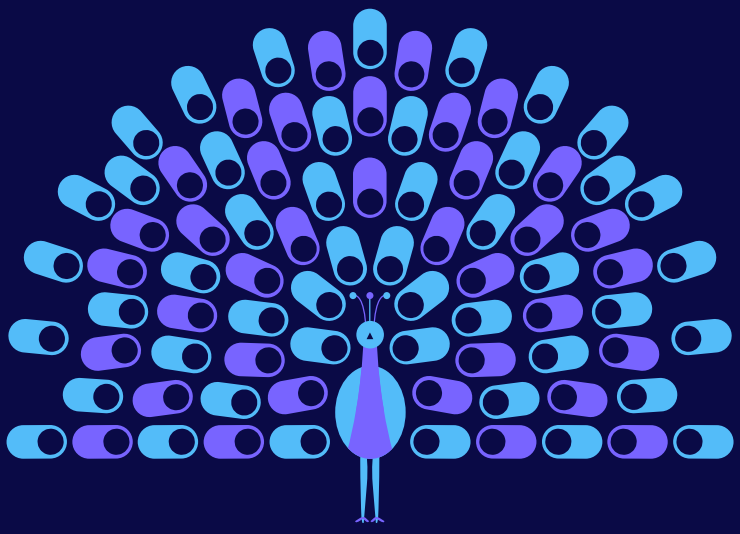
The State of Feature Management and Experimentation 2024

We asked 500 engineering leaders how they are releasing features in 2024. Here's what we found.



LeadDev

in partnership with

split by harness®

A note from the editor

Tinkering has always been at the heart of software engineering. Whether you are just starting out with a new programming language (hello world!), or trying to fix an annoying bug, you will inevitably end up trying some things and seeing what happens.

But how do you scale that approach and ensure that engineering teams are comfortable experimenting with new ideas, testing them gradually, and measuring the results? This is where feature management and experimentation come in, but surprisingly, it is not yet a universal practice.

Alongside our partner Harness, we wanted to better understand why that's the case, just how widespread feature management and experimentation practices are today, what tools are being used, and, perhaps most importantly, how the best organizations are doing it and why.

What we started to see was a clear maturity curve, from those who are still dabbling with ad hoc feature toggles and experiments, all the way up to the top 1% of respondents who have truly baked feature management into their software development practices.

Where are you on that curve?

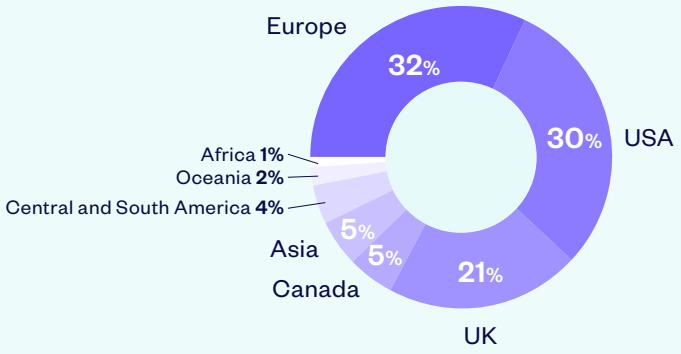
Scott Carey
Editor in Chief, LeadDev

Methodology

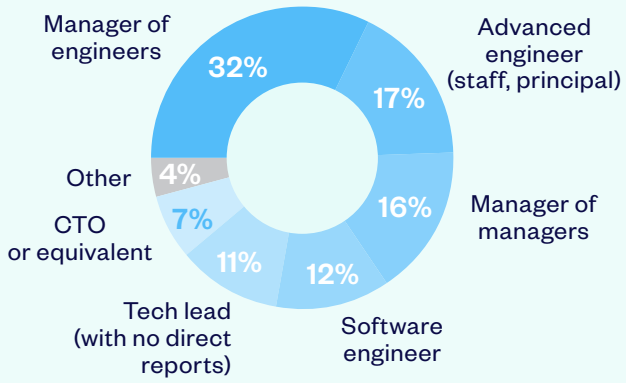
We conducted the survey in July 2024, receiving 496 completed surveys.

