for SREs to consider best practices and the strategies of competitors and peers to ensure their organization remains at the forefront of the industry.



How do you identify the targets for each of your SLOs?

26% We estimated what felt like the right target **based on end user experience**

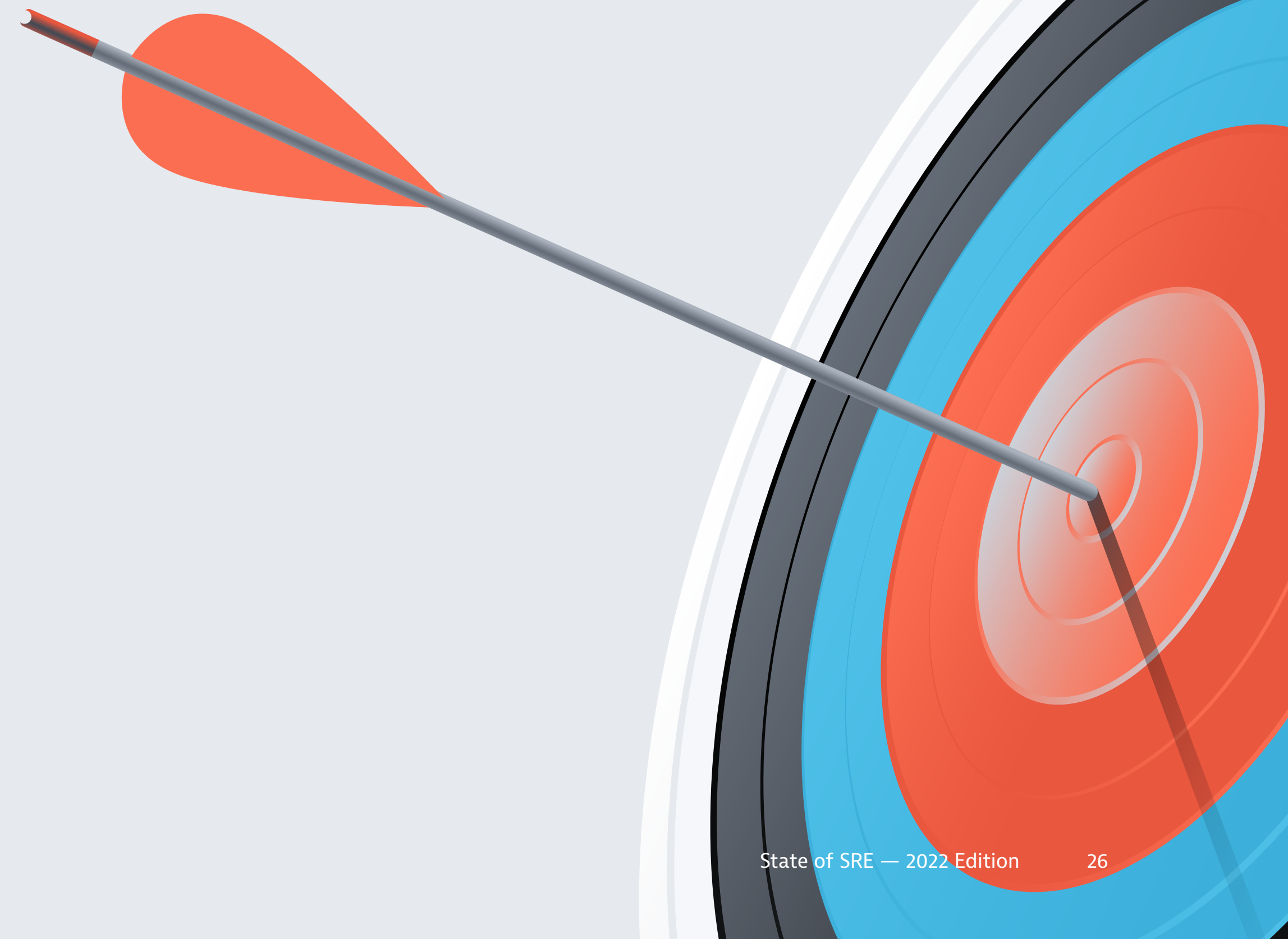
24% Our solution guides us toward smart SLO thresholds **based on historical data and industry standards**

20% We base them on **whatever our system is doing today**

18% They are **dictated by a senior manager** in the IT department

11% We estimated what felt like the right target to be aiming for **based on gut feel**

1% **We don't know** where the targets came from, we just need to adhere to them



The SRE view:

“SLOs are our semaphore and our thermometer. They tell us when we can do a change/deployment on our system and when we need to fix something that is broken or enhance something that isn’t good enough.

The key is to choose the proper SLI for an SLO. As in automation, if you use poor data to feed your SLO, you should expect poor results.”



