

# The state of feature management

20  
22

LaunchDarkly →



# A look at IT spend, adoption trends, and overall maturity.

**Feature management is poised for a strong year. Despite turbulent financial times, this report demonstrates why companies are investing in feature management, and what they're getting in return.**

We'll also show how product delivery teams are leveraging feature management to smooth releases, avoid disasters, and improve the overall quality of software.



# Feature flags vs. Feature management

Before getting started, let's clarify the term "feature management," since it's still a relatively new concept. Feature management is powered by feature flags, so we'll begin with that definition.



## Feature flags

Conditional statements that change how code behaves in production after it has been deployed. Feature flags can be as straightforward as an on/off switch. However, a more complicated and valuable implementation that involves targeting, multivariate flags, and consideration of multiple teams, environments, and applications motivates organizations to pursue feature management.



## Feature management

A class of software development tools and techniques that provides a framework for feature flag-driven development, infrastructure, and experimentation. Feature management enables teams to use flags on a massive scale across a variety of use cases. It is foundational to modern development practices such as DevOps, CI/CD, trunk-based development, and progressive delivery. Teams that employ feature management increase developer productivity, maintain high system performance, and continuously improve the quality of their software.





# Survey methods

The independent research firm Wakefield Research conducted the survey for this year's report. They queried:

# 1000

**software & IT professionals**—more than double the number from last year's survey— from companies with a minimum of 200 employees in the United States.

The survey was conducted between Sept. 21-Oct. 3, 2022. "Software and IT professionals" include software engineers, developers, release managers, product managers, platform managers, and software architects. The survey also included a requirement that respondents currently use, or plan to use, feature flags.

While we released a [State of Feature Management report in 2021](#), in that report, we separated respondents who were LaunchDarkly customers from non-customers. Because that sample was mixed and we did not group respondents into an aggregate, we decided to not compare this year's dataset to last year's, except in very specific cases.



# Executive summary

A large number of company leaders see investing in feature management as a high priority. In fact, all survey respondents say that their company's IT budget for feature management needs to increase in the upcoming year:

# 74%

of respondents say their **budget for feature management needs to increase by over \$100K**, including 80% of senior executives

# 23%

say by **\$250-499K** ▲

# 7%

say by **over \$1M** ▲

Clear majorities of those who use feature management say they've seen improved processes as a result, especially around software delivery speed (84% of respondents), application reliability (87%), and developer productivity (87%).

In fact, virtually all of the respondents who use feature flags (98%) agree that toggles **save their company money and deliver a demonstrable ROI**.

We're also seeing a lot of newcomers to feature management: 60% of respondents started using feature flags within the last year, perhaps suggesting a rapid rise in adoption.

In general, we found that organizations vary widely in their initial reasons for adopting feature management. But in practice, they gravitate toward similar use cases. The two most common use cases are long-term operational controls (55% of respondents) and entitlements (47%). Interestingly, both use cases rely on long-term/permanent feature flags. This suggests that flags are playing a foundational, enduring role in enabling organizations to both deliver and maintain the reliability of their software.



The majority of respondents (81%) have standardized feature management on a single team, multiple teams, and/or across the entire organization.

Through this lens of feature management adoption, we've begun to identify an emerging maturity model for feature management. In particular, we observed a pattern associated with maturity and software delivery performance and reliability.

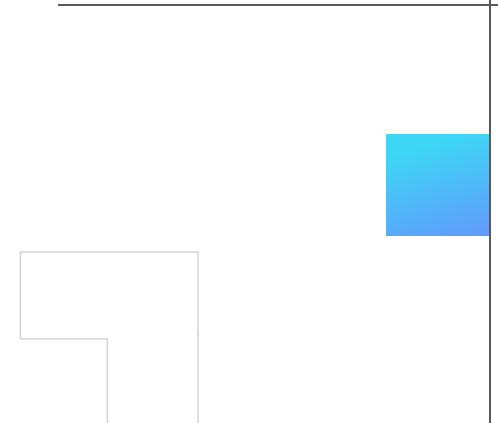
As organizations move from using flags in an ad hoc manner (less mature) to standardizing feature management on a single team (more mature), their performance dips. But as they keep maturing, their performance improves. Though organizations may face growing pains when first scaling the practice of feature management, it appears that, if they persist, they reap the rewards of their investment down the road.

Perhaps surprisingly, of all the roles within an organization who interact with feature flags, **IT operators (43%)\* are the most prevalent—even more than software developers (40%).**

*\* 43% of respondents say that "IT operators" at their organization interact with feature flags. Whereas, only 40% of respondents say the same of "software developers".*

The high percentage of IT operators reinforces the idea that feature management is playing a critical infrastructure role for organizations.

Read on for more about how feature management is evolving within the software industry and the paths many organizations are taking to keep pace.



# A business-critical investment

Leaders increasingly view feature management as mission-critical to their business and are adopting the practice accordingly.



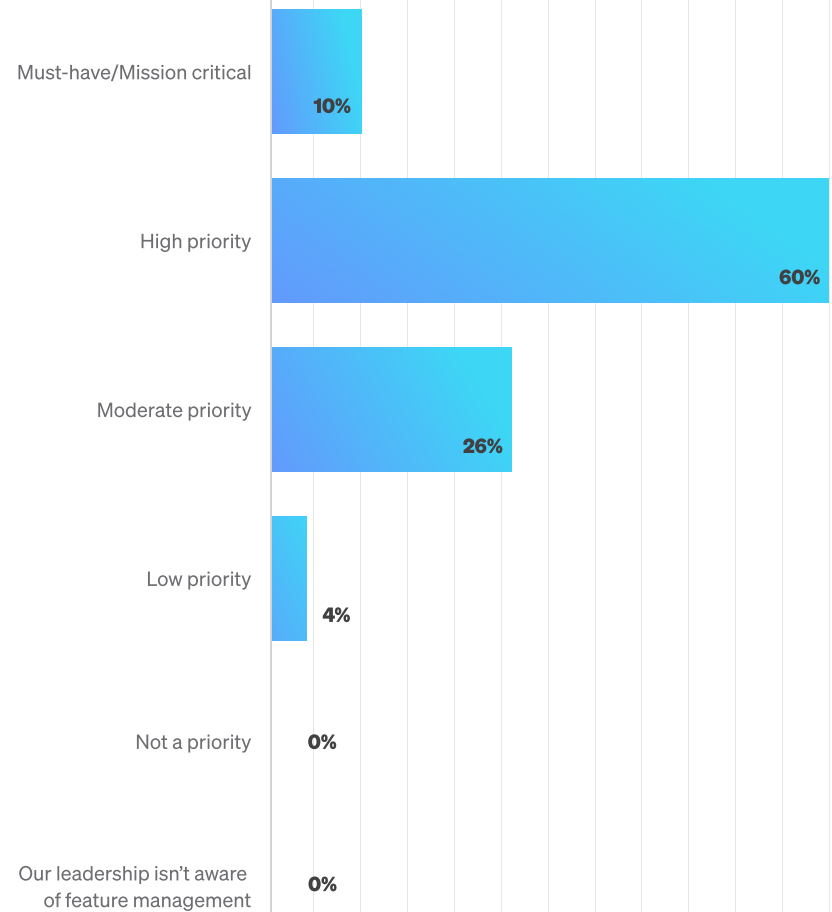
# How do leaders at your company view feature management as an investment?

Nearly

# 70%

of respondents view it as either a high priority or mission critical.