Respondents

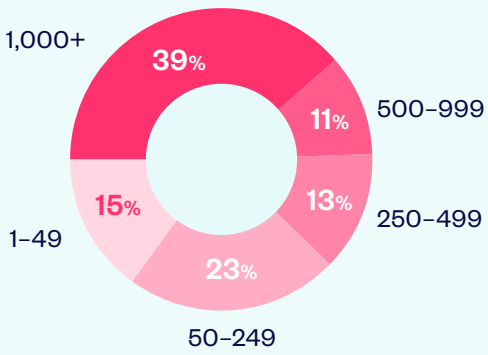
Location



Job role



Company size (employees)



5 top takeaways

38% say the main reason they do feature management and experimentation is to bring new features safely to market

42% have engineering take the lead on designing and executing feature experiments, followed closely by product

32% see feature management and experimentation as a key priority

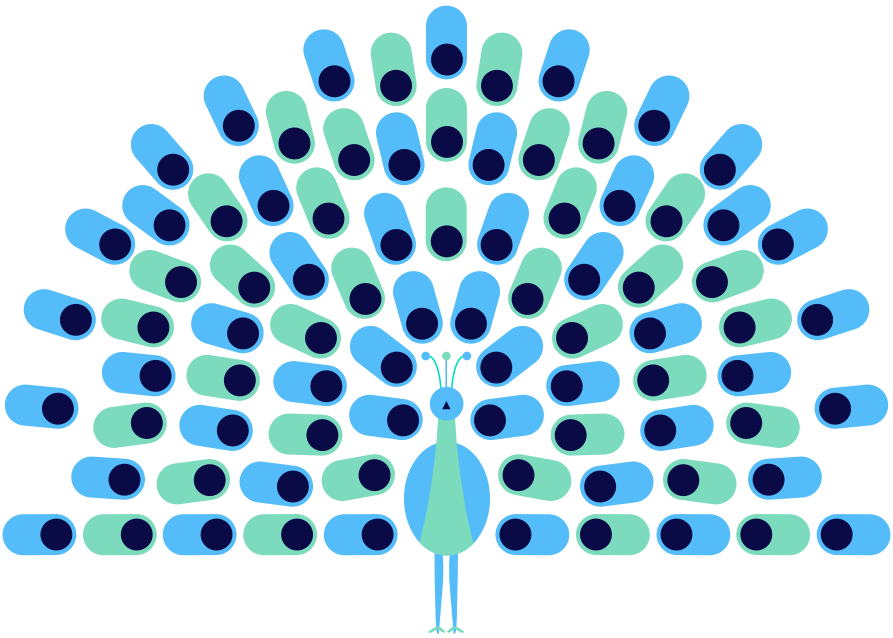
58% are using their own tools for feature management

84% feel like they have been at least "somewhat successful" with their feature management and experimentation efforts

Questions answered in this report

We wanted to gather the data to answer questions like:

- Why and when do engineering teams run experiments?
- How important are feature management and experimentation to technical strategy?
- What tools and methods are currently being used to manage new features?
- How successful do engineering teams think they are?



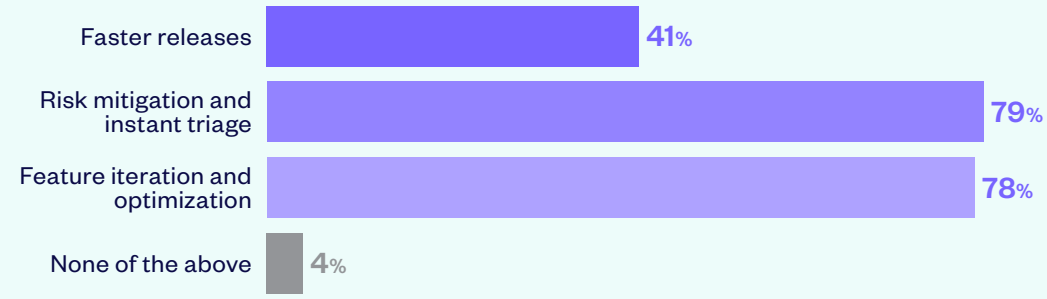
1. Why manage features anyway?

Summary

- Risk mitigation and feature optimization are the twin pillars of feature management, viewed by **78%** of respondents as the core role of these tools and processes.
- Feature safety is the most important concern amongst respondents (**38%**).
- **54%** of respondents are practicing progressive delivery.

