

2023

# State of DevOps Report

Platform Engineering Edition

Presented by



**puppet**  
by Perforce

Sponsored by



Strategy

Show Get

# Contents

<b>Executive Summary</b>	<b>3</b>
<b>Introduction</b>	<b>5</b>
<b>Platform Engineering and Platform Teams</b>	<b>9</b>
<b>What the Research Tells Us</b>	<b>14</b>
<b>A Promising Future</b>	<b>23</b>
<b>Organizational Design</b>	<b>35</b>
<b>Platform Teams at Global Scale</b>	<b>41</b>
<b>In Conclusion</b>	<b>42</b>
<b>Methodology</b>	<b>43</b>
<b>Who Took The Survey?</b>	<b>44</b>
<b>Author Biographies</b>	<b>46</b>

# Executive Summary

## 1. Platform engineering is a proven approach to unlock DevOps success at scale in the enterprise.

*Organizations that have adopted platform engineering are experiencing significant improvements across standardization and process, helping them to better realize the benefits of DevOps.*

- An overwhelming majority (94%) agree that the concept is helping their organizations better realize the benefits of DevOps.
- The majority (68%) of respondents are experiencing an increase in development velocity, with 42% reporting that speed of development has improved “a great deal.”
- The majority of respondents are experiencing a wide variety of benefits, ranging from improvements in system reliability (60%) to greater productivity/efficiency (59%) to better workflow standards (57%).

## 2. When done well, platform engineering benefits the entire organization.

*From developers to operations to security to senior IT business leaders, platform engineering substantially benefits both business and technical objectives.*

- An overwhelming majority (93%) of respondents believe the adoption of platform engineering is a step in the right direction, and 44% believe it is a big step in the right direction.
- Respondents are most likely (30%) to state that platform engineering best serves the needs of the entire company – not just a single department.
- 37% of respondents at firms with platform engineering report being “very satisfied” with the efficiency of their product delivery process compared to just 20% of respondents at firms without platform engineering.

### **3. Organizations are waking up to the benefits of platform engineering.**

*The longer an organization has implemented platform engineering, the better the results and the higher the confidence, encouraging firms to hire rapidly for platform team roles.*

- At firms in which a platform team has been in operation for less than three years, 56% of respondents report they are “completely confident” or “very confident” that the platform team will continue to meet its goals. For respondents at firms in which a platform team has existed for more than three years, this rises to 74%.
- Most (53%) firms that have been practicing platform engineering for more than three years report speed has improved “a great deal,” compared to just 35% of firms that implemented platform engineering within the last three years.
- 71% of respondents report their firm plans to hire people with platform engineering experience over the next 12 months, with the majority (53%) planning to hire within the next 6 months.

### **4. Organizations are under-investing in product management skills on their platform teams.**

*A product manager role on the platform team is crucial to cultivating a product mindset, evangelizing the platform, and effectively disseminating information throughout the firm.*

- Only 66% of respondents somewhat or strongly agree that their platform team has a product owner who evangelizes the platform services and capabilities.
- Respondents' top priorities for the platform team are in line with product management responsibilities, including increasing awareness of platform capabilities (47%), setting realistic expectations for the platform team’s role across the organization (44%), and more closely following industry trends and keeping up with feature requirements (37%).
- However, firms are more likely to be focusing on other roles aside from product management. Indeed, 48% of roles fall under “DevOps,” followed by 45% software engineers, and 44% platform engineers.



## Introduction

Welcome to the 2023 State of DevOps Report, our 11th edition! For more than a decade, we've reported on DevOps as it has emerged from a grassroots, practitioner-led trend into a massive movement of adopters, experts, vendors, and consultants. It's undeniable that DevOps has changed the IT industry for the better, with a significant uptick in automation, infrastructure as code, measurement, collaboration, and systems thinking, all helping us deliver better software, more quickly. It's become so prevalent that many organizations don't even use the term "DevOps" anymore – they've internalized all its lessons. This is simply how they work.

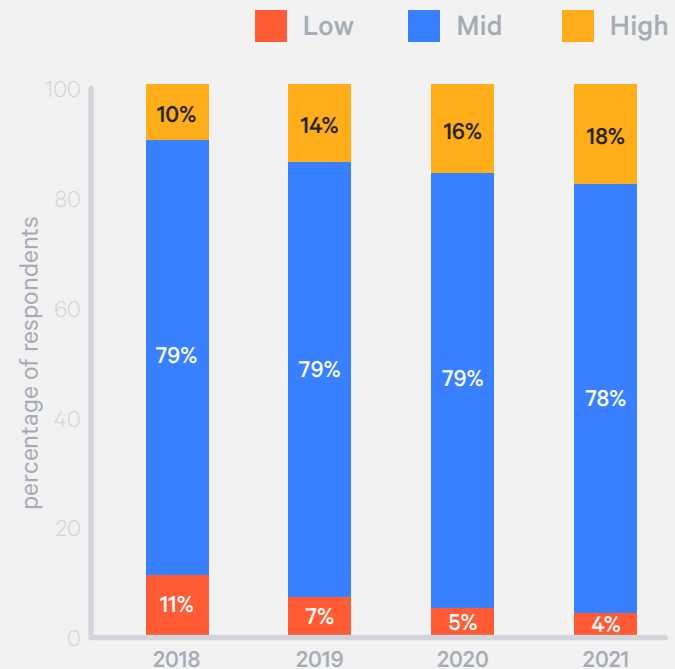
At the same time, there is a huge variation in what DevOps actually means, particularly inside large companies – so much so that last year we recommended no longer using the term “DevOps team” at all, as it’s become meaningless.

Given this vast spectrum between teams that are so “DevOps” that they don’t use the term at all, and teams that use the term “DevOps” but work in a way that runs counter to the spirit of the movement, we decided it was fruitless to continue to survey the entire DevOps landscape, and that we should instead focus on platform engineering, a trend we’ve been tracking for a number of years in our reports, and that has been rapidly gaining traction as a way for larger companies to improve software delivery at scale.

In our most recent [State of DevOps Report](#), we found that despite the prevalence of DevOps practices across organizations, nearly 80% remain in the middle of their DevOps journey, experiencing varying degrees of success at the team level but not across the entire organization.

### DevOps evolutionary levels

Over the last four State of DevOps surveys, the number of respondents that identify as “highly evolved” firms has grown; however, the amount of organizations in the middle level has remained stagnant.



### What is Platform Engineering?

Platform engineering is the discipline of **designing and building self-service capabilities to minimize cognitive load for developers and to enable fast flow software delivery**. Platform teams deliver **shared infrastructure platforms to internal users** responsible for delivering a value stream – typically software developers and engineers. The platform team continuously develops, builds, maintains, and supports underlying infrastructure in order to build self-service solutions, enabling development teams to deliver quickly, and providing consistency to the rest of the organization. Platform teams **must adopt a product mindset**, ensuring that they have feedback loops with their users and that they're building systems that solve the problems their users face. The same principles and processes that apply to product management – user research, product roadmaps, soliciting feedback, iterating, launching, maintaining, and marketing – are in use by highly effective platform teams.

Last year's research showed that simply adopting automation and infrastructure as code does not create a highly evolved DevOps practice. Rather, a focus on improving organizational structure, team identities, and interaction paradigms between teams are the common attributes of highly developed DevOps organizations.

A clear pattern emerged: enterprises with more mature DevOps practices tend to use platform teams. This doesn't mean you must adopt a platform team model to be good at DevOps; rather, it's that a platform team is a well-defined and proven path to succeed with DevOps at scale – *particularly* in the enterprise.

In fact, last year's research identified platform team adoption as a *key differentiator* between those toward the higher end of firms in the middle of their DevOps journey evolution from those toward the lower end, with 65% of those toward the higher end using self-service platforms compared to only 40% of those toward the lower end.

Why do platform teams make such a difference to organizations of a certain scale and complexity? Do we have a common understanding across the industry as to what platform engineering and platform teams actually are? What are the critical attributes of a highly functional platform team? These questions are central to our focus this year, and we hope that you find this report useful if you're embarking on or considering a platform engineering approach.



## Acknowledgements

We've chosen to stop doing reports on the broad concept of DevOps, yet we believe that the large body of research we've produced over the last decade still holds true to help people adopt DevOps or improve their practices. Going forward, expect to see reports like this one: shorter, more focused on specific topics within DevOps, but with the same depth of research and reporting you've come to expect from our prior work.

For every person who completed the 2022 State of DevOps survey (which maps to the 2023 State of DevOps Report), we donated to the National Coalition for the Homeless and the World Central Kitchen. Our partners this year are Continuous Delivery Foundation, Humanitec, and Women in DevOps.



# Platform Engineering and Platform Teams

## Platform Engineering is Not New and DevOps is Not Dead

Building digital platforms as a way to deliver software at scale is not a recent invention, and it predates the emergence of the DevOps movement in the mid-2000s. Many large tech companies whose primary business was building software realized decades ago that by standardizing infrastructure, building self-service interfaces for developers, providing increasingly higher level abstractions, and dedicating a team to maintaining all of this as a platform enabled developer teams to build, ship, and operate applications more quickly and with higher quality. However, these organizations had to build these platforms entirely from scratch, as many of the core DevOps tools around infrastructure as code did not yet exist. These companies also didn't have problems with masses of legacy IT, and it was far simpler for them to mandate the use of the platform internally.



Enterprises have attempted to replicate this same approach for a long time, with self-service catalogs, PaaS, and opening up controlled access to virtualization platforms, with varying degrees of success. The burdens of legacy IT, a higher degree of commercial off-the-shelf software, a blinkered focus on project-based funding, and the difficulty of enforcing mandates across a much more varied internal landscape all posed significant challenges.