Among organizations that deploy several times a week or more, 77% say their leaders view feature management as a top priority.



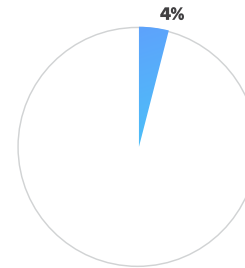


Over the next 12 months,

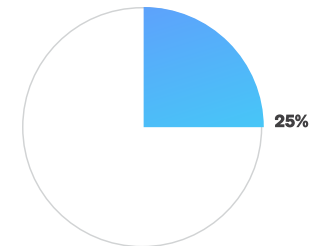
# How much do you expect your company's IT department to increase or decrease its budget for feature management?

Nearly 80% of respondents say their budget will grow for feature management within the next year, with only a tiny percentage (4%) saying they think it'll decrease.

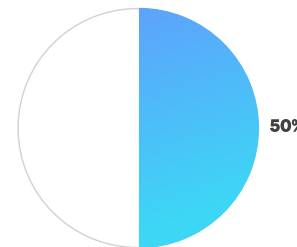
Given the widespread cost-cutting we've seen in the technology sector in recent months, these projected spending answers tell us that feature management is becoming a core component of software development.



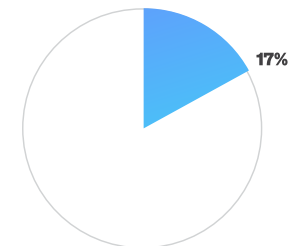
▲ More than 20% increase



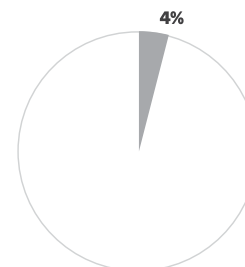
▲ 11-20% increase



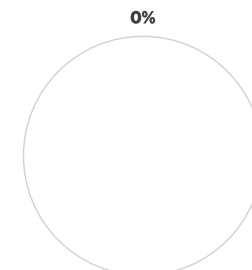
▲ 1-10% increase



Stay about the same



▼ 1-10% decrease

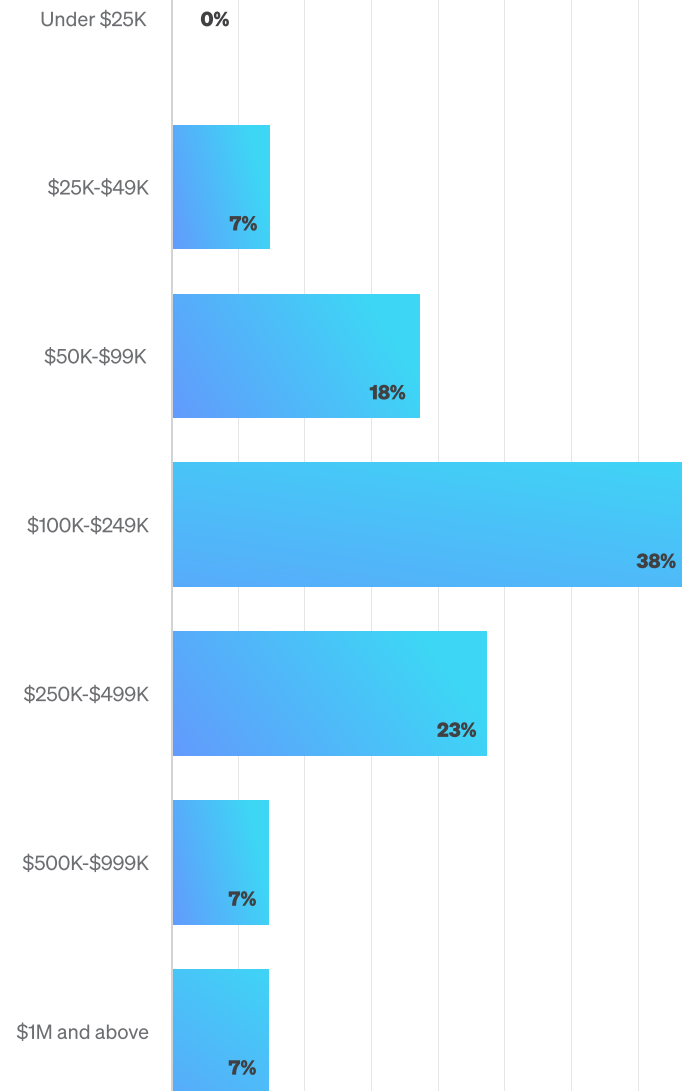


▼ 11-20% decrease



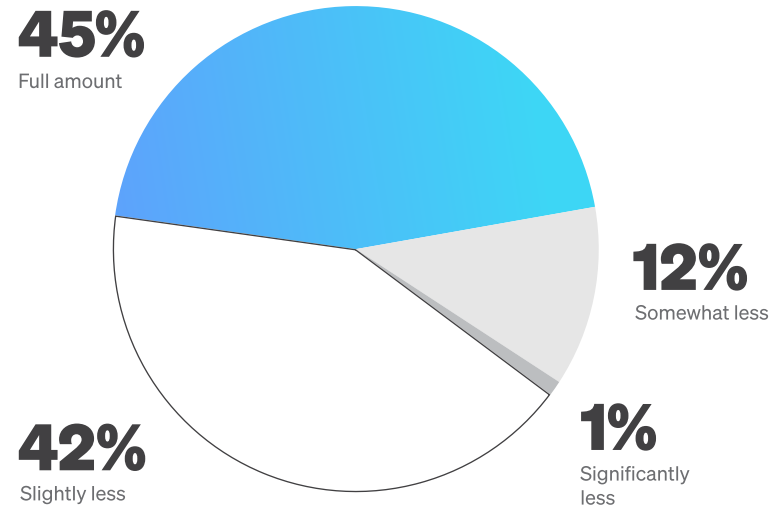
# How much do you think your company's IT department budget on feature management needs to be for the upcoming year?

The overwhelming majority of our respondents hope to **invest between \$100K and \$1M in feature management in 2023**. That includes 80% of senior executives, who say the minimum investment needs to be over \$100K.



# How much of the budget needed for feature management do you expect to get?

// asked among those who think their company's IT department will budget for feature management



Wants and needs when it comes to development tools are not the same, but as we've seen, feature management is a big priority for organizations. Nearly 90% of respondents say they expect to get exactly what they need or at least somewhere in the range.

That said, budget predictions look different depending on how close IT professionals are to the C-suite: while 61% of senior managers and 61% of mid-managers or non-managers worry

they will not get the feature management budget they need, only 45% of senior executives say the same. This discrepancy in expectations might point to an important issue:

**43%** say that they **need more management buy-in** to advance their feature management goals.



# Feature management motivations

**It's clear companies feel feature management is worth investing in, but why?**

We've got [our own ideas](#), but in this section, we'll dive into what our respondents say are their top reasons for using feature flags.