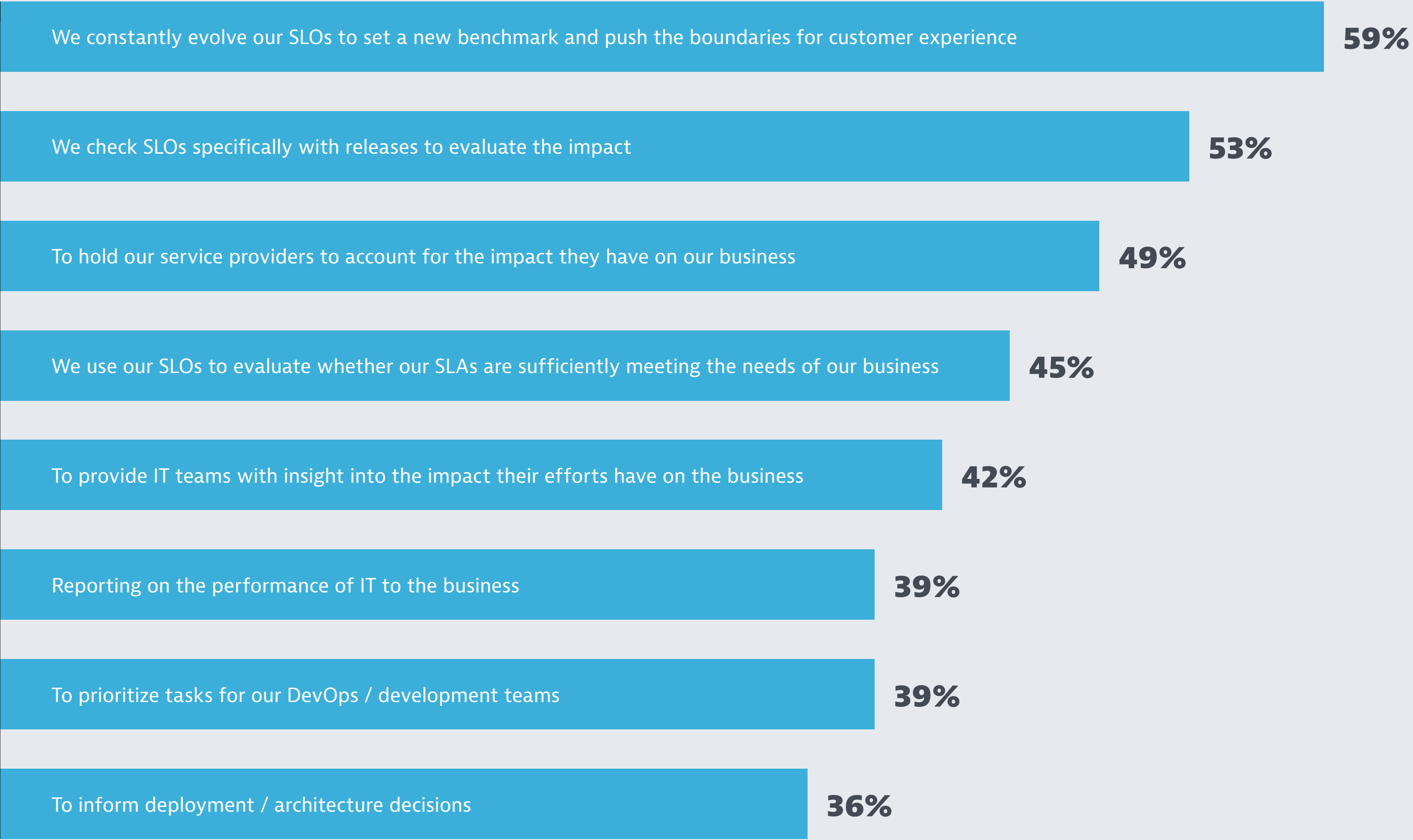
Danne Meira Castro, SRE
kyndryl.

SLOs are used for multiple goals

The use of SLOs is continuing to mature, as SREs harness them for a growing range of purposes that are central to their role and the success of the organization beyond ensuring SLAs are being met. However, there is still progress to make as there are no strong outliers in the data that indicate high levels of adoption of SLOs for any of the purposes identified.

How are SLOs used within your organization?

SLOs are becoming more strategically important, as they bleed beyond ensuring SLAs are being met into a multitude of other aspects of the business.



SRE best practice

Identify and prioritize the objectives that will create the highest impact for the business, and bring together stakeholders from business, development, and operations teams to establish SLOs that help to meet those goals