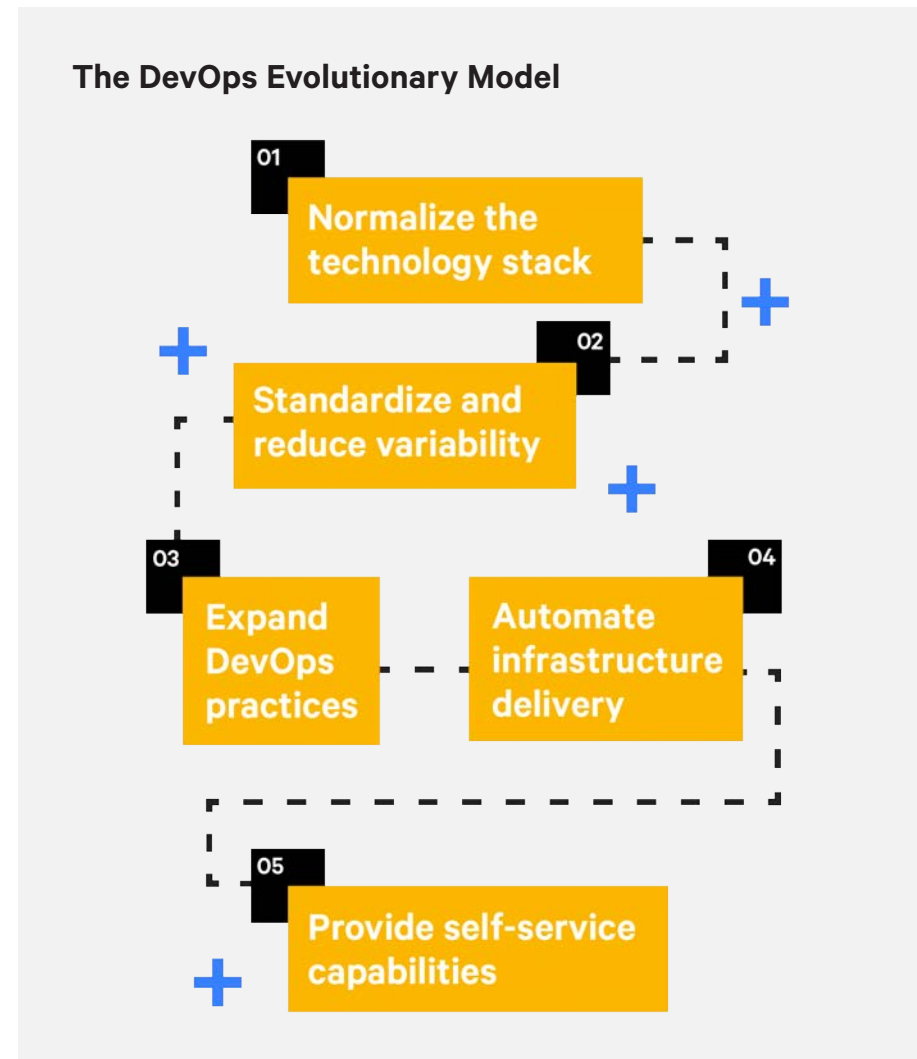
The platform approach is not new, but the **surge in popularity** is, particularly in the enterprise space. Given how many large companies have struggled to experience the benefits of DevOps across their organizations, and that this more prescriptive movement of platform engineering is proving to deliver value quickly, some have argued that “DevOps is dead” and that modern platform engineering has supplanted it.

We firmly disagree with this position. DevOps has always borrowed from other movements such as Agile, Lean, Safety Science, and The Toyota Way, and platform engineering is just another one to add to the list.

DevOps is about using automation as a tool to align incentives and increase collaboration across all of the teams involved in the software delivery lifecycle in order to deliver software better, more quickly, and with less stress. Modern platform engineering has taken the already existing platform approach and added an explicit focus on treating the platform as a *product* rather than as a *project*, as well as clear guidance for where teams should interact via collaboration, and where they should interact via self-service interfaces.

In 2018, we first identified that DevOps success within the enterprise required significant standardization on the way to providing self-service as part of our five stages evolutionary model. Platform engineering, with its prescriptive approach to organizational design and dynamics, is proving to be a viable path to delivering on this model.

Like DevOps, platform engineering makes heavy use of automation, focuses on collaboration, requires empathy across organizational functions, and keeps people rather than technology front and center. It's perfectly aligned with DevOps, and is proving to be a viable way for many enterprises to do DevOps at scale, in highly complex and varied environments.



## What Does It Mean Today?

There are several core attributes of a modern platform engineering approach. The most overarching one is having a product mindset, by which we mean that your platform is treated as a set of self-service products that continuously evolve to meet developer needs. The platform aims to reduce developer cognitive load and enables fast-flow software delivery, where the teams responsible for building value streams can deliver value to their consumers without requiring direct interaction with other teams.

For IT platforms, this approach begins with delivering underlying infrastructure services as a self-service product that are consumed by developer (or “value stream”) teams. The design and evolution of the platform should be informed by collaborative organizational learning practices, and the usage of the platform internally should be driven by internal evangelism, not via a top-down mandate. All of this means that your platform team needs to have development, operations, product management, and product marketing skills, no matter the actual job titles within the team.

A significant mindset shift for many enterprises is that while the platform team must have IT operations expertise, they do not *operate* the applications on the platform itself. They are responsible for delivering a reliable and resilient platform that empowers value stream teams to build, release, *and operate* their own applications.



*It's impossible to overstate the significance of the [Team Topologies](#) work on this topic, and it has strongly influenced us, which is why we were overjoyed to have Matthew and Manuel contribute to the 2021 report. Across their books, courses, and blog posts, they provide an excellent and in-depth set of resources to optimize your software organization for fast-flow software delivery with low developer cognitive load. We cannot recommend their work highly enough.*





## The Cool New Kid in Town

All of this leads us to the question of why we are seeing such a rapid rise in popularity of the platform engineering approach today. The answer? Because it appeals to dev teams, ops teams, security teams, and senior IT business leaders. For dev teams, it promises to let them work at their own cadence and focus on the application rather than dealing with underlying infrastructure. For ops, delivering via self-service promises to help them deal with ever-increasing scale of demands. For security teams, a consistent platform promises a smaller surface area to secure, and for business leaders, it promises easier IT governance.

Even better, our data this year shows that this approach actually appears to be delivering on all of those promises relatively rapidly. Most (51%) organizations that have adopted it did so within the last three years, and 93% of respondents declared that platform engineering adoption is a step in the right direction.

Ultimately, this year's data points to a rapidly increasing trend that platform engineering helps complex organizations realize the kinds of benefits that the DevOps movement has been aiming for all along.



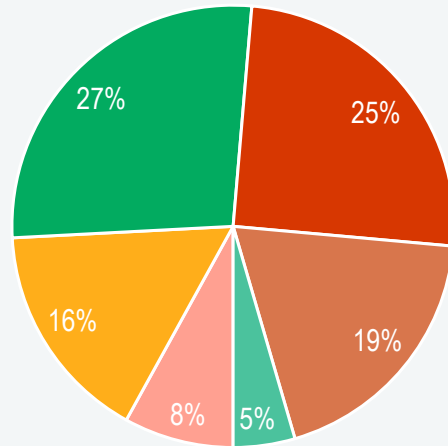
# What the Research Tells Us

## Platform Team Adoption

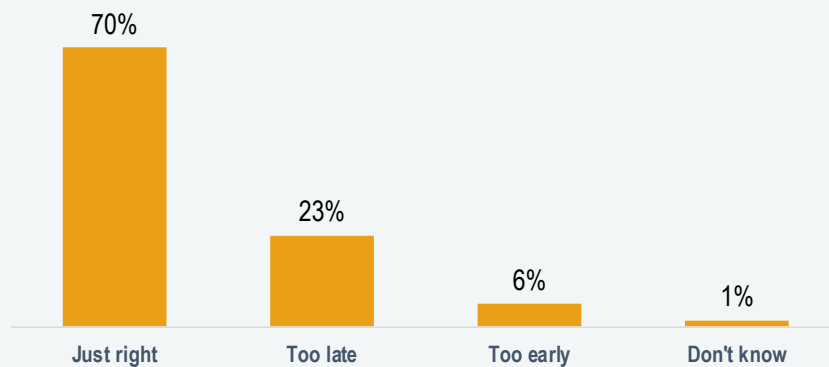
While teams have been implementing platforms for years, the formation of platform teams is relatively new within the history of DevOps, with a plurality (27%) reporting that platform teams at their organization were formed within the last two to three years. A full 8% of respondents report that platform teams have been implemented within the past year, while twice that (16%) report this happened within the last one to two years.

## How long has there been a platform team at your organization?

- Less than 1 year
- 1 to 2 years
- 2 to 3 years
- 3 to 5 years
- 5+ years
- Don't know



## Was the platform team created at the right time, too early, or too late?



To put it another way, only 19% of respondents report that platform teams were established more than five years ago. Successful large-scale platform teams have predominantly been the domain of relatively new Big Tech companies, and – despite what you may feel if you work in tech – there aren't that many of those compared to the much larger set of enterprise companies. Every company relies upon software, but not every company is a software company. Our data shows that tech companies are more likely than others to have adopted platform engineering five or more years ago.

For many firms, the time is now for adoption. In fact, 70% of respondents believe that their timing of platform engineering adoption was “just right,” while nearly a quarter (23%) believe it was too late. Only 6% claim it was too early, underscoring how few teams have regrets about adoption.



## Why Platform Engineering?

We've discussed the ways in which platform engineering benefits the entire organization, from developers to operations to security to senior IT business leaders. This is reflected in the data from this year's question on what drives adoption, which we can classify into three categories – business outcomes, team and technical needs, and top-down guidance – furthering the stance that platform engineering is beneficial to the entire organization.

It comes as no surprise that the most common driver is the need to increase speed of delivery (22%) followed by the need to scale (18%), as businesses seek new ways to achieve greater efficiency and growth. Ten percent of respondents report that engineers had been taking on too much work, while another 10% cite external guidance as the catalyst for adoption. Nine percent point to the desire to control infrastructure spending; another 9% report that different product teams had been duplicating work; 8% attribute the adoption to the product catalog expansion; and another 8% report that it came from the top down, with a newly hired senior leader advocating for platform engineering.