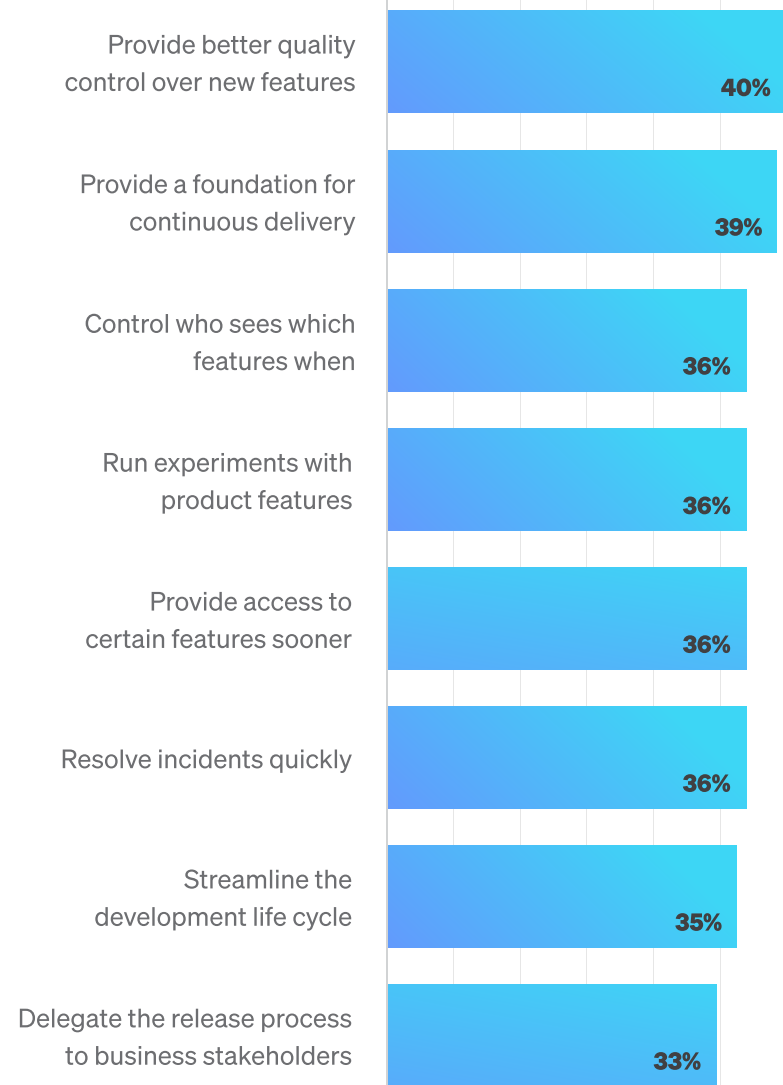


# Which of the following are currently your top reasons you use feature flags in your software development practice?

// asked among those who use feature flags

Teams share a lot of the same motivations for using flags. The top reason, by a slim margin, is **"providing better quality control" (40%)**.

Using feature flags to roll out new features and then immediately turn them off if there's an issue minimizes risk and maximizes reliability. Teams don't need to perform a massive rollback and dig through code for issues following a botched deployment. All they need to do is flip a feature flag off.



The next most popular reason is feature flags' ability to **provide a foundation for continuous delivery (39%)**.

This likely ties in to many organizations' move away from big bang or even quarterly releases to ones that are more frequent. Since feature flags reduce the number of things that can go wrong with releases, they give teams more confidence when accelerating their software delivery timelines. This shift coincides with teams making feature management an integral part of their software delivery pipeline, often added to their larger, internal self-service platform.

We've already talked a bit about the power of feature flags to control who sees which features and when, and we'll be touching on incident resolution in a bit. So let's first talk about experimentation with feature flags, which looks like it's on the rise.

The third most-cited reason for using feature flags is a **four-way tie at 36% each**:



Run experiments with product features



Control who sees which features when



Provide access to features sooner



Resolve incidents quickly



# Experimentation and feature management

The popularity of experimentation as a use case for feature management demonstrates the importance of understanding user needs and the business impact of software changes.

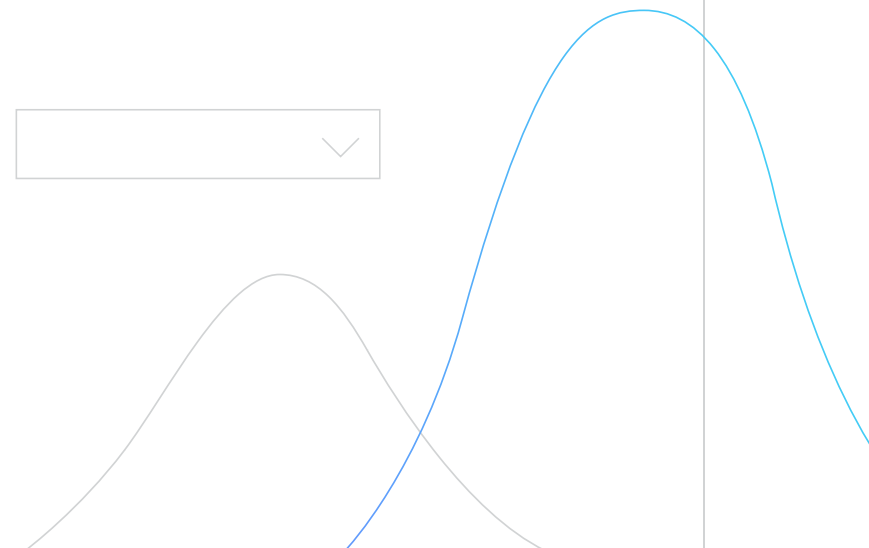
While feature management is helping organizations accelerate their development cycles, fast delivery alone is not enough to satisfy stakeholders. *What* you deliver is just as critical as *when*. That's where experimentation comes in.

Experimentation is nothing new to the industry. Elite organizations have been running hundreds or thousands of experiments a year, finding optimizations and new opportunities that accelerate their business growth and provide an edge over their competition. But this success comes with a price tag—these same organizations also hire massive teams of data scientists and engineers dedicated to building and maintaining homegrown experimentation tools.

For the vast majority of organizations, the full benefits of experimentation remain out of reach. The good news is feature flags and their [associated platforms](#) are gaining visibility as an approach to ensure that fast delivery is ultimately providing value to the business, while helping to ease some of the complexity from experimentation.

With feature management at the core, experimenting becomes easy and powerful, and brings siloed teams together to run more high-quality experiments and get faster results.

We believe that using feature flags will help increase the number of organizations who are experimenting with their software releases. And that's a win for both companies and end-users.



# Many newcomers off to a fast start

**While feature flags have a long history in programming, many people have just begun using their powers.**

The majority of our respondents (60%) began using feature flags within the past year. And these feature flag freshmen are outperforming some veterans in key areas.