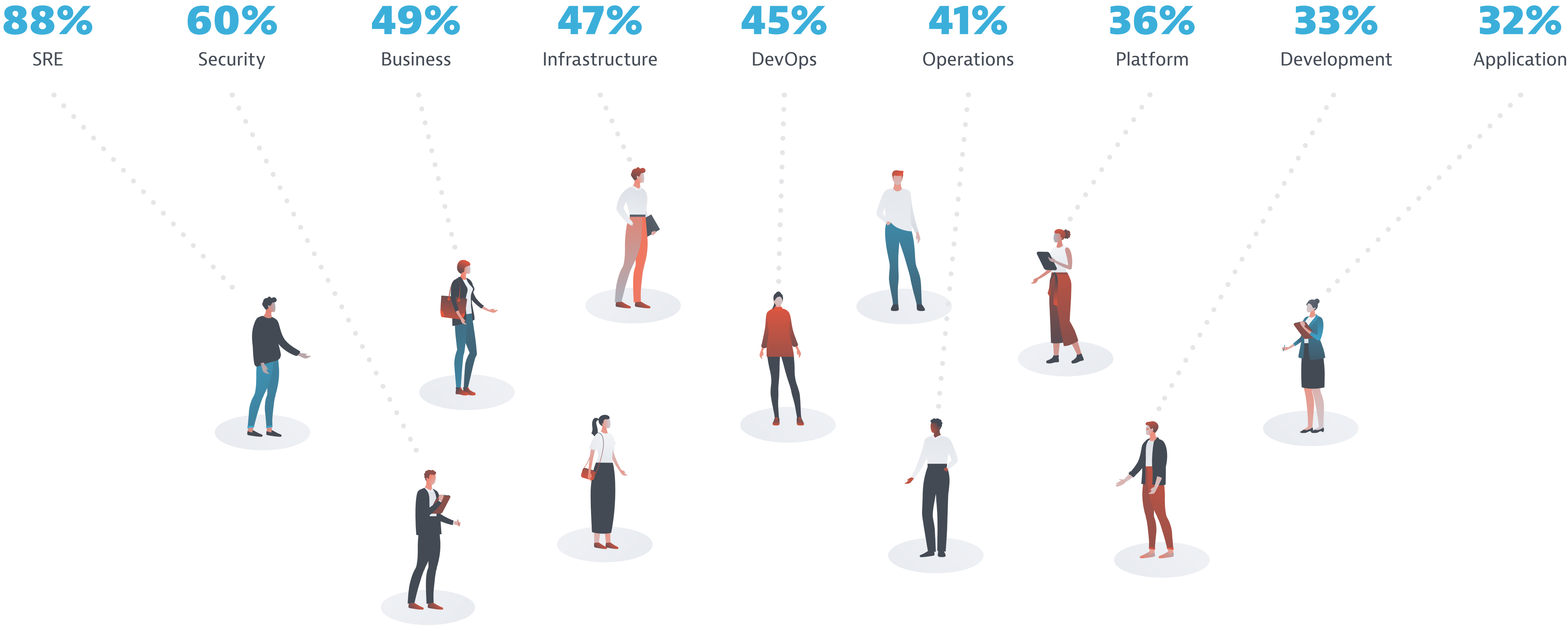
SREs are captain of a team game

Unsurprisingly, SREs say they have primary responsibility for SLOs, but multiple other teams are involved, with particular emphasis on security and business operations. This underscores the convergence of SRE with the move to DevSecOps, as organizations recognize the need to ensure their systems are both reliable and secure by default.

The shift toward these methodologies will be more successful in organizations that foster closer collaboration between business, development, security, and operations teams. The greater the collaboration between these teams, the more meaningful the SLOs they set will be, and the more effectively they can be evaluated to improve processes and outcomes for the business. This level of collaboration can be achieved only through a cultural change driven from the top, with senior IT leaders setting an example for others to follow.



Which team/s take responsibility for driving SLO adoption and managing SLOs across your organization?