*Last year's data showed that top-down enablement of bottom-up transformation was required for DevOps to succeed. This may be even more profound for platform engineering as its very nature requires organizational change in terms of how teams are organized, and often also requires a shift away from time-limited, project-based funding.*

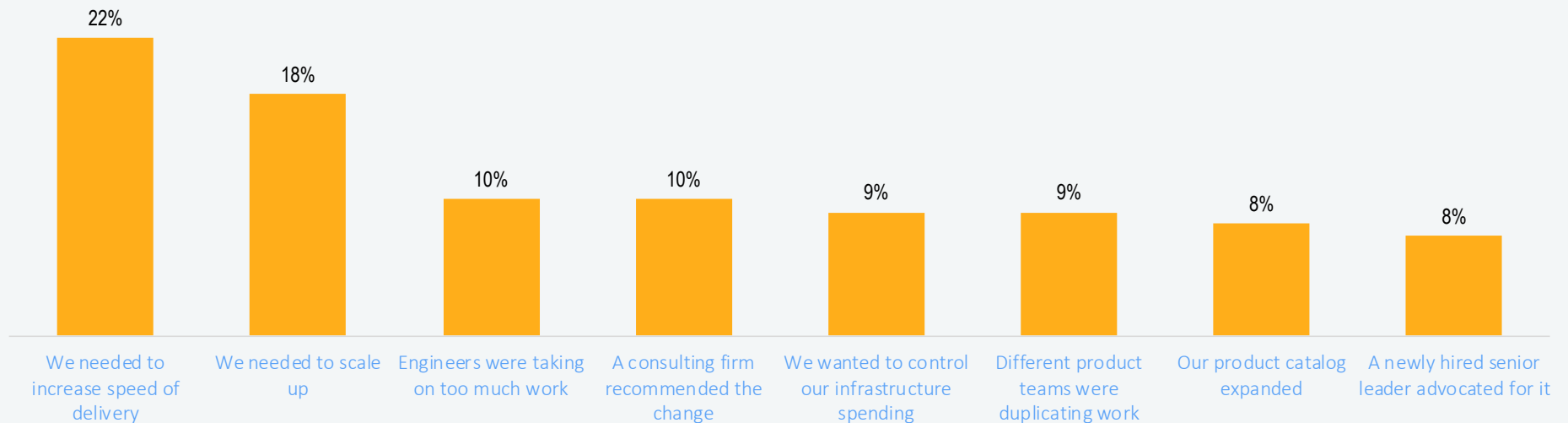


While we can categorize these responses into the three buckets listed above, the responses do not fall neatly into these three classifications. Rather, there is distinct overlap across several of these drivers. For example, the need to increase speed of delivery is both beneficial to the business and to the team; the desire to scale is driven both by business needs and by technical ambition. Again, platform engineering is good for business and good for teams.

For the majority of respondents (54%), problem solving is the platform's primary goal, which likely reflects how platform teams aim to prevent other teams from reinventing the wheel by solving common problems time and again.



### What led to the creation of a platform team at your organization?

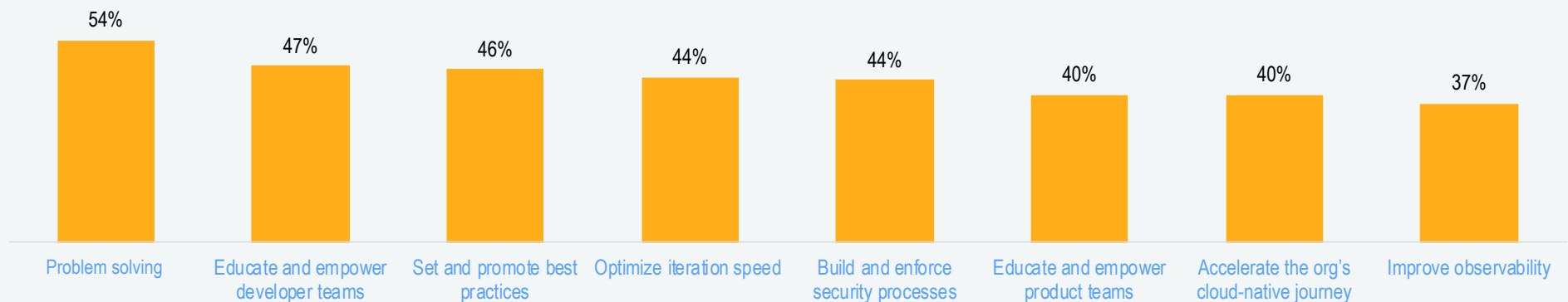


Notably, respondents are very much focused on enabling the users of their platforms, with 47% citing the education and empowerment of developer teams as the goal, and 46% citing the setting and promotion of best practices. This people-centric approach follows in the footsteps of DevOps, and as our prior reports have shown, a focus on tech over people is what leads to enterprises getting stuck in the middle. Streamlining, organizing, automating, and guiding how work gets done – in a way that continues to meet developer needs – helps the entire organization move more quickly and more accurately to meet customer expectations and drive competitive edge.

Respondents also cite speed and security (both 44%) as platform engineering goals, paralleling common priorities in the tech industry today.



### What are the key goals of the platform team?



## What's Being Done in the Name of Platform Engineering?

Platform engineering is successful in part because it has a much clearer definition and more apparent path to implementation than DevOps. But what goes into a platform varies from firm to firm.

When asked which services should be included in the platform engineering scope of work, the majority of respondents cite two capabilities that highlight how platform teams and DevOps operate in the same realm: 1) the automation of workflows (59%) and 2) building and managing infrastructure (59%).

Close behind these top-tier services are “services that enable application development teams to build, deploy, and run their applications,” once again underscoring the collaborative ecosystem of DevOps. Merely providing developers with the ability to provision infrastructure is a necessary step, but it's not sufficient. The goal must be to fully enable those teams to independently ship value to *their* users.

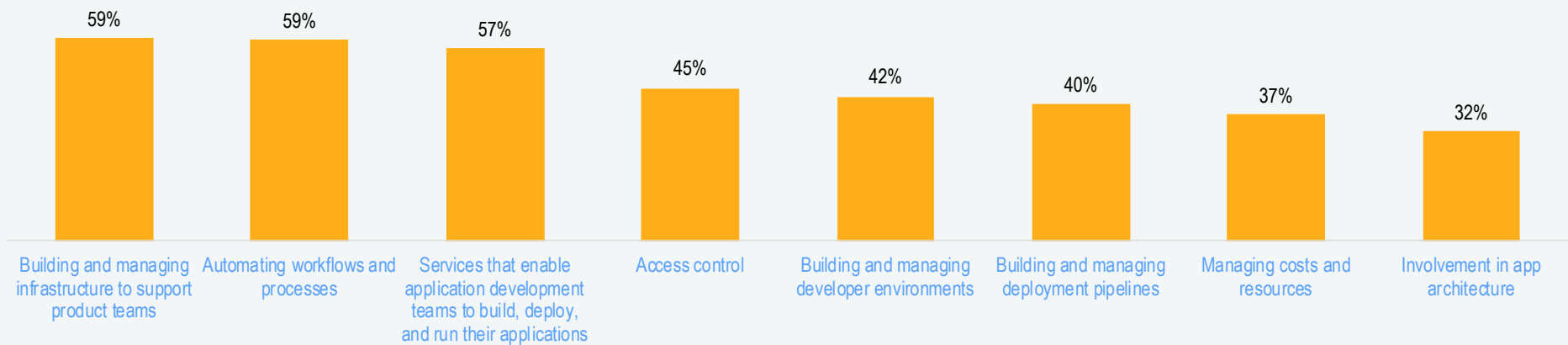


We believe trends will emerge here over time, and predict that the bulk of efforts will move up the stack over the next few cycles of research. Many firms do not provide highly functional internal infrastructure-as-a-service to all of their developer teams, so it stands to reason that building and managing infrastructure would be a top priority service now. As these platforms evolve and continue to focus on solving the next most important set of problems for development teams, we will see the platform capabilities include higher order functionality well above the infrastructure layer and much closer to the realm of application development. These features may not be novel to tech companies whose entire business is writing software, but they are new to enterprises in other sectors.

For enterprises in particular we expect to soon see an increased focus on network automation and providing self-service access to firewall changes and load balancer configurations. It's far too common in the enterprise for users to be able to request virtual machines on demand, yet wait weeks for network changes to be approved and implemented, even for software-defined networking environments.

We want to shout it loud and clear to enterprise leadership that platform teams require ongoing investment. These are not something to build and walk away from; they are not cut out for project-based funding. Your organization must continuously invest in order to continue to reduce developers' cognitive load and increase flow of delivery.

### What services should fall within the platform team's scope of work?

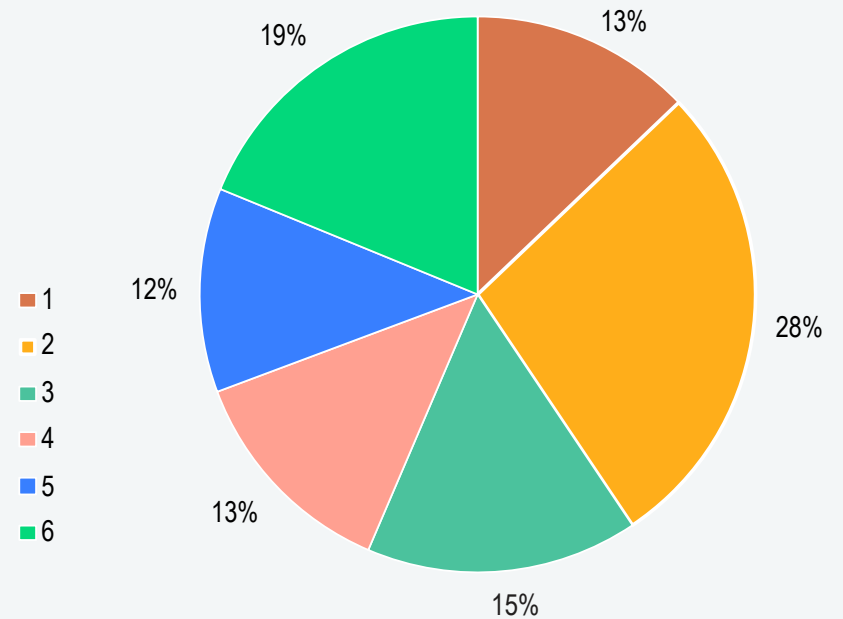




Platform teams can emerge from different areas of the organization: organically from value stream developer teams looking to share solutions with each other, from particularly advanced and capable operations teams with high organizational credibility, or from a top-down approach by progressive management. We believe that the correct number of platforms for an organization depends upon context, and to automatically demand that all services be jammed into one platform is the wrong approach; this will inhibit progress.

The majority (57%) of respondents report that they have between two and four platforms, while 30% have five or more, which indicates that platforms are emerging from multiple areas of the organization.

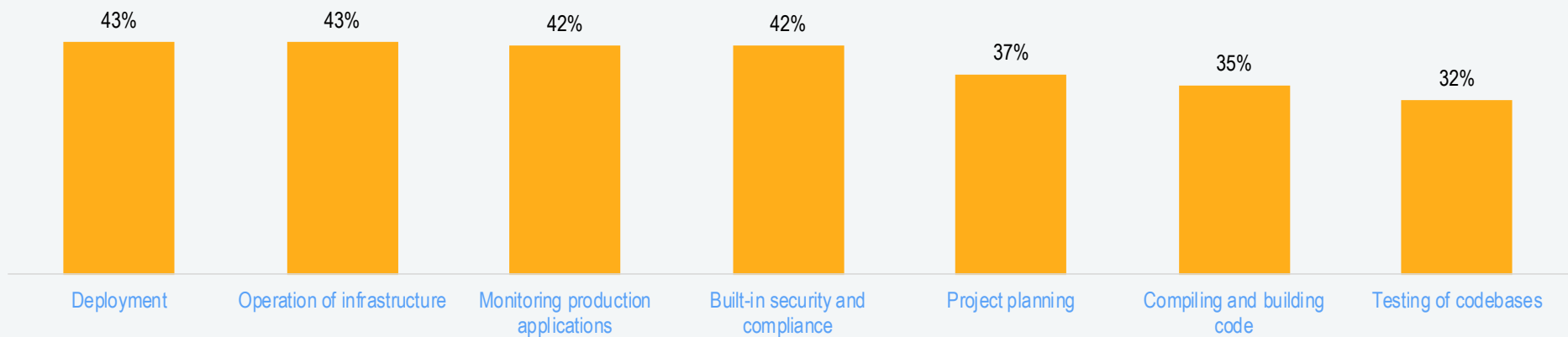
### How many internal self-service platforms are within your organization?