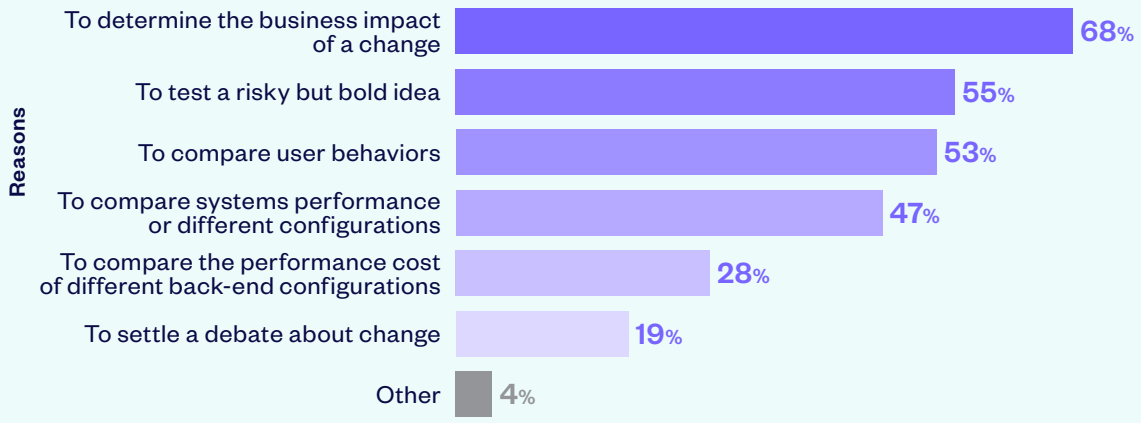
Feature flags are a versatile tool in the software developer's armory, allowing teams to toggle features off or on within an application. Strategic feature management can help speed up delivery cycles, experiment with new ideas, and help developers sleep better at night knowing that any changes can be safely rolled back.

This research shows that the heart of effective feature management lies in two key strategies: minimizing deployment risks and optimizing features. In fact, 78% of respondents agree these are the primary reasons for managing and experimenting with features, no matter the company's size or where it operates.



Q: What role does feature-level monitoring and experimentation play when releasing new technologies/features?

The primary reason organizations run experiments is to measure the business impact of a change, with 68% of respondents highlighting this as their main goal. This jumps to 79% amongst organizations that feel they are successfully doing feature management and experimentation today. The next most popular reason is to test risky ideas (55%) or compare user behaviors (53%).



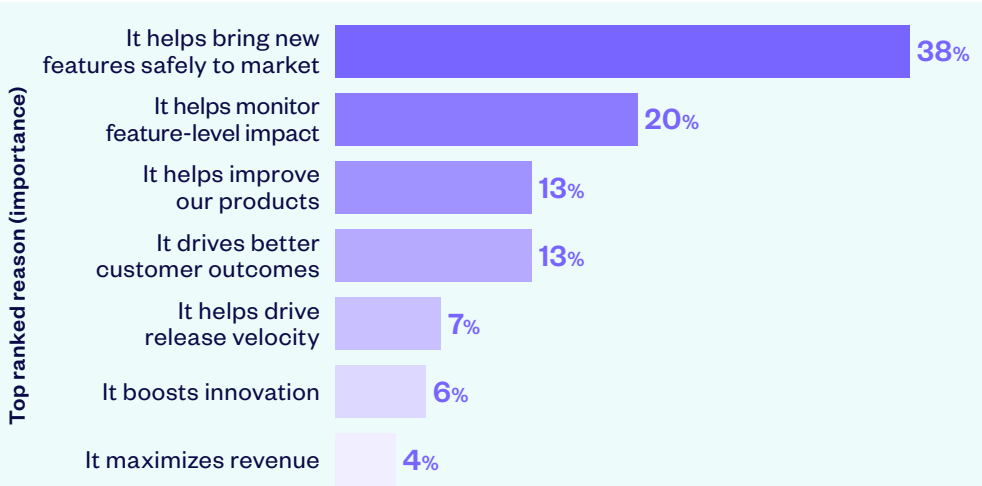
Q: Why do your engineering teams decide when to run an experiment?