SLO ownership needs to be clearly defined

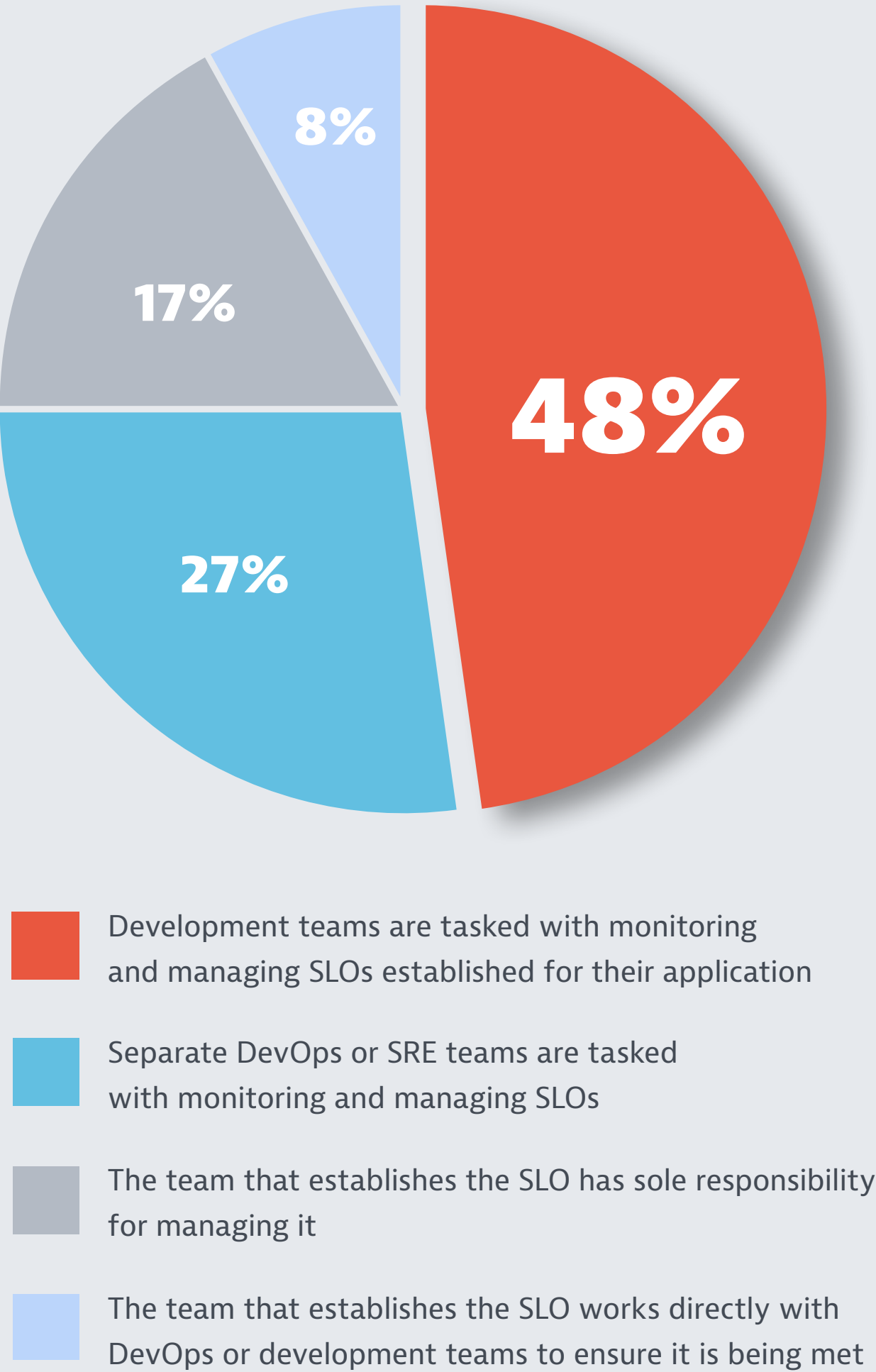
Once SLOs have been established, it is primarily development teams tasked with upholding them. While this makes sense for non-production applications, it is more appropriate for operations or SRE teams to be responsible for the SLOs in other environments.

This suggests uncertainty within most organizations about who should be responsible for SLOs, which presents a challenge for SREs. If other teams aren't aware of the importance of their own role in ensuring SLOs are met, it will be difficult to uphold them and drive SRE maturity across the organization.

Only 8% of SREs say the team that establishes the SLO works directly with DevOps or development teams to ensure it is being met. This indicates a reversion to passing responsibility to another team, rather than adhering to true DevOps best practices.

It's important to remember, there is no one-size-fits-all approach to who owns SLOs. Development, operations, and DevOps teams all have their own part to play, but it is up to SREs to guide them and ensure all teams are upholding the SLOs established for their environments.

How do the teams within your organization work to ensure SLOs are being met?





CHAPTER 3

Keys to SRE success

What are the barriers to SRE?

Despite the widespread adoption of SRE methodologies, the majority (97%) of organizations face barriers to implementing a dedicated practice. There are challenges around gaining access to the necessary skills, either by bringing in new hires or upskilling existing teams. This indicates a need for new approaches that reduce some of these barriers by enabling DevOps and developer teams to become SREs without additional specialized skills.

What do you perceive to be the most significant challenges to implementing an SRE practice across your organization?

59% Perception that it is hard to train/re-train existing ITOps/Sysadmin to become SREs

51% Perception that SREs are expensive and difficult to hire

43% Difficulty finding SRE skills in the market

20% Our IT organization doesn't have a culture to embrace modern IT principles

19% Not a priority for IT leadership to change the way of operating

97% of SREs say there are challenges to implementing an SRE practice in their organization.



Key 1

Open and extensible platforms are key to creating a unified toolchain that drives SRE success