Respondents report multiple capabilities offered by their internal self-service platforms, such as deployment (43%), operation of infrastructure (43%), monitoring of production applications (42%), and built-in security and compliance (42%). This comes as no surprise, as deployment, operation, monitoring, security, and testing are the bread and butter of self-service platforms. Teams must get these things right before advancing to the higher levels of abstraction. It will be fascinating to see what teams build next on top of these.



### What capabilities does the self-service platform offer?

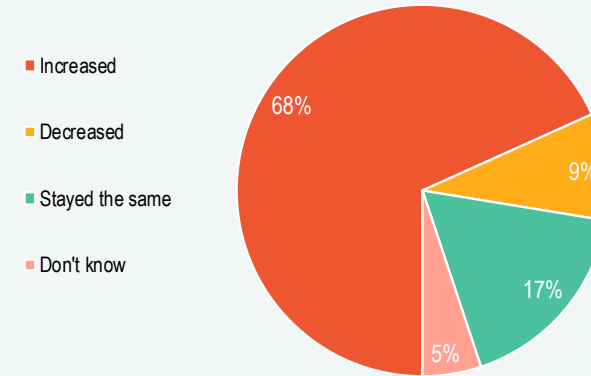


## A Promising Future

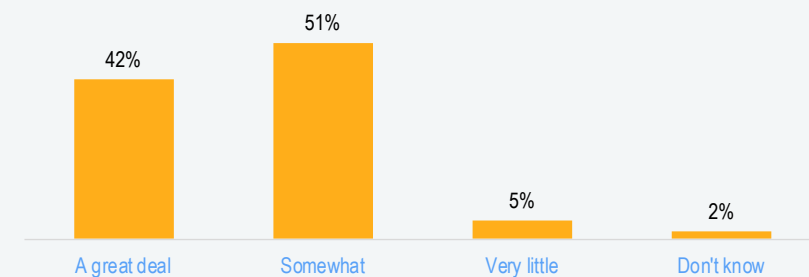
The results of the platform engineering concept are hugely promising. The positive impacts are noticeable for many, with most respondents (68%) already experiencing an increase in development velocity; 42% of respondents report that speed has improved “a great deal.”

The data shows that a key factor in improved development speed is how long the organization has had platform teams in place – most (53%) firms that have been practicing platform engineering for more than three years report speed has improved “a great deal,” compared to just 35% of firms that implemented platform engineering within the last three years. Even so, to create change in large organizations is difficult, and to see returns in the first few years is notable. Some companies have spent decades trying to become Agile without success, and many have failed to achieve organization-wide DevOps success over the last five years.

### Did the platform team have a direct impact on development velocity?



### By approximately how much did development speed increase after the inception of the platform team?





If you have struggled to expand DevOps success beyond the team level, then platform engineering may offer the right kind of prescriptive path to solving that problem.

DevOps has always talked about culture as a core pillar, but in our report last year we suggested that talking about “improving culture” inside large organizations was neither useful nor actionable, and that focusing on reducing obstacles to fast flow software delivery and lowering cognitive load for developers would result in the kinds of “cultural” changes most organizations are looking for.

Platform engineering offers a well-defined path to reducing those obstacles across the organization.

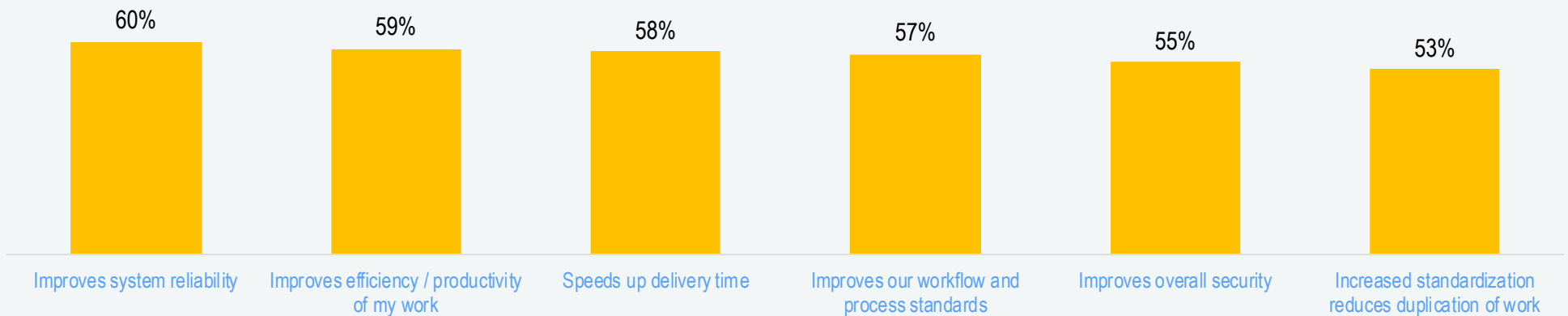


## The Organization-Wide Benefits of Platform Engineering

Platform engineering is not solely about improving development speed. Rather, platform engineering has a broad impact on organizations, with most respondents citing benefits across multiple areas, such as improving system reliability (60%), productivity (59%), workflows and process standards (57%), overall security (55%), and standardization, resulting in less duplication (53%).

The significant share of respondents who cite security improvement as a benefit of platform engineering is particularly interesting in light of the DevSecOps movement that emerged out of DevOps, as the industry realized that DevOps was about much more than just operations and development. Standardizing infrastructure and processes as part of a platform makes it much easier and more efficient for security teams to build or collaborate on self-service functionality.

### What are the benefits of platform engineering?





*Respondents are in widespread agreement (94%) that the adoption of platform engineering is helping their organization realize the benefits of DevOps.*

Overall, teams that have adopted platform engineering report much higher levels of satisfaction about their processes than those who have not begun the journey. Most notably, respondents at firms with platform engineering in place are more likely to identify as “very satisfied.”

- 35% of respondents at firms with platform engineering are very satisfied with internal IT operations resources, compared to just 18% of those without.
- 32% of respondents at firms with platform engineering are very satisfied with processes and practices for IT operations, compared to just 20% of those without.
- 37% of respondents at firms with platform engineering are very satisfied with the efficiency of the product delivery process, compared to just 20% of those without.

Ultimately, we believe platform engineering delivers satisfaction because it reduces cognitive load for developers, and reduces the overall burden on operations teams. It’s that simple! Respondents are in widespread agreement (94%) that the adoption of platform engineering is helping their organization realize the benefits of DevOps.

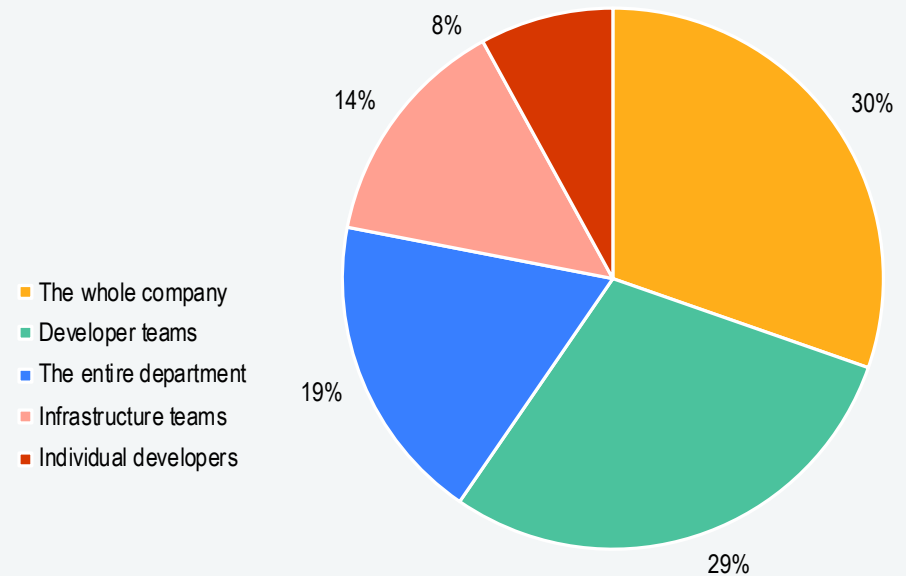


One of the challenges for large firms is to create global optimization across all of the teams involved in software delivery, not just optimization at the team level. It's striking to see that nearly a third (30%) of respondents report that platform engineering best serves the needs of the entire company. With the platform team focus on improving developer experience, it's not surprising to see that 29% report that developer teams are best served by the platform. This is followed by the entire department (19%), infrastructure teams (14%), and individual developers (8%).

Some of our infrastructure team readers may bristle at the low score here, but this is a good result. We've never run infrastructure for its own sake (except in our home labs); it has always been so other people can deliver value on top of it, and this focus on the end goal is great to see.

Evolution is a necessary attribute of a healthy platform engineering approach. It takes time to build effective platform teams and for those results to ripple across the firm. More than a third (36%) of organizations that have been practicing platform engineering for more than three years report the whole company benefits most, contrasted with only 25% of those who implemented it within the last three years. This reinforces the importance of setting realistic expectations within the company during the early years of platform team adoption.