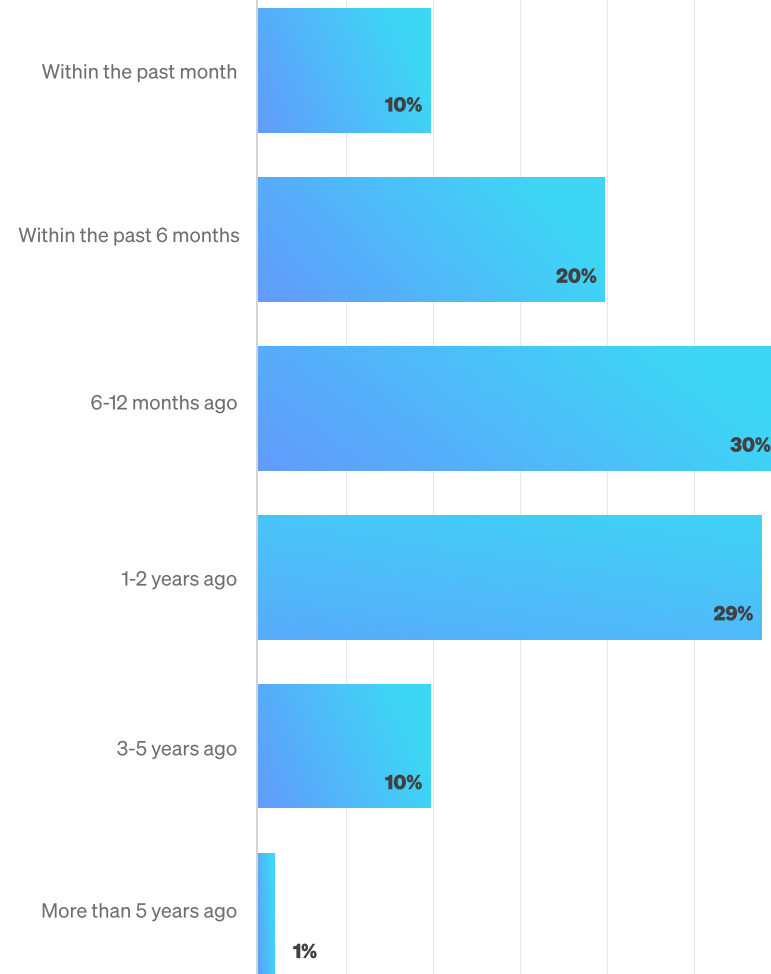


# When did you start using feature flags in your software development practices?

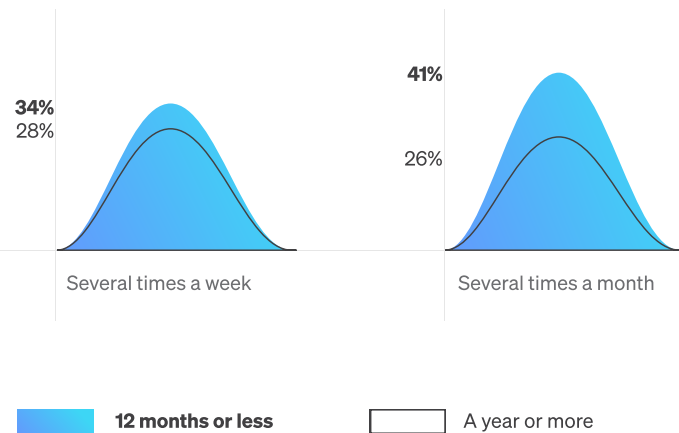
// asked among those who use feature flags

While it's interesting to us that 60% of our respondents have started their feature flagging journey within the past year, it's fascinating that 20% have done so in the past six months and even 10% within the last month.

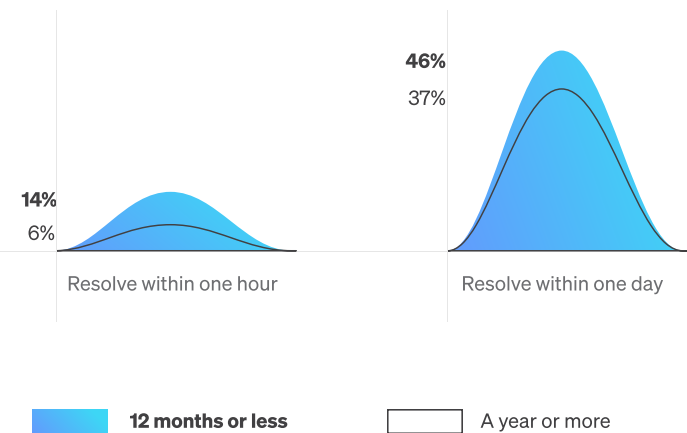
Given the prioritization feature management is getting from organizations, we expect these numbers to keep ticking up.



### Deployment frequency for those using feature flags:



### Mean time to resolution (MTTR) for those using feature flags:



Interestingly, when it comes to deployment frequency and mean time to resolution (MTTR), these **newcomers are outperforming those who have used flags for a year or more.**

One hypothesis for this difference in performance is that the earliest feature flag adopters have paved the way for later adopters. Newcomers can use and replay established practices and feel confident in making greater initial investments in feature management, which yield better results in less time.



# Feature management maturity

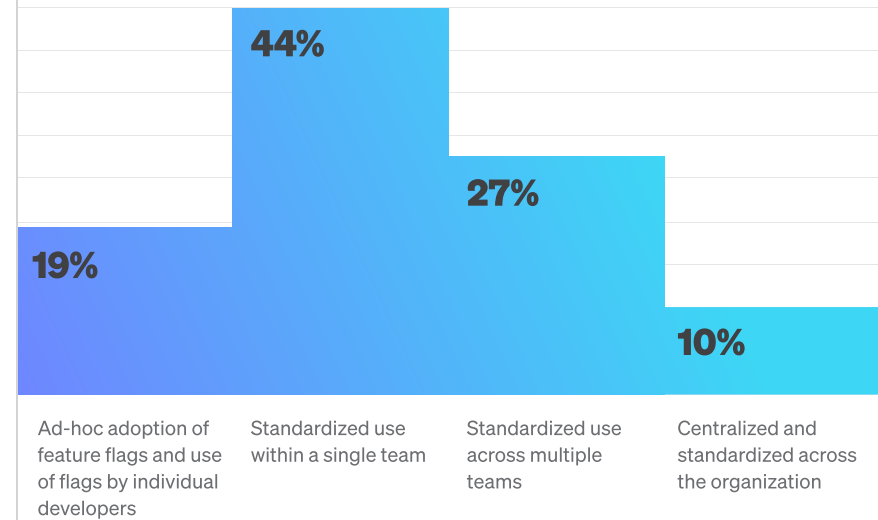
**In the survey, we organized respondents by their level of feature management adoption.**

At the low end of adoption, we see individual developers using feature flags for one-off practices, which we describe as "ad hoc." At the high end of adoption, we see flags implemented as a centralized standard practice across the entire organization.



# Which of the following best describes your team's practice of feature management?

// asked among those with a system to manage features



While we recognize that feature management adoption is not synonymous with maturity, the lens of adoption is a helpful starting point for defining feature management maturity. Indeed, some of the trends we observed help us better understand what maturity looks like in practice.

For example, **52%** of respondents in the "centralized/standardized" group (the most mature in terms of adoption) reported that their organizations only started using feature flags within the last 12 months.

This suggests that, in many cases, feature flag adoption can spread quickly, especially if there is centralized support.

One reason why the "centralized/standardized" group was able to expand their feature flag usage so quickly may be that the organizations are relatively small. 68% of respondents in the "centralized/standardized" group come from organizations with fewer than 1,000 employees. So, the smaller an organization, the easier it is to spread feature flag adoption across its entirety.



# A maturity model for feature management is emerging

For our report, we focused first on standardization of adoption because we see that as a critical foundation for maturity. While there's no clear, industry-accepted maturity model for feature management yet, our experience working with our customers has revealed an approximate version based on the following dimensions:

## Use cases

// How your company uses feature management, including the variety and complexity of those use cases

## Depth of deployment

// The degree to which feature flags permeate your deployment process (such as flag coverage), ranging from using flags for cosmetic tweaks intermittently to wrapping nearly all production changes in a flag

## Automation

// Progressing from manually managing the lifecycle of individual flags to automating entire release workflows and multivariate experiments

## Process (or adoption level)

// The degree to which your organization has adopted feature management, ranging from a few developers using flags casually to multiple stakeholders and teams relying on feature management as a business-critical system

## Integrations

// Progressing from zero integrations to leveraging all the relevant integrations between your feature management platform and workplace messaging apps, APM and observability solutions, integrated development environments, Infrastructure as Code tools, and more



As teams advance through each of these five areas, they begin to grow their feature management capabilities. For instance, if a team starts with a use case of dark launching for feature flags, it might then progress across several stages where it would pick up canary launches, then migrations, entitlements, and finally graduate to advanced methods like personalization.