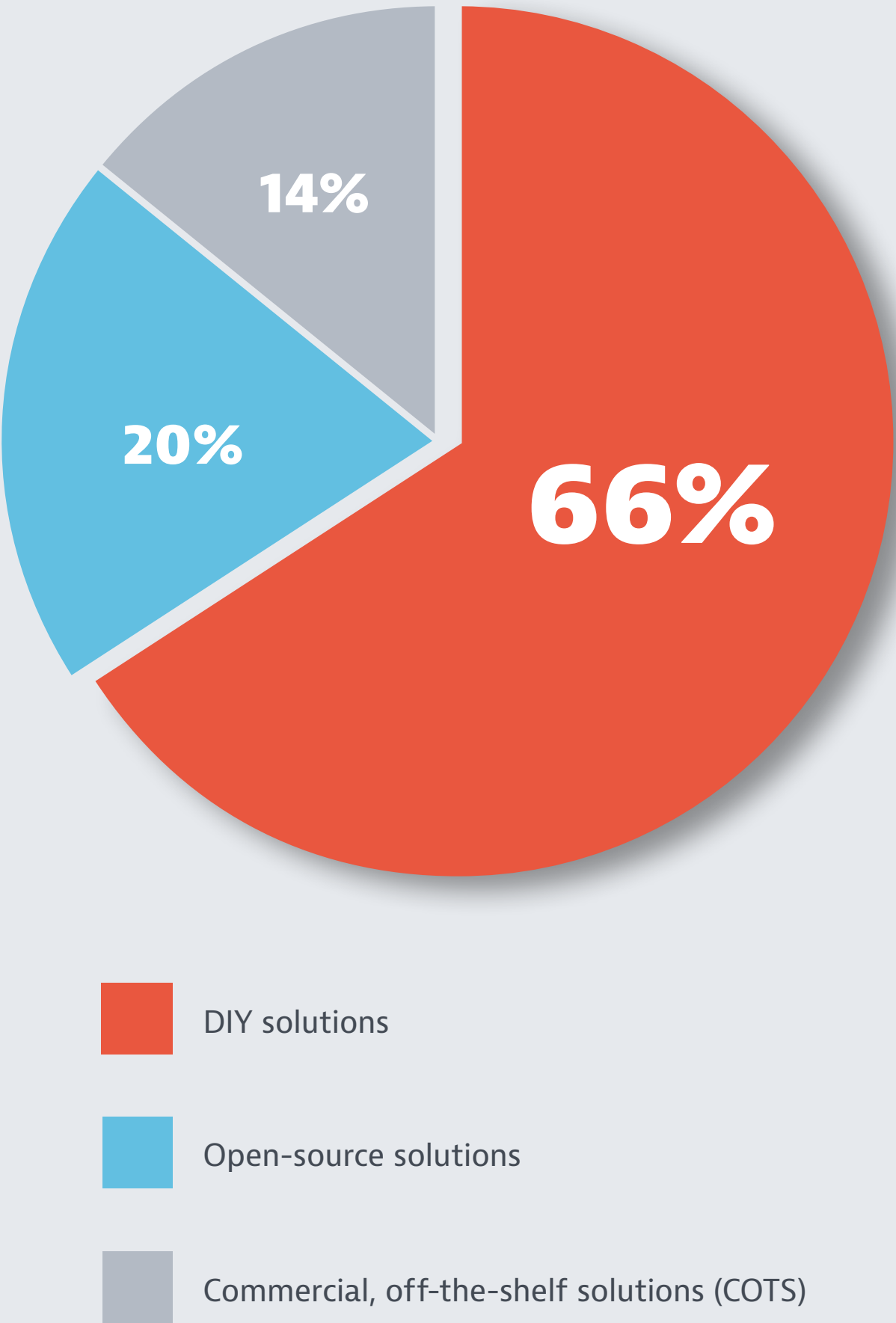
SREs largely rely on home grown, do-it-yourself, and open-source solutions to perform their tasks. This enables them to create a toolchain that’s purpose-built for their organization’s unique requirements. It also requires less of an upfront investment in new tooling and enables SREs to swap products in and out as their needs change and solutions advance.

The DIY approach, however, is difficult to scale, and can create problems over the longer term. These toolchains require a significant investment of time, manual effort, and specialist skills to maintain, which creates additional toil for SREs that distracts from core responsibilities. Commercial, off-the-shelf solutions also often prove ineffective, limiting the ability of SREs to benefit from open-source solutions.

Organizations, therefore, often find themselves in a difficult situation, as they need to divert their SREs to maintain the toolchains that were implemented to drive SRE practices. Hiring more SREs usually isn’t an option due to their scarcity, so organizations must find another way.

The most effective approach is to reduce the SRE toil of maintaining a toolchain, so teams can focus on the activities that are more core to their role and drive greater value for the business. Organizations should look for a platform-based solution that supports an open ecosystem, with the ability to seamlessly integrate with whichever tools their SREs, architects, and developers prefer to use, and orchestrate the data in a single place. Platforms with self-service everything-as-code approaches will significantly reduce toil for SRE teams, so they can scale quickly across the organization.

Solutions that are most prevalent in SRE toolsets



SRE best practice

Look for a solution that provides end-to-end observability and is based on a single data model, to ensure automation can be driven with precision.

Key 2

Automation is key to reducing SRE toil

Unsurprisingly, organizations are focused on how automation can ease the burden for developers and SREs.

Teams seek to automate the resolution of security vulnerabilities and application failures, highlighting the accelerating drive toward application self-healing. Observability will be critical to this goal, providing the data needed to drive automation with precision. Combining this data with runtime vulnerability management will also

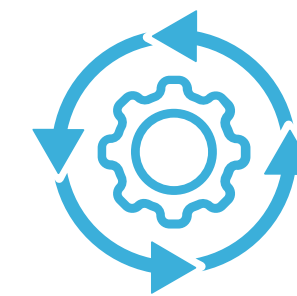
be critical, giving teams the ability to always know what is running in production and apply AI to prioritize the vulnerabilities that pose the biggest threat to the business.