### Whose needs are best served by platform engineering?

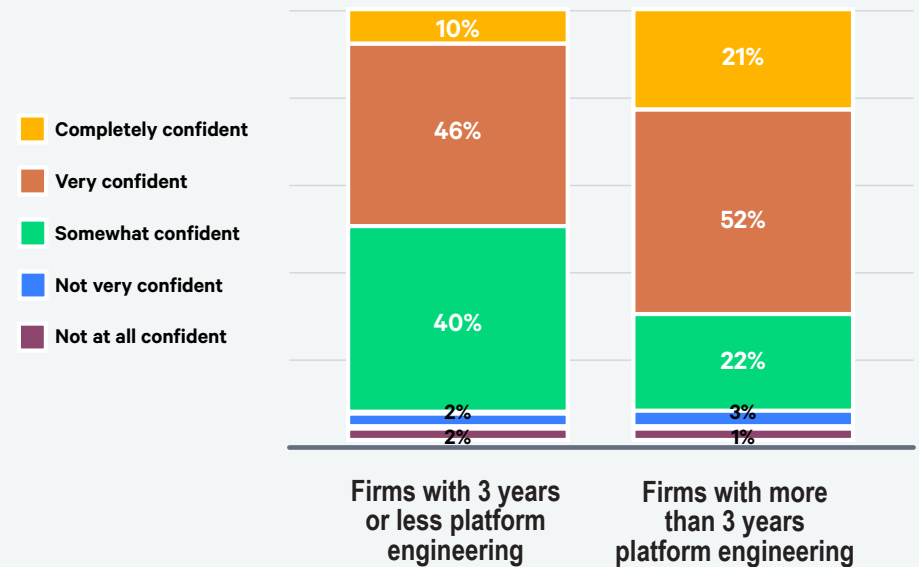


## Platform Team Evolution

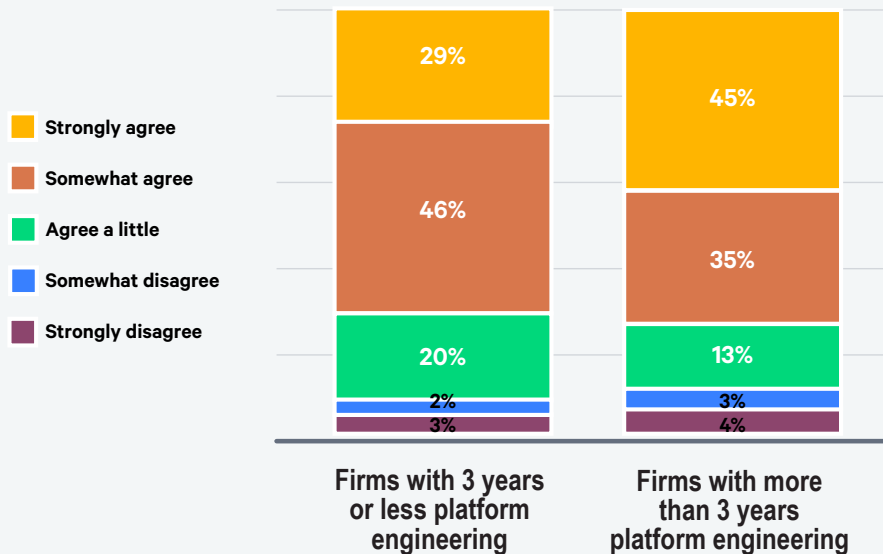
The longevity of a platform team directly correlates to the confidence it instills in survey respondents. Given how cynical our industry can be, we find it notable that respondents display a high degree of confidence in platform engineering's ability to meet organizational goals regardless of how long the platform team has existed, but it's even more telling that this confidence increases over time.

At firms in which a platform team has been in operation for less than three years, 56% of respondents report they are “completely confident” or “very confident” that the platform team will continue to meet its goals. For respondents at firms in which a platform team has existed for more than three years, this rises to 74%.

### How confident are you that the platform team can continually meet their goals?



### Platform engineering is helping my organization realize the benefits of DevOps.



The level of confidence in the platform team correlates with the findings mentioned earlier – that a key factor in improved development speed is the length of time a firm has had a platform team in place – and this makes sense. (As we noted above, 53% of respondents at firms in which platform teams were implemented more than three years ago report that development speed has improved “a great deal,” compared to just 35% of respondents stating the same at firms that implemented platform teams within the last three years.) The longer a platform team has been implemented, the more quickly development happens, instilling confidence in the platform team’s abilities.

This theory is also supported by the fact that those who’ve had platform teams at their firms for more than three years are more likely (45%) to “strongly agree” that platform engineering is helping them realize the benefits of DevOps, compared to those at firms where platform teams have been implemented for less than three years (29%).

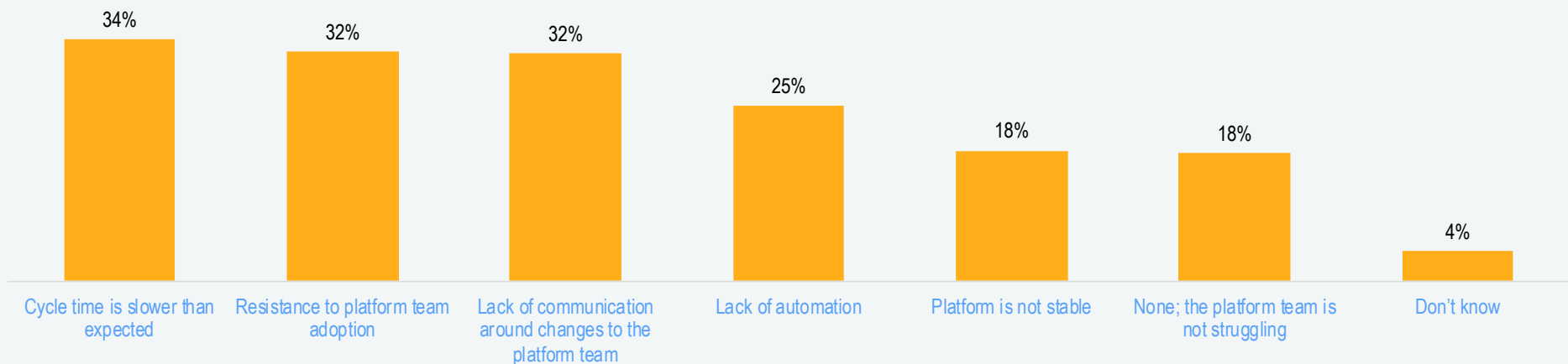


Interestingly, the data shows the longer a platform team has existed at a firm, the stronger the opinion there should be a product manager on the platform team. At firms in which a platform team has existed for three or fewer years, 60% of respondents identify a need for a product manager; at firms in which a platform team has existed for more than three years, this number rises to 74%. We believe this is because the initial capabilities of an internal developer platform (namely infrastructure provisioning and automation) are relatively obvious and well understood across the organization. Once those capabilities are in place, identifying and prioritizing the next most valuable capabilities is a much more difficult problem – and one that requires the expertise and dedication of a product manager.

We don't want to pretend that platform teams are a panacea for every organizational challenge, however. A substantial number of respondents (34%) report that cycle time is slower than expected. Given that the next two most common areas where respondents believe platform teams are struggling are "resistance to platform team adoption" (32%) and "lack of communication around changes to the platform team" (32%), we believe this is further evidence to suggest that organizations are under-investing in communication and evangelism.

Interestingly, these issues are almost equally prevalent at firms that have adopted platform engineering within the last three years and at those who implemented it earlier, indicating that time alone will not solve these issues, and an explicit focus on treating your platform as a product is required.

### What are the platform team's biggest pain points?

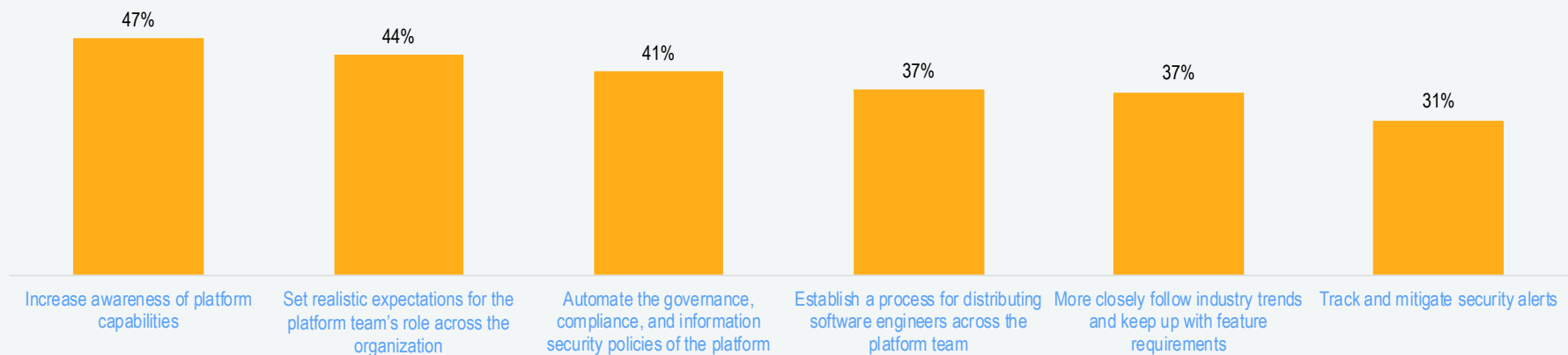


## Where Are the Platform Product Managers?

Treating your platform as a product is a key tenet of modern platform engineering, yet while this is generally well understood, the data tells us that organizations are under-investing in product management skills on their platform teams. Delivering on organization-wide benefits and evangelizing platform capabilities while maintaining a resilient and reliable platform operationally requires someone to be doing product management, no matter their actual job title.

The good news is the findings show a high degree of self-awareness about this whole set of problems. If we look at the following top priorities for respondents, we find a clear cry for active product management of the platform. Respondents' priorities are in line with product management responsibilities, including but not limited to increasing awareness of platform capabilities (47%), setting realistic expectations for the platform team's role across the organization (44%), and more closely following industry trends and keeping up with feature requirements (37%).

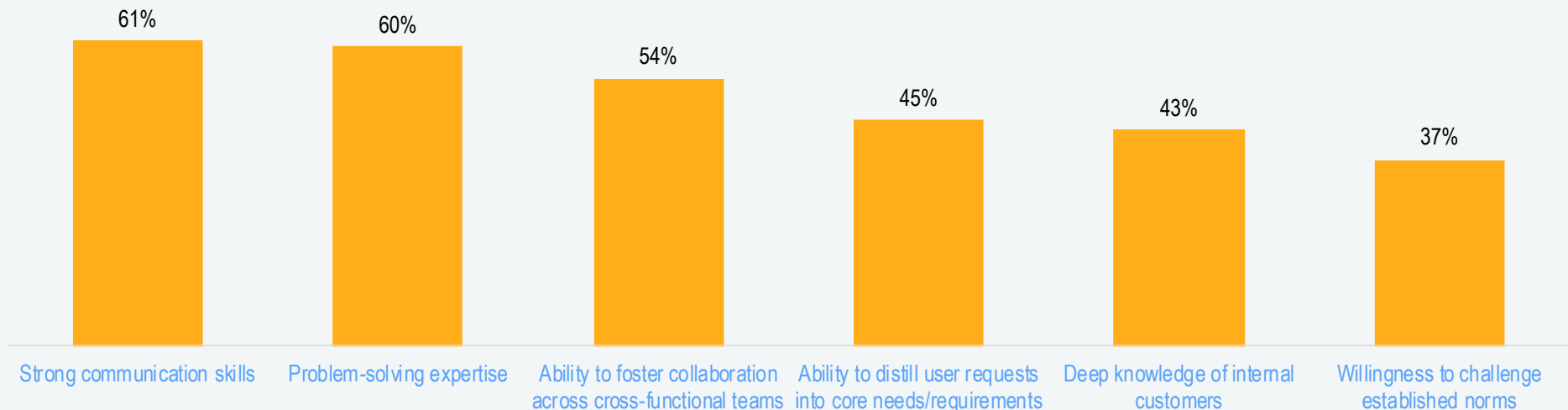
### What are your priorities for the platform team over the next 12 months?