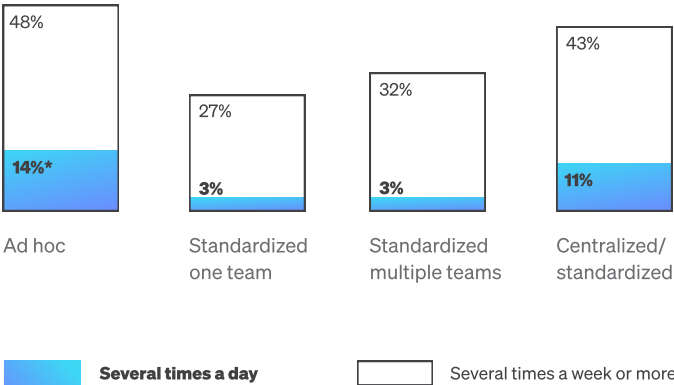
As we examine responses from this year’s survey, we notice that some teams are adept at feature flagging initially, but as they standardize the practice to reap the biggest rewards, they face a learning curve. This is possibly due to needing to explore new use cases, onboard users and teams to feature management practices, and change existing release and infrastructure expectations. Once teams have mastered some of those new practices, performance rises again.

As you scale adoption across teams, complex new challenges often arise. But the data suggest that when organizations get these growing pains under control, they reap increasing business and software performance benefits over time.

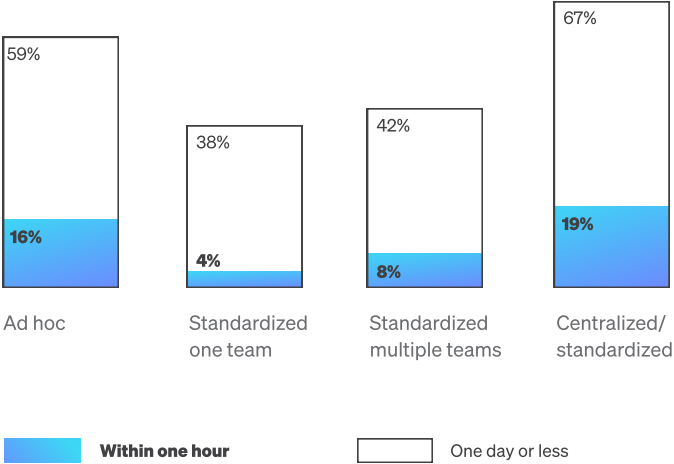
The following tables\* show the association between software delivery performance metrics and feature management adoption levels.

\* How to read the tables: In the first table in the top right, 14% of respondents in the “Ad hoc” group say they deploy new code several times a day. Only 3% of respondents in the “standardized on one team” group say they deploy new code several times a day. And so on.

### Deploying new code

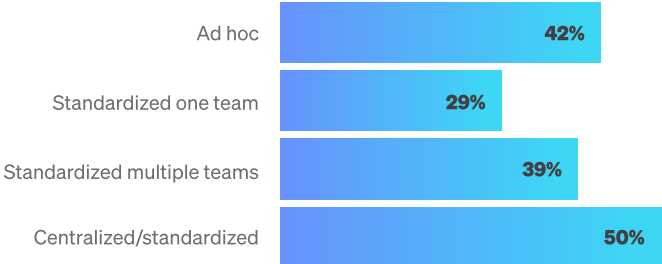


### Current MTTR

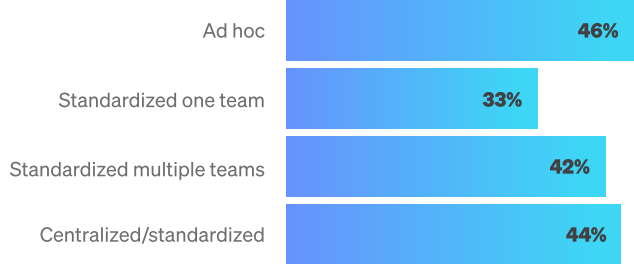


# Those who say feature management...

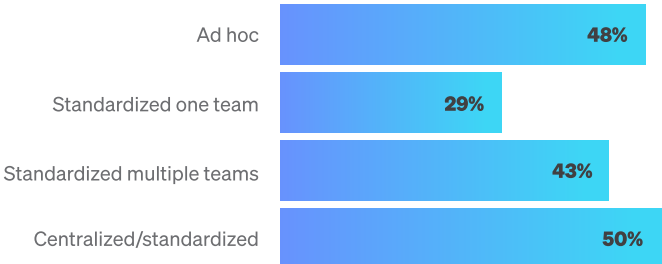
has significantly **improved the speed** at which they deliver software.



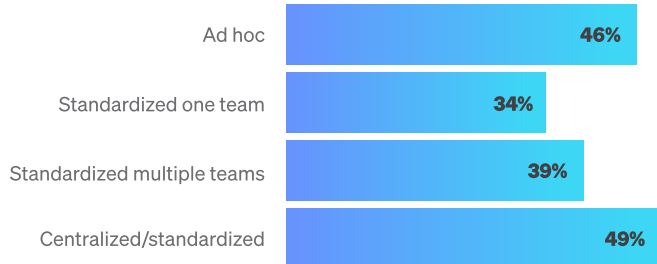
significantly **enables faster feedback loops** from customers regarding new features.