If they can achieve these objectives, organizations will reduce a significant amount of toil for their developers and SREs by removing the need to invest time firefighting, so they can focus on work that drives greater value for the business.

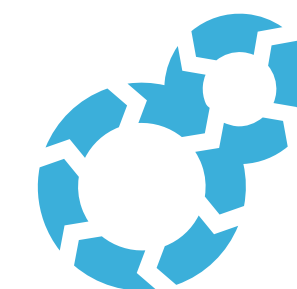
85%

of SREs say their ability to scale SRE practices further across the organization will be extremely dependent on the availability of automation and AI capabilities.

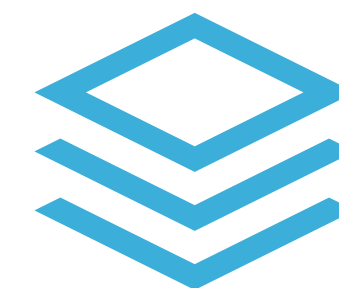
What is your organization doing to reduce toil for your developers and SREs?



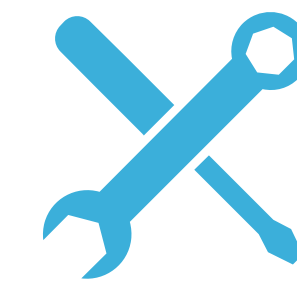
71% are increasing the use of automation across every part of the lifecycle



58% are increasing the use of automation in the CI/CD pipeline

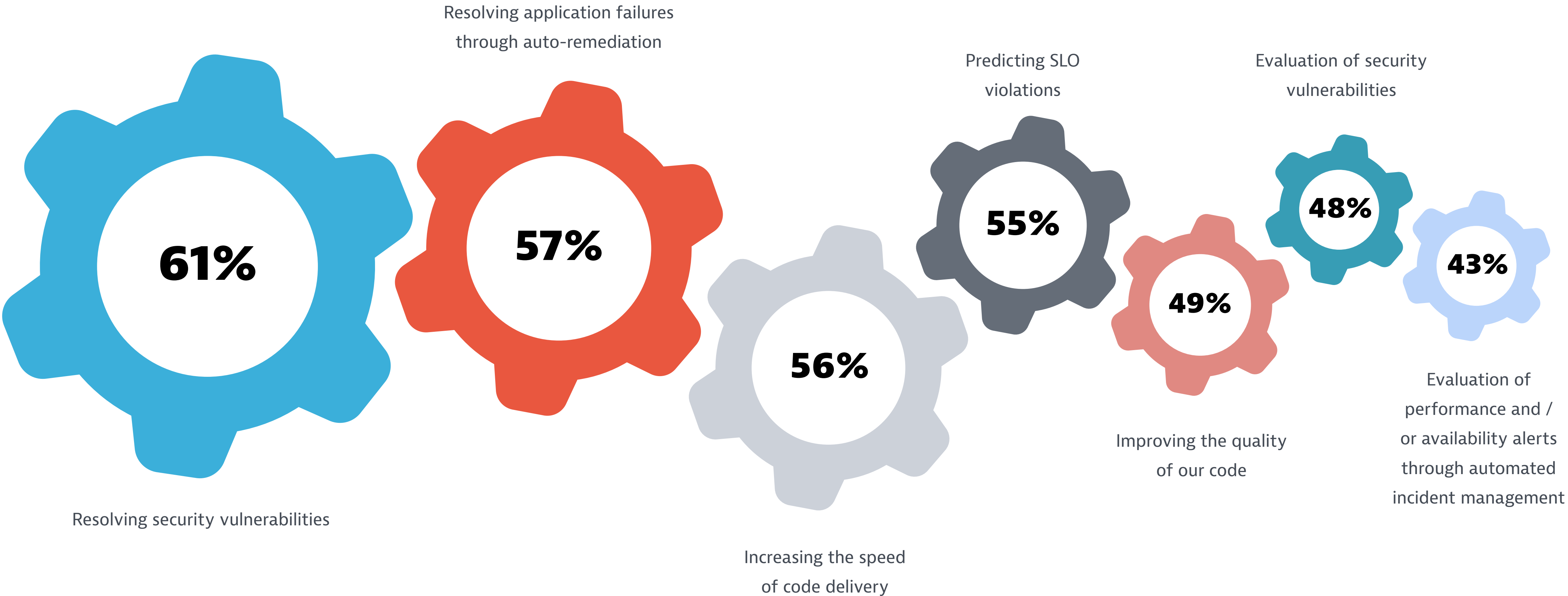


58% are consolidating tool stacks



46% are modernizing tool stacks

In which of the following SRE tasks are your teams currently using automation to support their efforts?