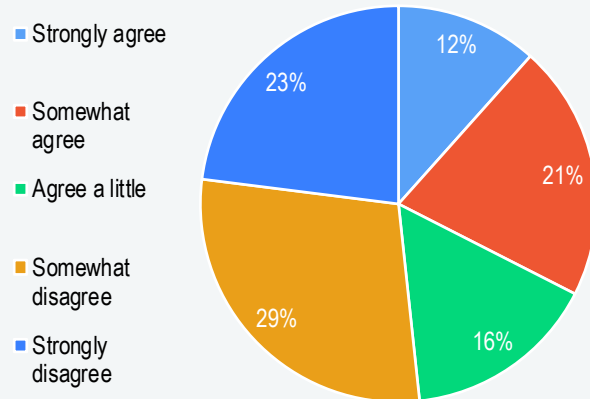
In more good news, respondents demonstrate a good understanding of the most important product management skills, with strong communication skills (61%) and problem solving expertise (60%) high on the list, followed by the ability to foster collaboration across cross-functional teams (54%), the ability to distill user requests into core requirements (45%), deep knowledge of internal customers (43%), and willingness to challenge established norms (37%).



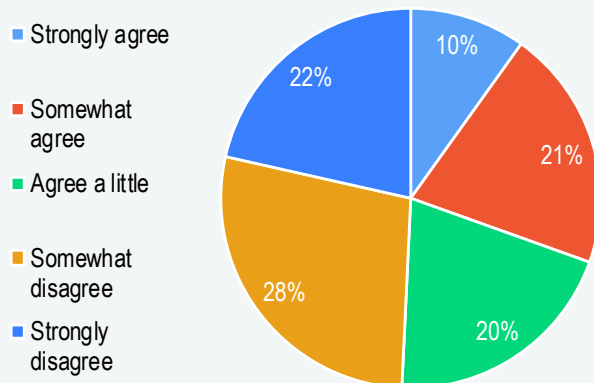
### What are the most important product management skills on a successful platform team?



### Senior management at my organization does not understand the value that platform engineering can provide.



### I (the respondent) often find the concept of platform engineering confusing.

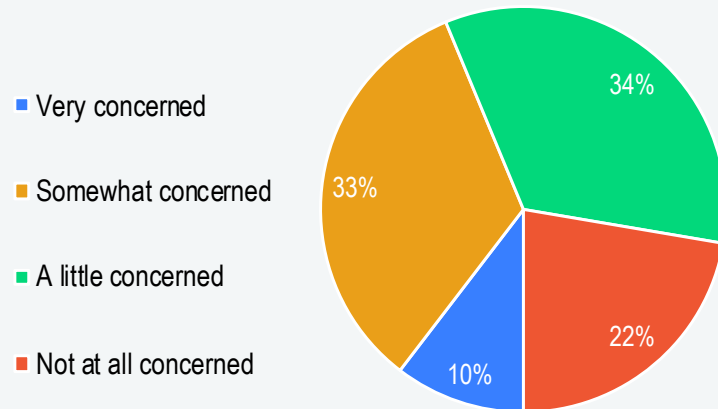


Now for the less good news: Despite sharing priorities that clearly indicate the need for product management and possessing a good understanding that product management brings necessary skills to the table that enable delivering on those priorities – not to mention the fact that 66% of respondents strongly or somewhat agree that their platform engineering team has a product owner who evangelizes the platform services and capabilities – there are still clearly issues that should be getting solved via product management.

Nearly half (48%) of respondents agree that senior management at their organization does not understand the value of platform engineering, and a similar degree of respondents themselves (51%) admit that they are sometimes equally confused about the concept. Again, this points towards an underinvestment in the product manager role, which requires good skills in effectively disseminating information up, down, and across the organization.



**Are you concerned that the platform team will not evolve as quickly as the needs of product teams using the platform?**



Additionally, 78% of respondents have at least some concerns about whether their platform team can keep pace with the evolution of the product teams using the platform. This highlights that respondents are acutely aware that evolution is a key facet of platform engineering, and that continuing to ensure your product meets user needs is also a key part of the product management necessary to platform teams.

We didn't dig into the background and skill set of folks operating as product managers of the platform this year, but in our experience, many of the people operating as product owners in enterprise environments have not come from a product management background, and are not enabled by their organizations to develop those skills. As with all great product managers, a mix of domain expertise and product management skills is required, and we've identified this area as one to analyze more deeply in possible future reports.

As we will find in the next section, despite all of these clear signals that investment in product management is needed, organizations are failing to make those investments in terms of hiring.

## Organizational Design

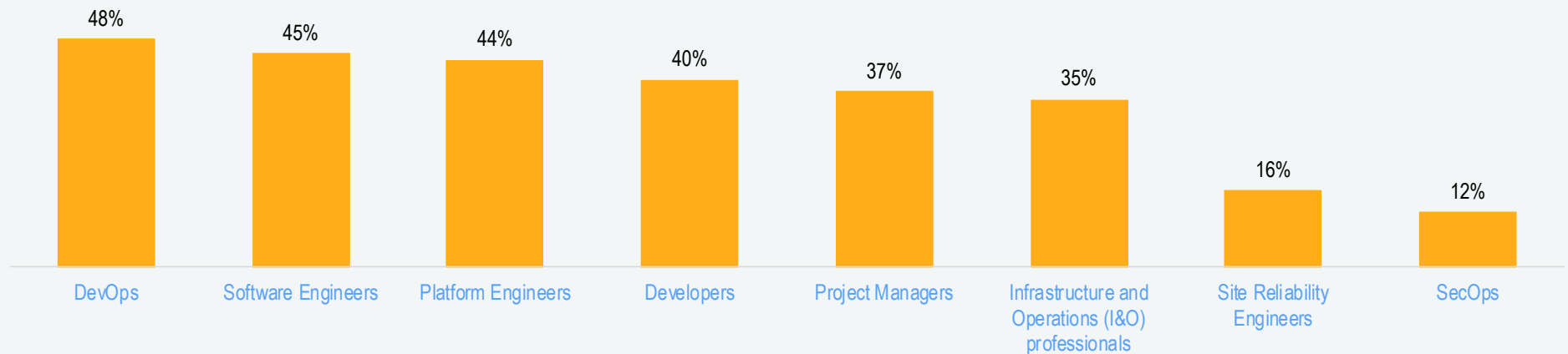
Platform teams must be staffed and run by people with the right mix of skills and understanding of the business in which they operate. Most respondents (53%) report their organization plans to hire people with platform engineering experience over the next six months. This urgency is likely driven not only by the growth of platform engineering at organizations, but also by the diverse range of skills needed on platform teams. Most commonly, firms are employing people with DevOps experience (48%) on platform teams, again illustrating the close relationship between DevOps and platform engineering (and yet we find ourselves wondering whether many of these folks know what they mean by “DevOps skills,” and we’re quite sure they don’t all mean the same thing).

An array of other roles is also commonplace, including software engineers (45%), platform engineers (44%), developers (40%), project managers (37%), I&O professionals (35%), site reliability engineers (SREs) (16%), and SecOps (12%).

Here we should note that product managers are missing from this list and that project managers are strongly represented. Project management is necessary for large enterprises in order to keep complex projects running smoothly, but it should absolutely not be confused with product management. The data shows that respondents recognize that they need the kind of skills that fall within the domain of product management, yet they are not hiring for these roles.

In our experience, one of the most common failure patterns for large organizations implementing platforms is to implement them via project management, rather than product management.

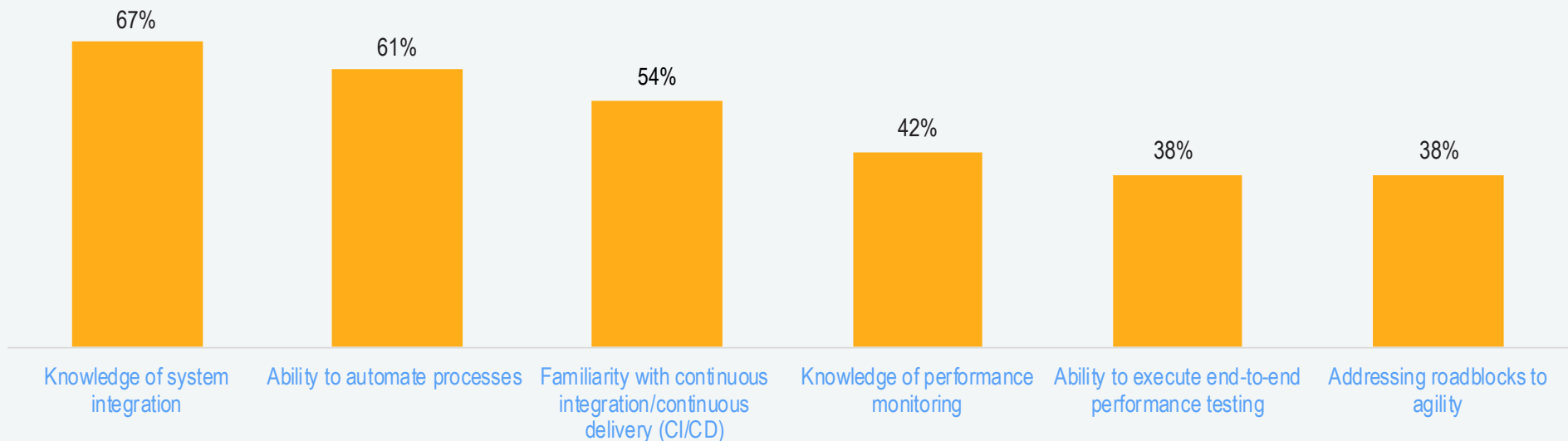
### Which roles comprise the platform team at your organization?