has significantly **improved application reliability.**



has significantly **improved developer productivity.**

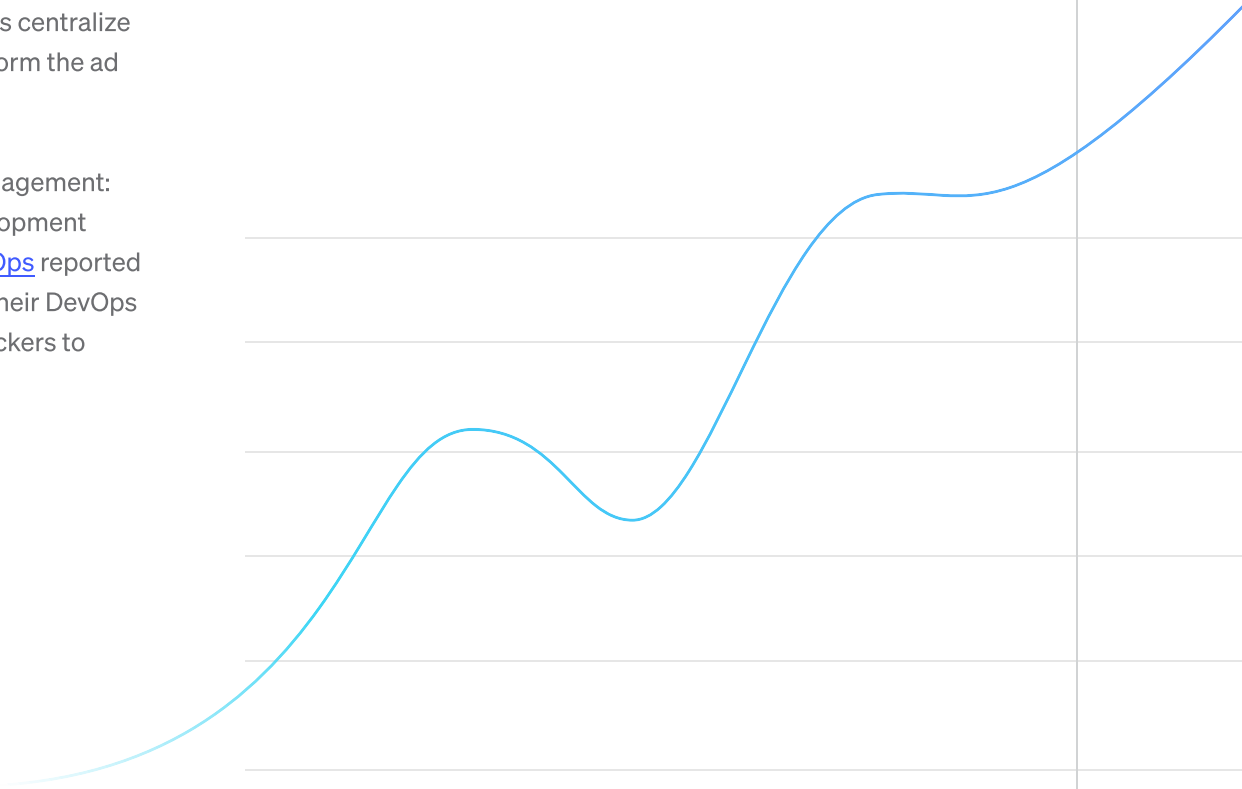


In all of the above examples, you see a decline in performance from the initial, ad hoc adoption of feature management to the point when organizations standardize the practice on one team.

As adoption grows, performance gets back on the uptick, and feature management can be successfully replicated because the infrastructure is in place. Finally, as organizations centralize and standardize feature management, they outperform the ad hoc group across some metrics.

The growing pains stage isn't unique to feature management: we see it play out in other aspects of software development transformation. For example, the [2021 State of DevOps](#) reported on stagnation at companies in the middle stage of their DevOps evolution, when the initial technical and cultural blockers to adoption shift to almost entirely cultural.

The good news (in both DevOps and feature management) is that unlike for early adopters, there is now a host of resources to help with the learning curve. Organizations can draw from prior experiences and processes of similar organizations, and by observing and engaging members of their community or consultants.





# A permanent role in applications

**Feature flags are sometimes thought of as fleeting: primarily for experimentation, to test new functionality in production, or for targeted rollouts.**

In fact, our survey findings show that many organizations' initial use for feature flags was actually for core application behavior, and adoption of this use case has increased. The following use cases were cited as the top two ways companies are actively using feature flags.



# How does your company use feature flags?\*

// asked among those who use feature flags

\* This question was multiple answer.

## Long-term operational controls

The majority of respondents (55%) use feature management as a mission-critical component of the operation of their application. This encompasses dynamic configuration—changing the behavior of an application on the fly, without having to deploy new code.

### For example:

- API rate limiting in the event of a DDoS attack or based on customer tiers
- Changing the mode of your server logs in response to error rates to enable quicker debugging
- Activating a circuit breaker to disable a service causing an outage
- Linking observability tools with feature flags to create automated or self-healing systems

**55%** Long-term operational controls

**47%** Entitlements and plan management

### Long-term operational controls

// Using permanent flags to change back-end configurations on the fly (e.g., when APIs are getting throttled or response times are surging)

### Entitlements and plan management

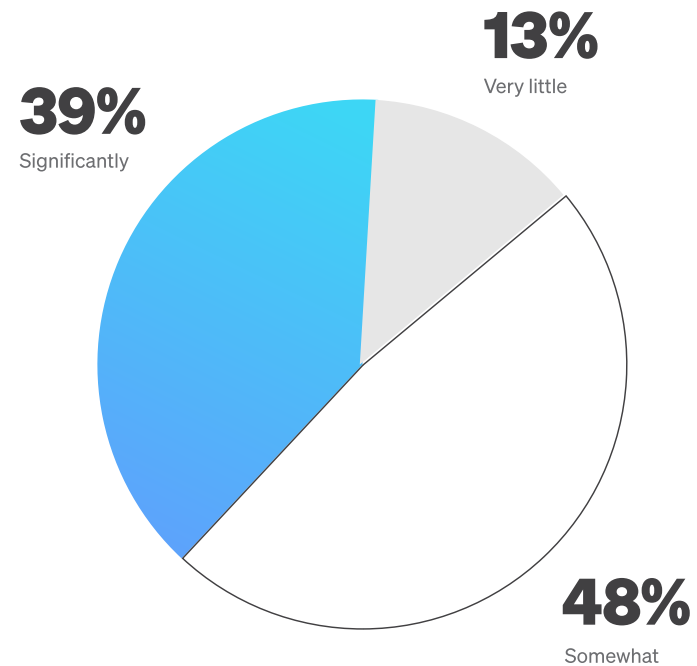
// Granting or denying access to feature sets or entire versions of your application (e.g., enabling premium features for premium customers)



# To what extent has feature management improved application reliability?

// asked among those with a system to manage features

Feature flags used as permanent infrastructure controls are an integral, permanent part of how an application functions, and can help reduce risk in your operations.



When we posed the question, “to what extent has feature management improved application reliability?,” the majority of respondents agreed that feature management has improved their reliability at least somewhat, with 39% confirming that feature management has *significantly* improved application reliability at their organization.



# Managing entitlements

The next most popular use case for feature management (47%) is **entitlements and plan management**.

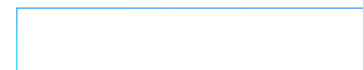
Entitlements and plan management are ways that an organization can grant or deny access to feature sets or versions. For example, if you've ever logged into a premium version of a website to avoid ads or paywalls, you have used entitlements.

One of the benefits of using feature flags to manage entitlements is that instead of creating a custom build to grant access to a feature, you can wrap the feature in a flag which is then easily toggled by the team member best placed to manage access. For example, your sales team can grant access to a premium feature for a limited time to encourage a customer to upgrade.

Decisions about pricing and packaging entitlements are not an engineering responsibility; these are usually the domain of customer-facing teams, such as product, sales, or customer success. No salesperson wants to have to request changes from their support or engineering team to extend a customer's

free trial or renew their license at the last minute. Using feature flags to manage entitlements saves engineering time, empowers the most relevant team members to make decisions about entitlements and access, and results in a better, more personalized experience for users. It can be as simple as flipping a switch.

Managing entitlements may involve granting access on a limited or one-off basis, but managing access to features is not a fleeting job for a feature flag: it's a permanent function dictating core application behavior. With nearly half of respondents citing this as a top use for feature flags, it's clear that users are seeing the benefits.