Safety first

Drilling down further, bringing features safely to market was clearly the dominant factor for organizations when it comes to employing feature management and experimentation controls into their software development lifecycle, at 38%.

Alongside safety, understanding the impact of a feature and improving products and customer outcomes were also ranked highly.

> "[Feature management] fosters a culture of continuous improvement within engineering teams. By constantly experimenting and measuring results, teams can learn from both successes and failures, driving overall improvement in product quality and user satisfaction." – **anonymous respondent.**



Q: Why is feature management and experimentation important to your engineering teams?

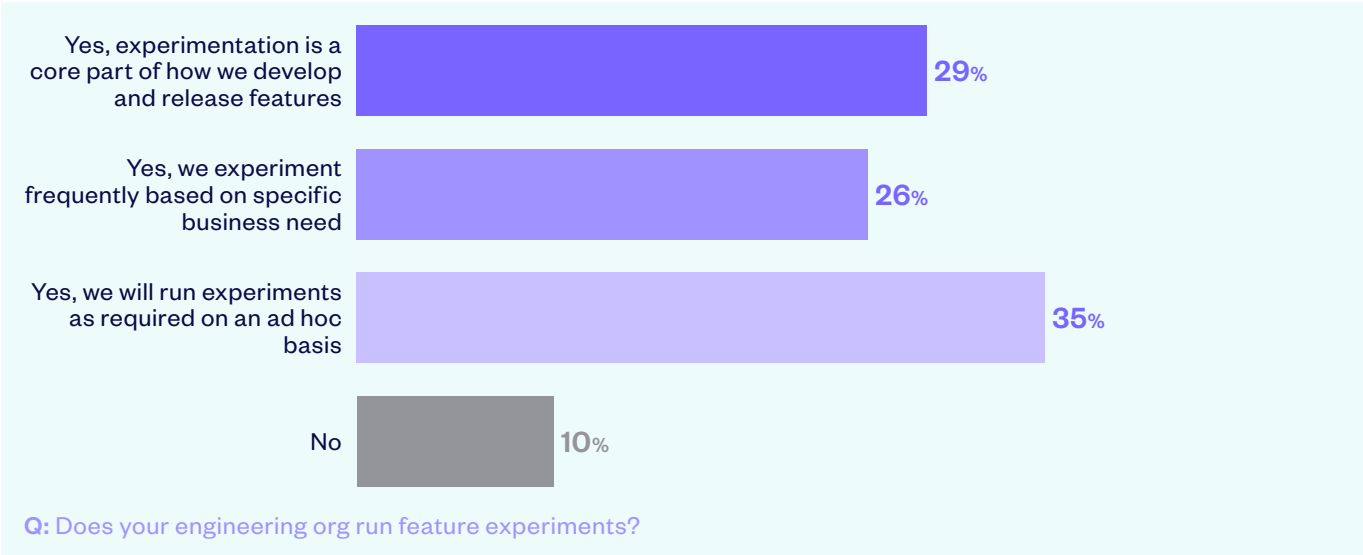
2. Feature management and experimentation in 2024

Summary

- Just over half of respondents are regularly managing feature releases, but **35%** are still only doing it on an ad hoc basis.
- Feature experimentation is far more popular at large organizations.
- CTOs see feature management and experimentation as more of a priority than individual contributors.

There are plenty of organizations that already see the value of these practices, with 55% of respondents regularly managing and experimenting with features. That leaves 35% that still only do so on an ad hoc basis and 10% that don't at all.

Typically, these ad hoc practitioners are trying something and manually gathering results, be it through database queries, processing log files, or checking existing reports. However, this is only the first step toward true feature management and experimentation, which brings statistical and professional rigor to the process.



Q: Does your engineering org run feature experiments?