When it comes to specific technical skills, most firms are looking for knowledge of system integration (67%) and the ability to automate processes (61%), followed by familiarity with CI/CD (54%), knowledge of performance monitoring (42%), ability to execute end-to-end performance testing (38%), and addressing roadblocks to agility (38%). We're thrilled to see folks focusing on more than just infrastructure-as-code in order to deliver a complete system.



### What are the most important technical skills on a successful platform team?







## What Do We Want? Centralization! When Do We Want It? Now!

Most firms have multiple development teams who are generally responsible for their own work, but only one centralized IT team, who has responsibility to much more of the organization, including development teams. We've seen a trend – partly due to DevOps, self-service, cloud – in which dev teams have taken on more responsibility in managing infrastructure themselves because their needs were not being served by central IT, and because the use of public cloud platforms allowed them to bypass IT to get their jobs done. Sometimes these development teams build solutions for themselves that get shared and adopted within another team, and a platform begins to emerge organically.

In such situations, we wanted to investigate whether there was a desire for development teams to continue to maintain the platform, or whether they were happy for it to move towards a more centralized team with a mandate for supporting more of the organization.

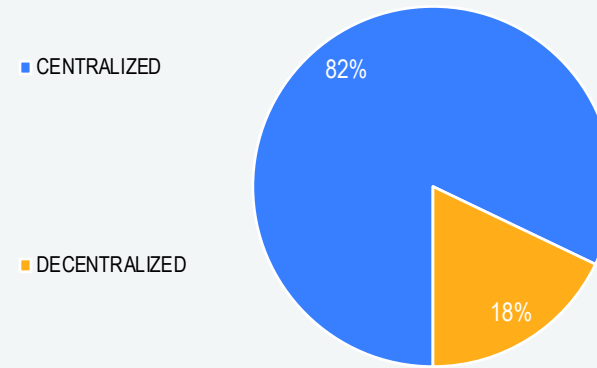
### What does centralized vs decentralized mean in this report?

- **Centralized:** The platform team supports various business units across the organization.
- **Decentralized:** The platform team is embedded in an individual business unit and only serves that team.

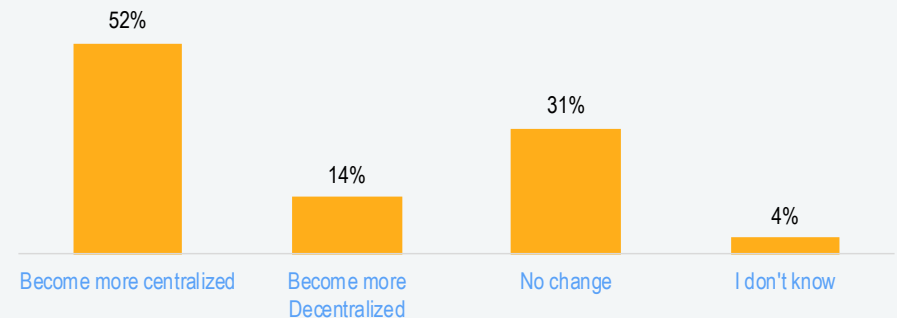
The vast majority of respondents (82%) describe their platform team as supporting multiple business units across the organization, an unsurprising finding given that the whole point of running a platform is to provide efficiency at scale. We also see that the majority of respondents (52%) report that the trend towards centralization has increased over time.

These interesting results are worth future research to help us distinguish between a central IT team starting small and iterating towards more scale, and a platform emerging organically from development teams before becoming owned by a team with a more explicit mandate of broader organizational support. With current data, we are not able to identify whether the reason behind the 14% of respondents who describe a trend towards more decentralization is due to a central platform that isn't meeting developer needs.

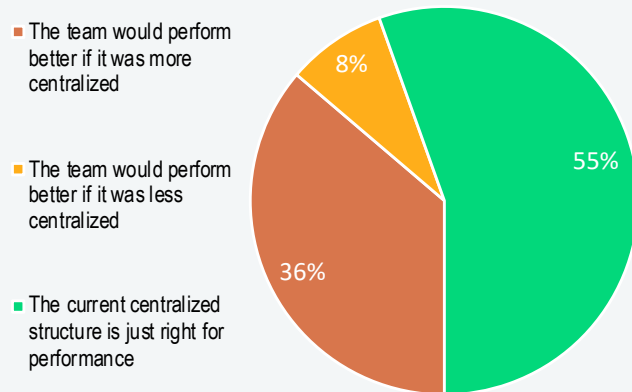
### Is the platform team(s) at your firm centralized or decentralized?



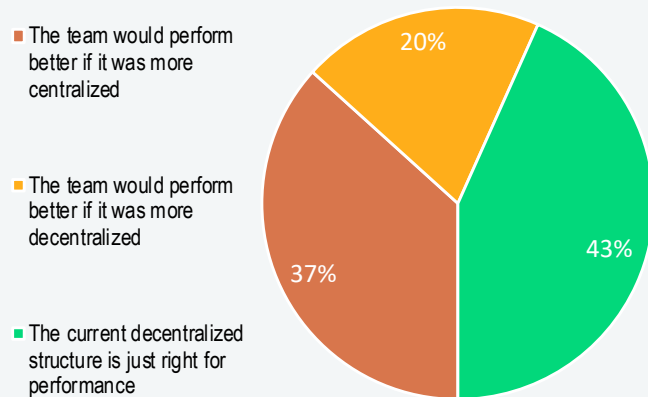
### To what extent has the platform team structure changed over time?



### Preference among centralized organizations



### Preference among decentralized organizations



Regardless, there's still a significant appetite for more centralization: 36% of those with a currently centralized approach believe the team would perform better if it was even more centralized. Among those with a decentralized approach, more than a third (37%) claim that performance would improve if a more centralized approach was taken. Notably, both users and operators of the platform teams are on the same page here, as there are similar shares of both who want more centralization. This is not just the operators of a platform seeking greater control and influence organizationally.

## Platform Teams at Global Scale

Platform engineering is already a global trend, but some regions are having a better experience than others. Respondents from Asia Pacific are the most upbeat about platform engineering, with 86% reporting that their platform team is at or close to optimal performance – compared to 64% in EMEA and 73% in the Americas. More respondents in Asia Pacific believe that platform engineering is helping their organizations realize the benefits of DevOps, with 44% in “strong” agreement compared with smaller shares of 34% in EMEA and 32% in the Americas.

The driver of this sentiment is likely the improvement in development velocity that Asia Pacific respondents are experiencing: most (53%) report speed is improving “a great deal,” while 41% in the Americas and only 29% in EMEA report the same.

Two features of Asia Pacific platform engineering adoption stand out that could account for these positive results. The first is a greater move toward centralization over time: larger shares of Asia Pacific (61%) respondents are experiencing this trend compared to those in the Americas (55%) and EMEA (40%).

Secondly, Asia Pacific organizations are more likely to have a product owner role. Nearly three quarters (74%) in the region strongly or somewhat agree that their product engineering team has someone in this role who evangelizes the platform services and capabilities, while the Americas (63%) and EMEA (62%) lag. As noted earlier, the product owner can be the gatekeeper to unlock communication roadblocks, boost problem solving capabilities, and address the need for more platform engineering education across the organization.

We expect that platform engineering will continue to increase globally just as DevOps did, and as it matures across organizations, greater impacts and new best practices will be established.



## In Conclusion

Platform teams take time to establish, but once they are running effectively, we find that they drive true impact across an organization in the form of speed, scalability, and consistency, and do so relatively quickly. However, we simply aren't finding enough of a focus on platform-as-product and a recognition that you need people filling the role of a product manager, no matter what actual job title they have.

Just as most enterprise DevOps initiatives failed to achieve their potential due to a lack of focus on organizational dynamics and that imprecise cluster of things we often call "culture," so too will most platform approaches fail if technology remains the primary focus.

Software development may be a team sport, but enterprise IT is an even more complex social activity that builds socio-technical systems. The companies that understand this and explicitly focus on team interactions, developer experience, feedback loops, and product management will leapfrog those who do not.

**Treat your platform as a product, not as a project.**

# Methodology

## Target population and sampling method

This year's survey collected data from IT practitioners and leaders who work with or are part of their company's platform engineering teams. We also included a control group of similar participants who do not have platform teams at their organizations for comparison. These two participant groups allowed us to understand the direct impacts of platform engineering teams on important DevOps metrics. The survey was conducted online from September 14th to October 25th, 2022, and respondents were gathered through two avenues, a snowball sampling method and a professional panel.

## Snowball sampling

Snowball sampling is a process in which respondents are encouraged to share a survey with their networks, causing the sample to grow (like a snowball rolling downhill). Promotion was done via email lists, social media, and various partners, and the sample was collected globally, from Europe, the Middle East, Africa, the Asia-Pacific region and the Americas. Given the channels of promotion and the nature of snowball sampling, this portion of the sample is likely limited to firms and teams that were already familiar with DevOps, and as such, who may be doing some DevOps.