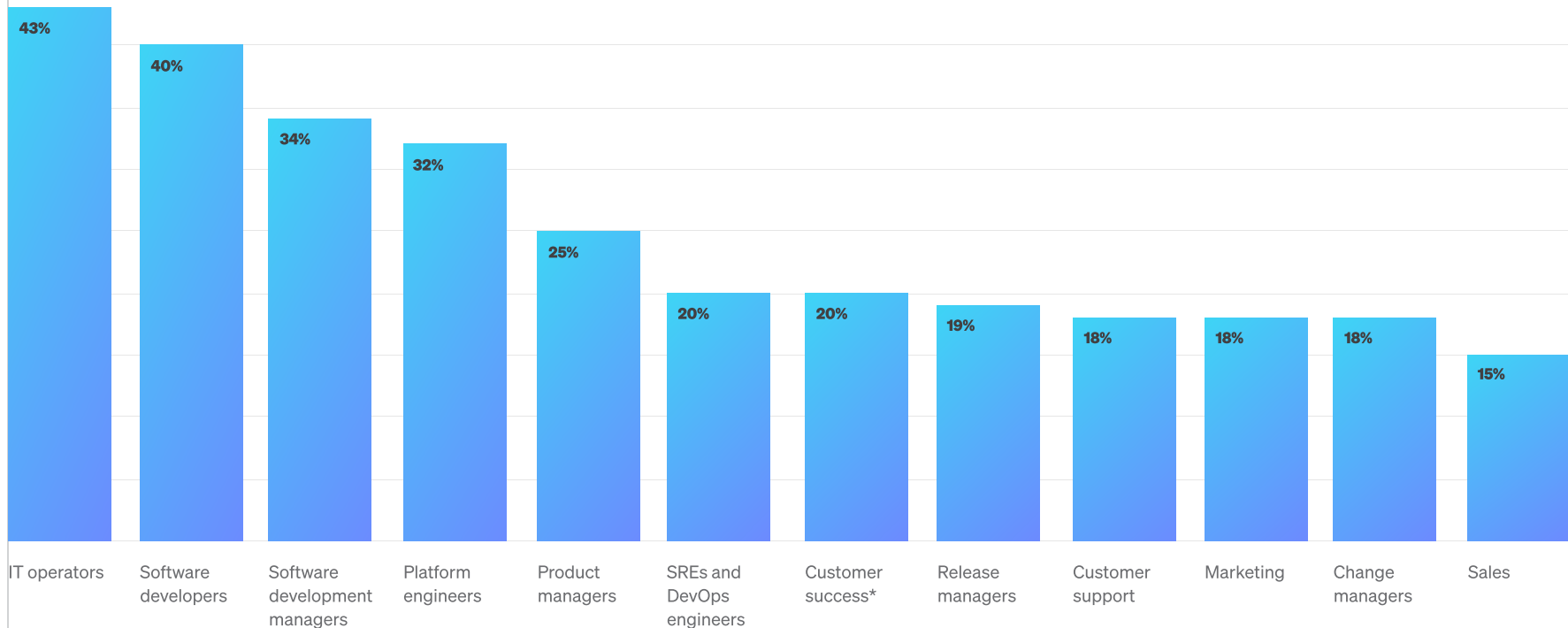
# Who interacts with feature flags the most

Generally, we get more insight into *how* organizations are using feature management based on *who* is using it the most.

So we asked respondents about who at their company directly interacts with feature flags, and IT operators was the top response at 43%, nudging out even software developers (40%).



# At your current level of adoption, who at your organization interacts with feature flags in some way?



*\*those who manage certain customer accounts*





## IT operators are leading adoption

The high prevalence of IT operators is consistent with long-term operational controls being the primary use case at most respondents' organizations. We explored the common long-term uses for feature flags in the previous section: API rate limiting, activating circuit breakers, or adjusting server log modes are all examples of the types of feature flags an IT operator is likely to build and manage.

Modernization and migration was the third most common use case for feature management at 40%, which also tracks with IT operators' high use of feature flags. Ops teams—ranging from DevOps, SRE, platform, and more—can use feature management to safely optimize applications for the cloud.



## Cloud migration

Teams can gradually migrate to new cloud infrastructure by controlling the flow of web traffic between legacy and new endpoints with feature flags. By slowly increasing traffic to the new cloud environment while validating performance (and rolling back to the legacy system as needed), teams can ensure business continuity and protect end-user experiences throughout a migration.



## App modernization

An organization's cloud journey does not end once they've migrated to public or private cloud infrastructure. Teams must continue to explore new technologies and application components to help improve costs, efficiency, and end-user value. The same gradual migration technique applied in a cloud migration can be leveraged for application refactoring, re-platforming, and re-architecting.

By adopting new architecture, infrastructure, and application components gradually via feature management, teams can reduce the likelihood of costly outages and more quickly understand the performance and cost impacts of changes to their technology stacks.



## Software developers also interface with feature flags a lot

The two other groups that interact with feature flags the most are software developers (40%) and software development managers (34%). These results aren't surprising, given that many of the common use cases for feature management are most relevant to software development, like building and delivering new features, bug fixes, and other code changes.

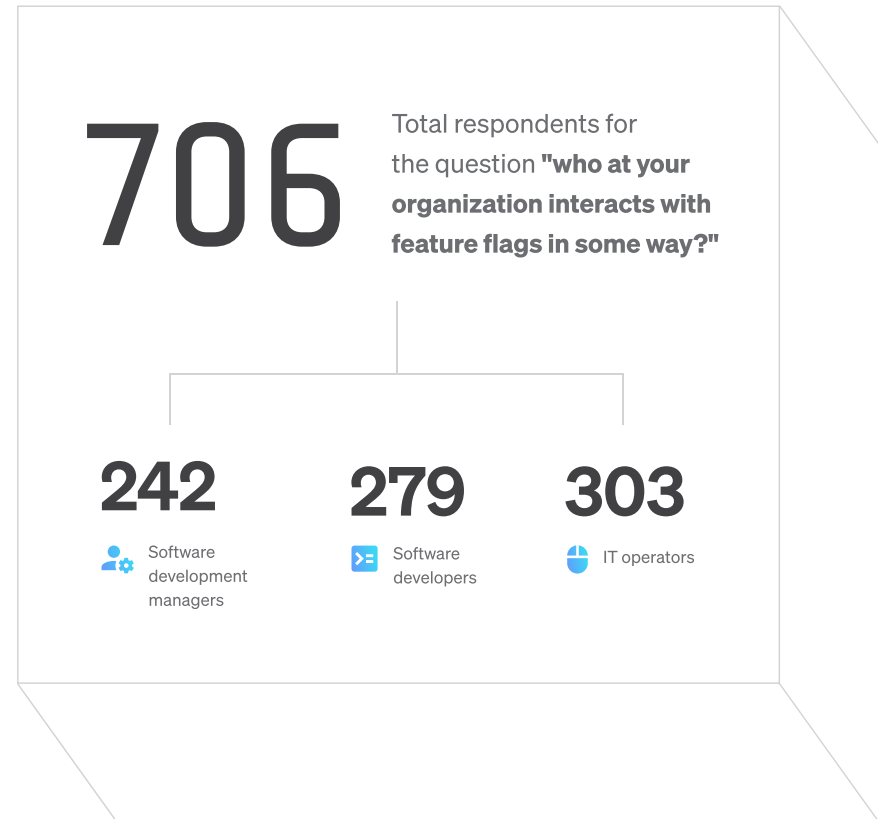


# How many developers have flag permissions

Of note in our survey results is the gap between larger and smaller companies with regards to who has feature flagging permissions:

**62%** of those at **companies with under \$500 million in revenue** say less than half of developers have permission to create and manage feature flags.

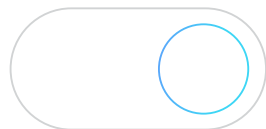
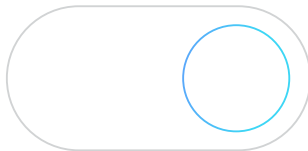
**49%** of respondents at **companies with \$500 million dollars in revenue or more** say the same.





This challenges the cliché that larger organizations automatically have more restrictive processes and red tape. Large, nimble companies that invest in tools and training may be starting a trend in which they reap more benefits from feature management and continue widening the gap between big businesses and their smaller counterparts.