Too big to fail

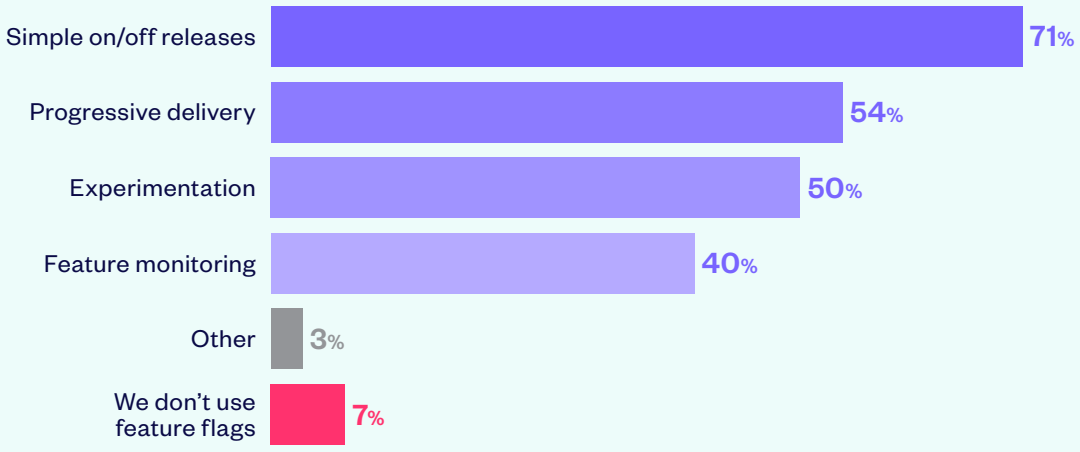
Broken down by company size, **feature experimentation is far more likely to be a central tenet of engineering practice at large organizations**, but is still done on a more ad hoc basis at smaller orgs.

Feature experimentation is seen as a core part of the software development process at 46% of large organizations surveyed, but has only reached this level of criticality at 32% of organizations with less than 250 people. 49% of those smaller organizations aren't running feature experiments at all.

Company size (employees)	Yes, experimentation is a core part of how we develop and release features	Yes, we experiment frequently based on specific business need	Yes, we will run experiments as required on an ad hoc basis	No
0–49	13%	12%	17%	16%
50–249	18%	20%	**26%**	**33%**
250–499	14%	16%	11%	8%
500–999	8%	13%	13%	4%
1,000+	**46%**	39%	32%	39%

Q: Does your engineering org run feature experiments?

In terms of usage, 71% of respondents are doing simple on/off releases, with 54% practicing progressive delivery. This is a set of software development practices aimed at gradually and safely releasing new features to users and subsets of users, starting with on/off releases but also incorporating canary testing, A/B testing, and feature-level monitoring. 50% of respondents are using these tools to experiment with features, and just 40% are monitoring their features.



Those who responded that they don't use feature flags said:

- CTO believes it will slow the team down and isn't worth it.
- We need to start using those. We haven't found a good system (overall tool) yet.
- Because the company is old and slow.

Q: How do you currently use feature flags?