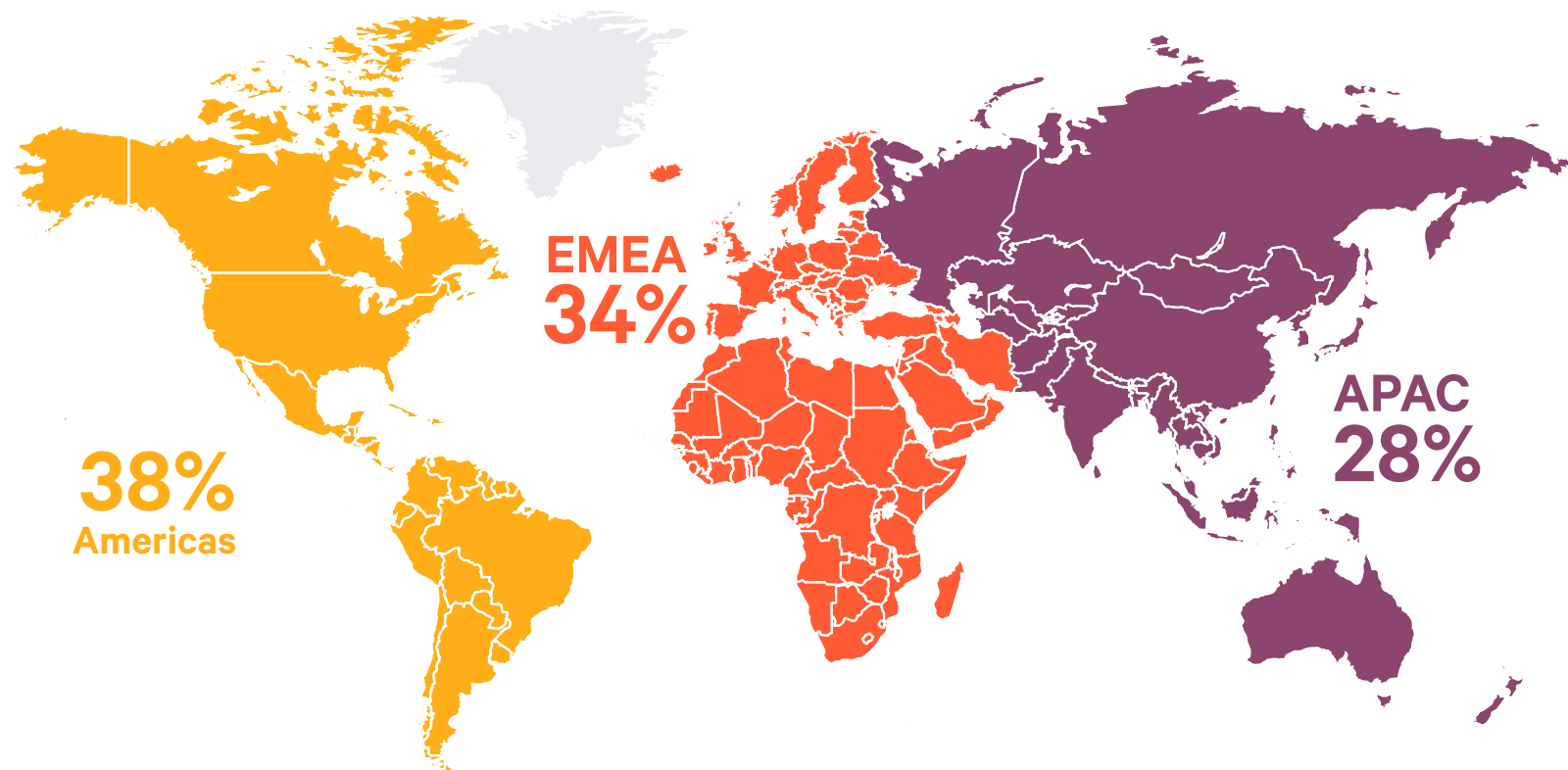
## Panel sample

The snowball sample was supplemented by a third-party panel, which reduces bias in our sample. Our third-party panel provider nurtures and maintains a quality, engaged membership panel built to support its market research clients and to benefit non-profit organizations. This panel provider's unique approach to recruiting yields a highly engaged group of people who, as respondents, are dedicated to helping market research clients fulfill their information needs. The panel provider's unique non-profit recruitment method enables the firm to source C-suite executives, directors, and managers who have key decision-making authority. In addition to their non-profit relationships, this sample provider also utilizes trade association partners to help drive certain audiences into online surveys. This approach provides access to the appropriate sample for each survey.

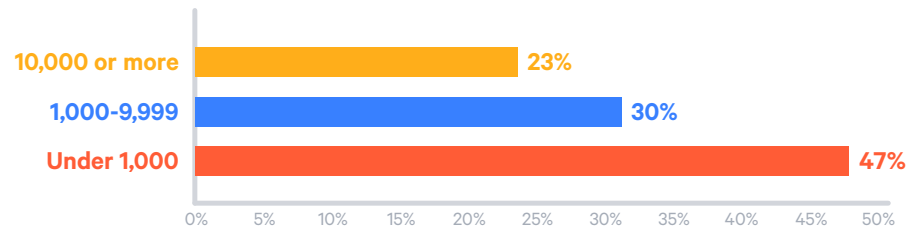
# Who Took The Survey?

As we have for the past decade, we sought survey respondents from as wide a range of geographic regions, industries, and company sizes as possible. We want to thank all 438 individuals who responded to this year's smaller-scale survey.

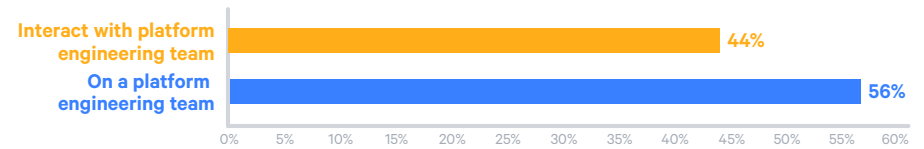
## Responses by global region



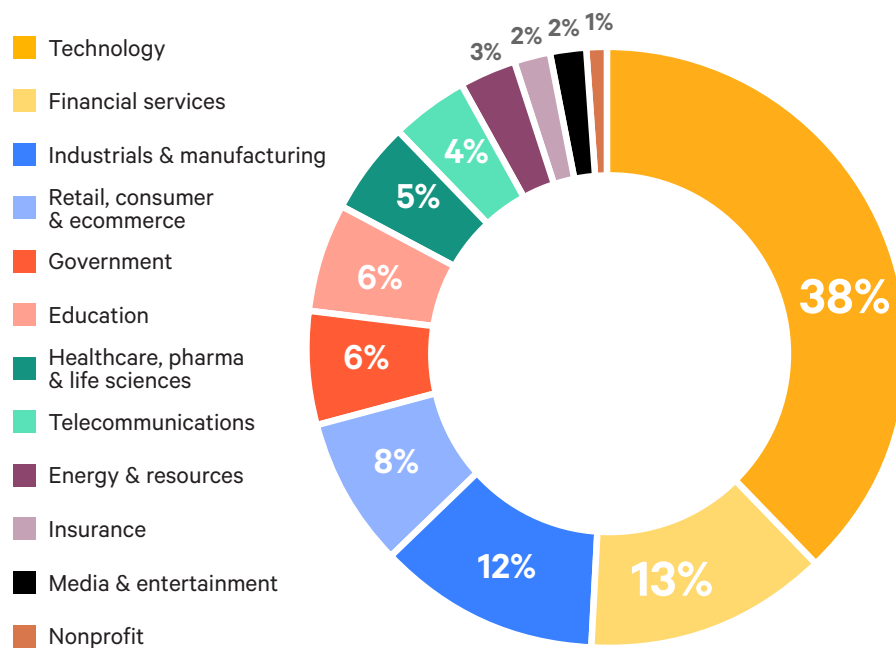
## Number of employees



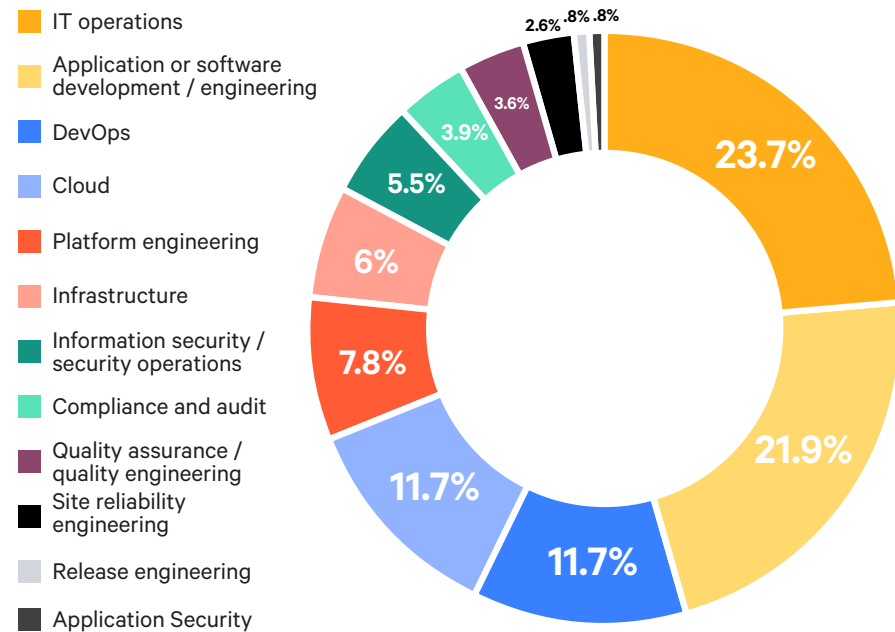
## Platform team role



## Principal industry



## Team





# Author Biographies



**Ronan Keenan** Ronan Keenan is Director of Research at Puppet by Perforce, leading a variety of market and product research initiatives. Previously, Ronan worked as director at research consultancy firms where he was a strategic partner to Fortune 500 brands. Ronan's research has appeared in the Wall Street Journal, USA Today, The Atlantic, and World Policy Journal, among other publications. This is his first State of DevOps Report.



**Nigel Kersten** is CTO at Puppet by Perforce. He works with many of Puppet's largest customers on the cultural and organizational changes necessary for large-scale DevOps implementations. He has been deeply involved in Puppet's DevOps initiatives, including the State of DevOps Reports, and has served in a range of executive roles at Puppet over the last 10+ years.



**Caitlyn O'Connell** Caitlyn O'Connell is head of brand and corporate content at Puppet by Perforce. This is her second State of DevOps Report. Previously, Caitlyn spent five years at the Linux Foundation where she ran marketing and communications for the open source project Cloud Foundry and spearheaded DEI programming for global events. She is passionate about ethical, collaborative tech, from open source to DevOps.

## Report Sponsors



### Continuous Delivery Foundation

Continuous Delivery Foundation is an open source community improving the world's ability to deliver software with security and speed. We help you figure out your best DevOps path to being a high performing team and how to use open source to get there.



### Humanitec

Platform engineering is revolutionizing how enterprises build and run their cloud-native setups. Humanitec is at the core of this revolution, enabling teams to build Internal Developer Platforms (IDPs) and reach true developer self-service. Humanitec's Platform Orchestrator is the engine at the heart of a dynamic IDP. It lets platform teams, from growing startups to enterprises, remove bottlenecks by letting them build golden paths for developers.



### Women in DevOps

Women in DevOps is a global community established to address the gender imbalance within the DevOps industry. Our aim is to help close the DevOps gender gap and empower the DevOps leaders of the future. Our unique platform has become a global movement and is used to not only amplify the voices of women, but of all minority groups within DevOps, to break down the barriers and drive positive change.



### **About Puppet by Perforce**

Puppet by Perforce empowers people to innovate through infrastructure automation. For more than a dozen years, Puppet has led the way in IT infrastructure automation to simplify complexity for the masses in order to strengthen customers' security posture, compliance standards, and business resiliency beyond the data center to the cloud. More than 40,000 organizations — including more than 80 percent of the Global 5000 — have benefited from Puppet's open source and commercial solutions. In 2022, Puppet was acquired by Perforce Software.

Learn more at [puppet.com](https://puppet.com).



2023

# State of DevOps Report

Platform Engineering Edition