Key 3

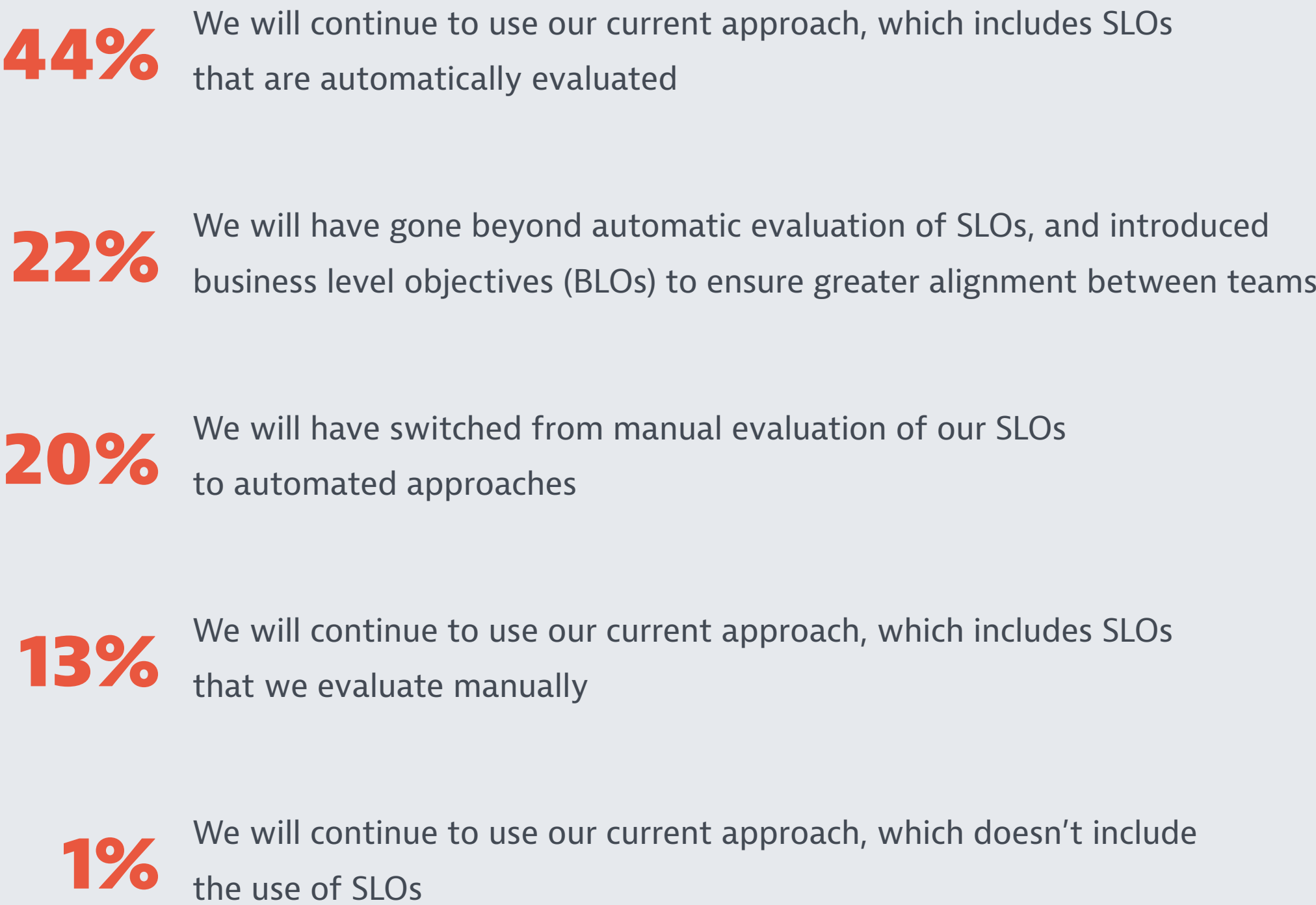
The future of SLOs will be automated

Automation will also play an increasingly central role in the way SREs manage and evaluate service levels through their SLOs in the future. This strategy will reduce manual effort for developers, DevOps, and SRE teams, freeing them up to focus on experimentation and continued innovation.

We will also see more organizations adopting business-level objectives to tie their success back to more meaningful metrics such as user satisfaction, because every second of downtime affects revenue and erodes brand. These approaches will increase SRE maturity even further.



How do you expect your approach to measuring service levels to have evolved by 2025?



SRE best practice:

Make AIOps a core pillar of your SRE strategy — but don't treat it as a bolt-on. Point solutions deliver only limited value; AIOps must be built into the solutions and platforms developers and engineers rely on.

Key 4

AIOps is at the heart of SRE maturity

In addition to their focus on automation, SREs see AIOps becoming increasingly critical to their role in the future, highlighting several significant benefits. SREs are looking to AIOps to help reduce toil even further and enable them to make more data-driven decisions around how they prioritize their time to drive the best outcomes for the business.

This indicates a growing maturity in SRE, as automation and AI help it to become laser-focused on meeting the needs of the business and its customers, by reducing toil and enabling teams to focus on making faster decisions.

68%

of organizations are increasing the use of AIOps across every part of the lifecycle to reduce toil for developers and SREs.