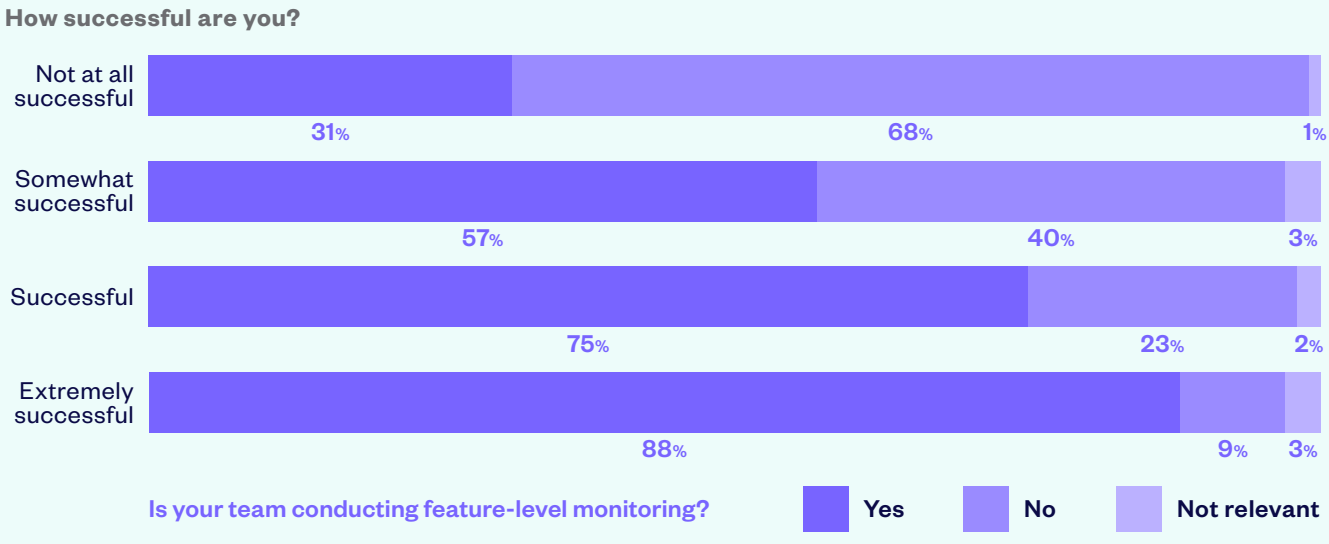
The importance of feature monitoring

For the 143 companies that have experimentation at the heart of their release processes, experimentation becomes the most popular use case (70%), but these companies have also graduated to monitoring features (62%), using on/off releases (63%), and practicing progressive delivery (65%) at a much higher rate than those that are experimenting either "frequently" or on an "ad hoc" basis.

Current uses of feature flags	Experimentation is a core part of how we develop features	We experiment frequently	We run experiments on an ad hoc basis
Simple on/off releases	63%	74%	77%
Progressive delivery	65%	59%	49%
Feature monitoring	62%	46%	25%
Experimentation	70%	65%	35%
Other	5%	1%	4%

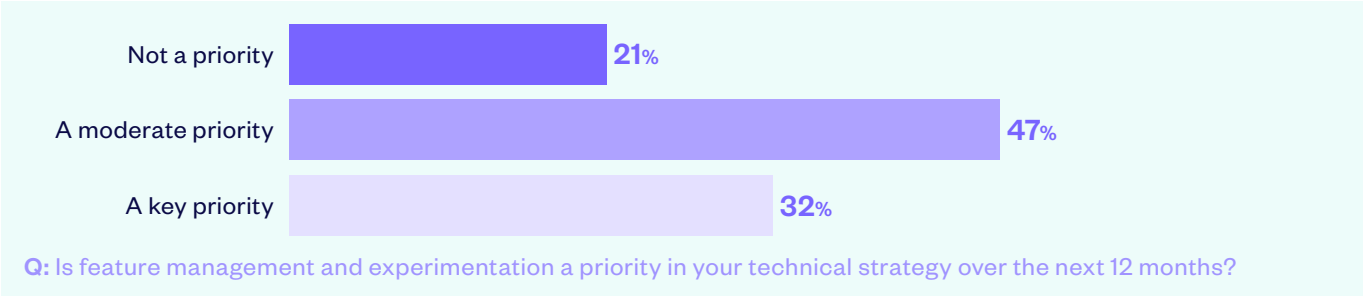
Q: Does your engineering org run feature experiments?

The ability to effectively monitor features is a clear indicator of success with feature management and experimentation efforts. Of those respondents who feel they are successful or extremely successful so far, 82% are monitoring at the feature level. Only 16% of those organizations that aren't monitoring at that level feel they have been successful with their feature management and experimentation efforts so far.