Leaders who want to increase adoption of feature management at their organizations may want to discover who has access to creating and managing feature flags, and work to expand granting those permissions to encourage adoption.



## Platform engineers are also seeing lots of feature flags

The fourth top response was platform engineers, at 32%. It appears that platform teams are also beginning to see the value in feature management as they seek both to deliver features to their internal customers and to enable feature management for other teams in their org.

One such use case for platform teams could be “providing a foundation for continuous delivery,” which was the second reason why people use feature flags (39%, just a hair behind “better quality control for new features,” 40%). Decoupling deployments from releases is a big perk of using feature flags, enabling quicker and more frequent deployments. Platform teams can integrate feature management into their internal developer platform, enabling self-service feature flags for other teams and paving the way to continuous delivery.



# The financial—and emotional—ROI

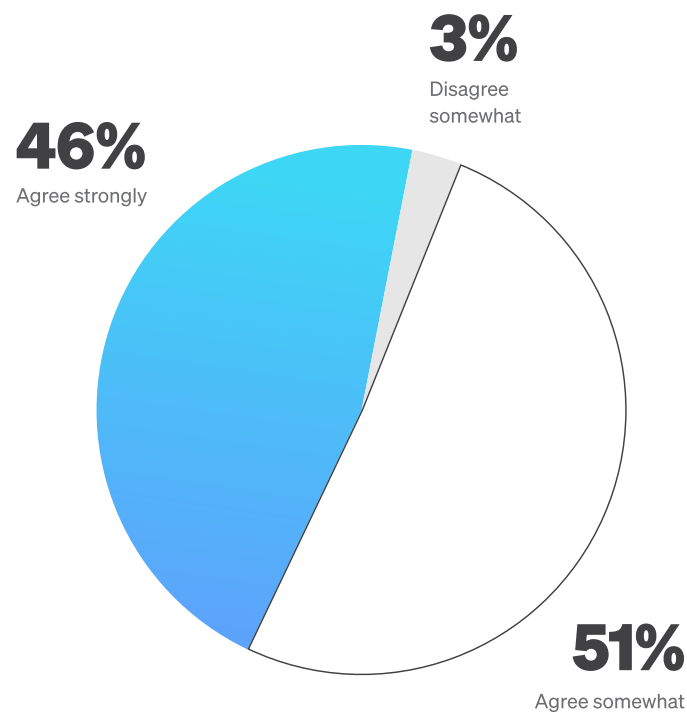
The earlier sections of this report cover the investments teams are making in feature management. We've also shown how teams are using feature management, and why they chose to get started with it. **In this section, we'll highlight some of the benefits teams are getting from feature management.**



How strongly do you agree or disagree with the following statement?

## Feature flags save our company money and deliver a demonstrable ROI.

// asked among those who use feature flags



Almost unanimously (97%), teams say that **feature management is saving them money** while giving them enough measurable results to showcase the value feature management is bringing to their organization.

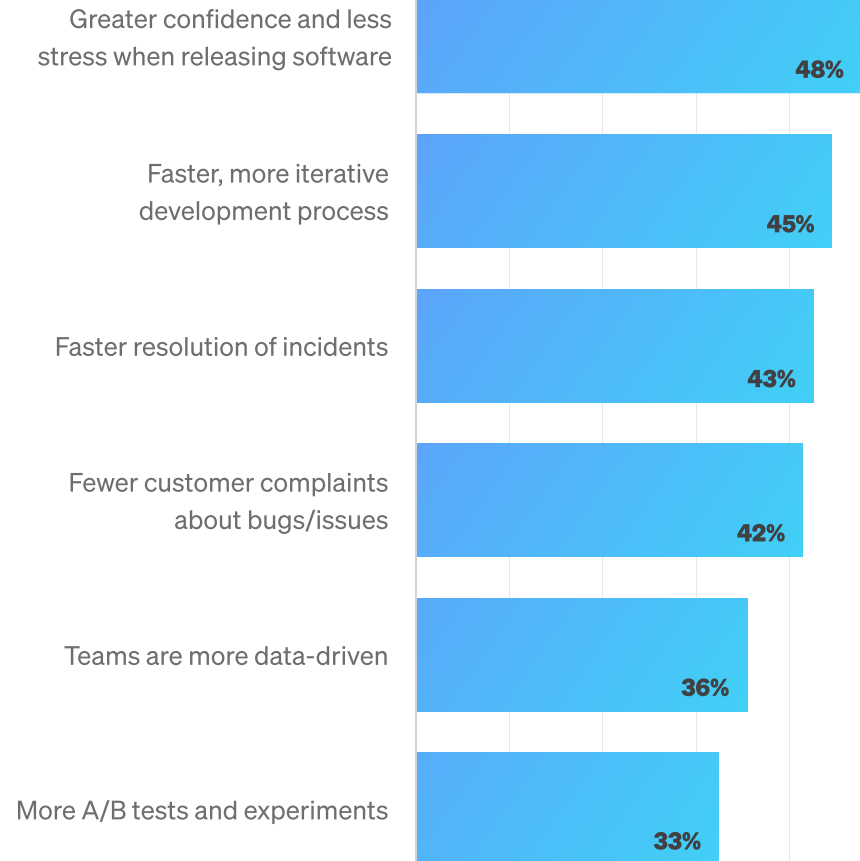
Some of the examples we've seen with feature management adoption are increased deployment frequency, improved reliability, and reduced MTTR, all of which are measurable benefits that can be used to motivate continued or increased investments in feature management. What about the intangible benefits?



# Which of the following improvements have you seen since your company adopted feature management?

// asked among those with a system to manage features

The greatest area where organizations are seeing improvement is **“greater confidence and less stress when releasing software” at 48%.**



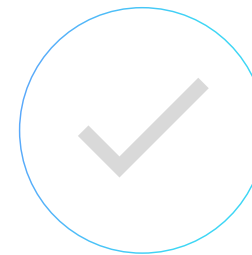
This finding is consistent with data from our [2021 State of Feature Management report](#), in which the top reported benefit of feature management was “reduced or mitigated deployment risk,” with “improved confidence around deployments” in second place.

We’re not surprised that reduced stress and greater confidence are called out by respondents.

The biggest stressor for developers is “deploying updates more quickly,” according to [Release Assurance: Why Innovative Software Delivery Starts with Trust and Psychological Safety](#), our report released earlier this year.

When you alleviate the stress associated with deployments by decoupling them from releases, developers are free to test and experiment with new features without dread. And that, in turn, can lead to more innovation in your product.

Benefits like “faster resolution of incidents” (43%) and “fewer customer complaints” (42%) are both tangible and provide significant emotional benefits: ask any engineer on call.



# Wrapping up

If there's one thing the past few years has taught us, it's to avoid bold predictions. That said, given the data in this report, we feel optimistic feature management is on a positive trajectory:



Budgets increasing



Adoption spreading and scaling



Organizational usage maturing, revealing greater potential



Deeper integration within applications

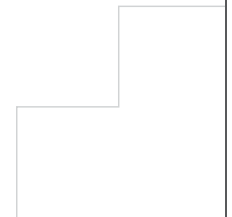


Strong ROI and improved quality of life

---

What we're seeing is teams developing a deeper trust of feature management, and using that to do amazing things.

With so many organizations leveraging feature management to better control and release their software, we all stand to benefit.



# Thank you

Thanks to everyone who contributed to this report. We look forward to learning about how your use of feature management changes in the coming year.

If you have any questions about our methods or conclusions, please let us know [HERE](#).

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