The SRE view:

"AIOps platforms empower SREs to move from a reactive to a proactive posture toward application-impacting incidents. As a result, SREs can respond more quickly to slowdowns and outages, with a lot less effort and toil."



Andrzej Gebski, SRE



How significant an impact will AIOps have on the following SRE practices?

- 64%** Enabling teams to automate more of the processes that are critical to ensuring service levels are continually met
- 63%** Enabling teams to prioritize the problems that have the biggest impact on service levels and user satisfaction
- 62%** Enabling teams to prioritize security vulnerabilities to minimize service downtime
- 62%** Eliminating the need for operations teams to 'run the business' entirely, so those assets can be redeployed to focus on improving development and engineering practices
- 61%** Enabling teams to predict SLO violations before they happen
- 59%** Freeing up more of DevOps teams' time to focus on SRE activities such as chaos engineering

Key 5

SREs bring teams together with unified solutions

Organizations are also looking at how they can update their tool stacks to create a more streamlined solution that enables SRE and DevOps teams to work more effectively.

This highlights the growing drive toward unified solutions that eliminate the need for teams to switch between different dashboards. These solutions provide a single source of truth that teams can unite behind, supporting their ability to work toward the shared goals that SRE advocates.

85% of SREs want to standardize on the same observability platform from Dev to Ops and security by 2025.



The SRE view:

“Observability is the fundamental basis for all SRE. Without it, you cannot measure success or identify areas for improvement.”

— Mario Biemans, SRE

Conclusion

SRE is a core pillar of the modern digital business. As the world has gone increasingly digital, reliability is a critical success factor when every second of downtime leads to lost revenue, declining share prices, and lasting reputational damage.

While there is general agreement that SRE is here to stay, we are only at the beginning of the journey and many organizations' practices remain relatively immature.

At a time when demand far outstrips supply of skilled engineers, organizations should do everything in their power to amplify SRE efforts. They need to enable SRE to shift further left, becoming more deeply ingrained in engineering and architectural design practices.

Despite this, we've seen that manual toil and unnecessary effort in tasks that aren't central to their role are a major distraction for SREs, which is holding SRE back in the early stages of maturity.

Automation is a major factor in overcoming this hurdle, but it can create more problems than it solves without the right strategy and approach. It's crucial to recognize that not all automation is created equal – there is good and bad automation.

If SREs are tied up writing automation scripts and duplicating them across multiple processes, it just moves the manual effort elsewhere, without reducing the burden. To be effective, SREs need a platform that enables them to drive reliability and automation by default, through self-serve and everything-as-code capabilities.

In this way, SREs can enable developers across the organization to easily build in critical capabilities to the services they create, from observability, to testing to establishing meaningful SLOs, and application self-healing.

As a result, teams can be free to focus on the things that are core to the role of an SRE; delivering greater value to their organizations by driving best practices that maximize reliability, resiliency, security, performance, and ultimately, business outcomes.

Methodology

This report is based on a global survey of 450 SREs in large enterprises, including 150 in the U.S., 150 across EMEA, and 150 in Asia Pacific.

The research was conducted by Coleman Parkes and commissioned by Dynatrace.

In which country are you primarily based?	
US	33%
UK	7%
Ireland	3%
France	3%
Sweden	3%
Norway	2%
Germany	6%
Netherlands	4%
Denmark	3%
Finland	4%
Australia	7%
Singapore	3%
India	17%
Malaysia	2%
Thailand	2%
New Zealand	3%

In which sector does your company primarily operate?	
Industrials, Manufacturing & Transport	14%
Financial Institutions	13%
Healthcare & Life Sciences	12%
Technology, Media & Telecommunications	12%
Consumer Goods & Retail	10%
Mining & Infrastructure	8%
Professional services	8%
Oil & Gas	8%
Hospitality, travel and tourism	8%
Public Sector (local and national government and education)	6%

What is your company's global annual revenue for the last financial year, expressed in US dollars equivalent?	
\$750 million - \$1 billion (875)	20%
\$1.1 billion - \$5 billion (3050)	27%
\$5.1 billion - \$10 billion (7550)	16%
\$10.1 billion - \$15 billion (12550)	18%
Over \$15 billion (15000)	19%
How many employees does the company have globally?	
Between 1,000 and 3,000 (2000)	12%
Between 3,000 and 5,000 (4000)	10%
Between 5,000 and 10,000 (7500)	23%
Between 10,000 and 20,000 (15000)	19%
Over 20,000 (25000)	36%

About Dynatrace

Dynatrace provides software intelligence to simplify cloud complexity and accelerate digital transformation. With automatic and intelligent observability at scale, our all-in-one platform delivers precise answers about the performance and security of applications, the underlying infrastructure, and the experience of all users to enable organizations to innovate faster, collaborate more efficiently, and deliver more value with dramatically less effort. That's why many of the world's largest enterprises trust Dynatrace® to modernize and automate cloud operations, release better software faster, and deliver unrivalled digital experiences. Learn more at dynatrace.com