Priorities, priorities

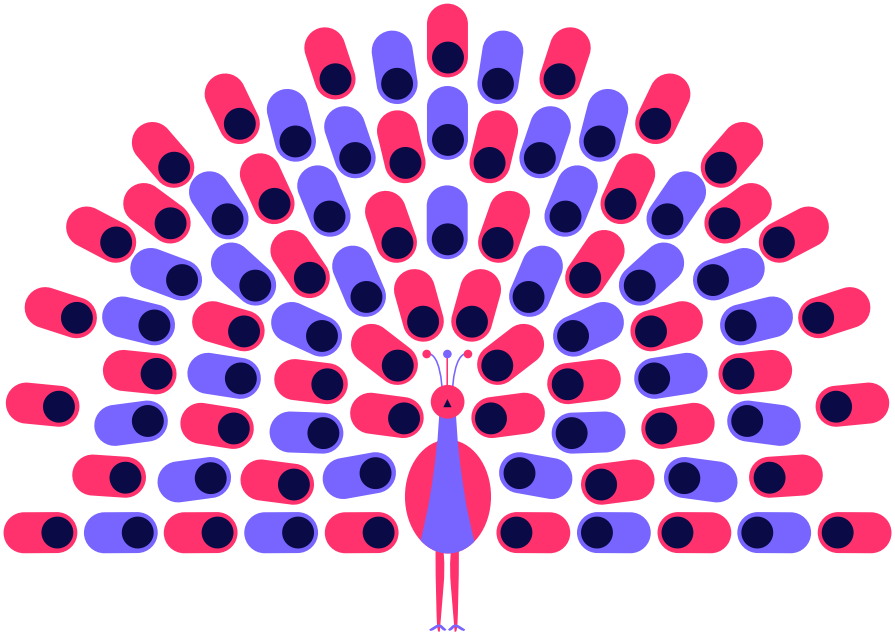
While just 32% of respondents see feature management and experimentation as a key priority for the year ahead, 44% of CTOs saw it as a major priority. **This shows a clear gap between software engineers and engineering leaders regarding the importance of feature management.** Only 29% of individual contributors believed this to be a key priority, and 31% of engineers think it shouldn't be a priority at all, compared to just 8% of CTOs and 16% of tech leads.



Q: Is feature management and experimentation a priority in your technical strategy over the next 12 months?

Feature management and experimentation are also not seen as a priority at the 33% of organizations that are still managing features on an ad hoc basis, with 54% marking it as a moderate priority for now. For organizations that experiment frequently, it's a key priority for 62% of respondents.

This gap seems to come from the top too, with just 6% of senior leadership seeing feature management and experimentation as a key priority at organizations that are still doing it on an ad hoc basis. Compare this with 83% of senior leaders seeing it as important or a key priority at those organizations that have made experimentation a core part of their developer culture.



3. Barriers to entry

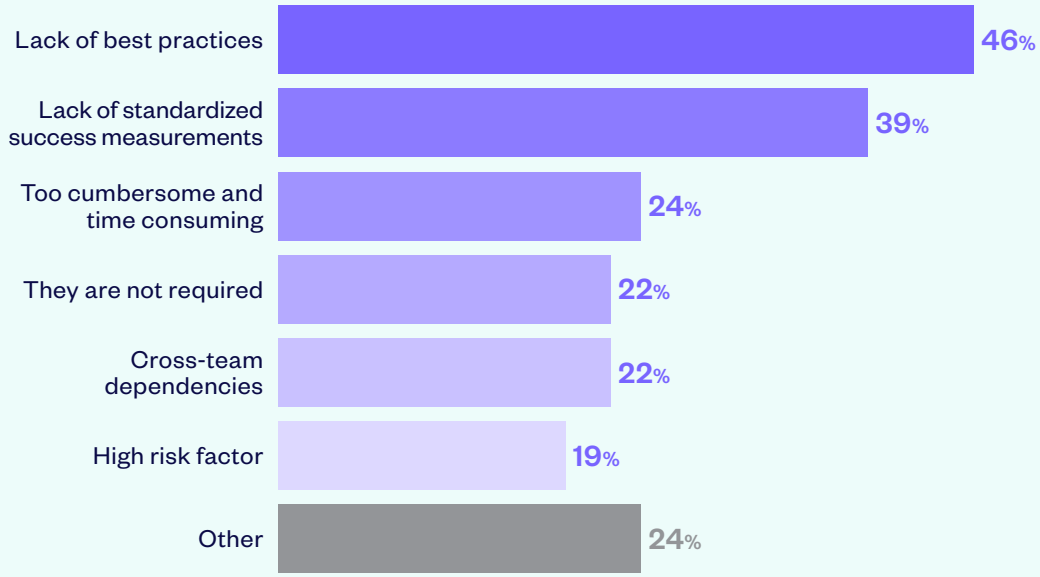
Summary

- Managing culture change and lack of best practice are holding organizations back.
- **31%** of respondents see feature management and experimentation as nice to have.
- Technical debt isn't a significant concern for **63%** of respondents, but needs to be carefully managed.

For those organizations not running feature flag experiments, lack of best practices was the biggest hindrance, holding back 46% of those orgs. Lack of standardized success metrics was the next largest barrier at 39%. A range of other factors impacted 20% of respondents, including a feeling that it is too time-consuming or unnecessary.

Being able to make the kind of cultural change that puts feature management and experimentation at the core of development practices was also highlighted by several respondents as a key blocker.

> "Culture needs to shift, our user experience (UX) and product do not engage with tech at the discovery phases, instead preferring to throw research items over the fence. We are moving to a more coordinated product model that should allow the teams to be involved in discovery earlier." – **anonymous respondent.**



Q: Why don't you run feature experiments?

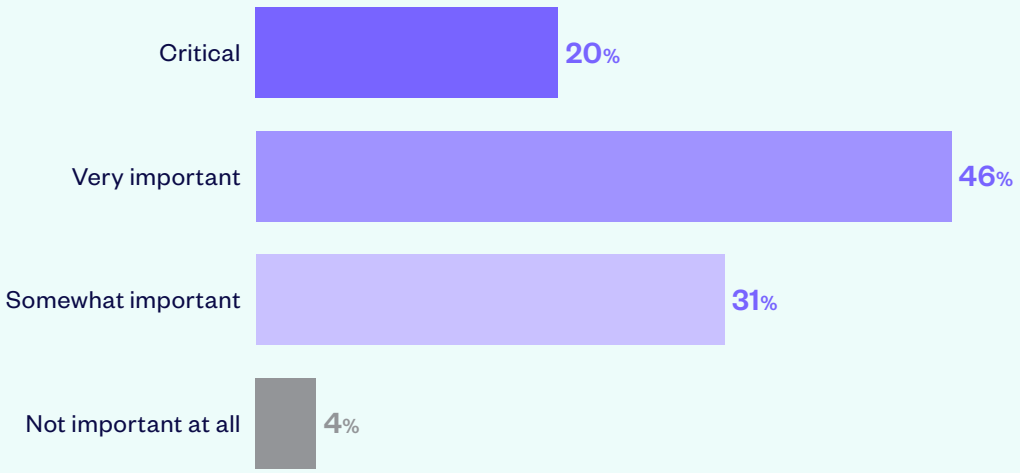
Culture change and tools, so everyone uses the same tools and speaks the same language.

Training on the theory of experimentation, understanding of the tools more deeply and cultural change to see experimentation not as a release tool but a way to verify ideas with real users

Tooling and cultural change so teams have autonomy to experiment

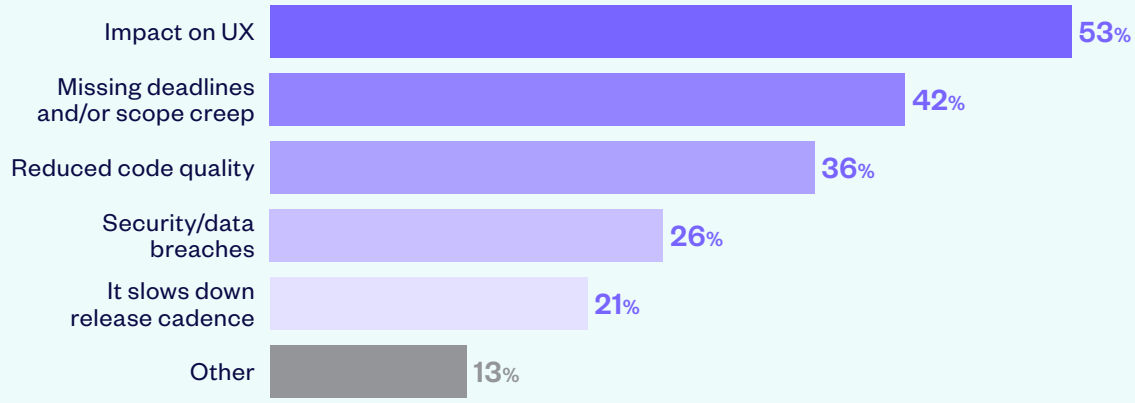
Q: What do you need to improve? (training, tools, culture change, etc.)

There is also a criticality factor at play. While 46% of respondents see feature management as very important and 20% as critical, 31% see it as more of a nice to have. Just 4% don't see the value at all.



Q: To what extent is feature management and experimentation important when releasing new software?

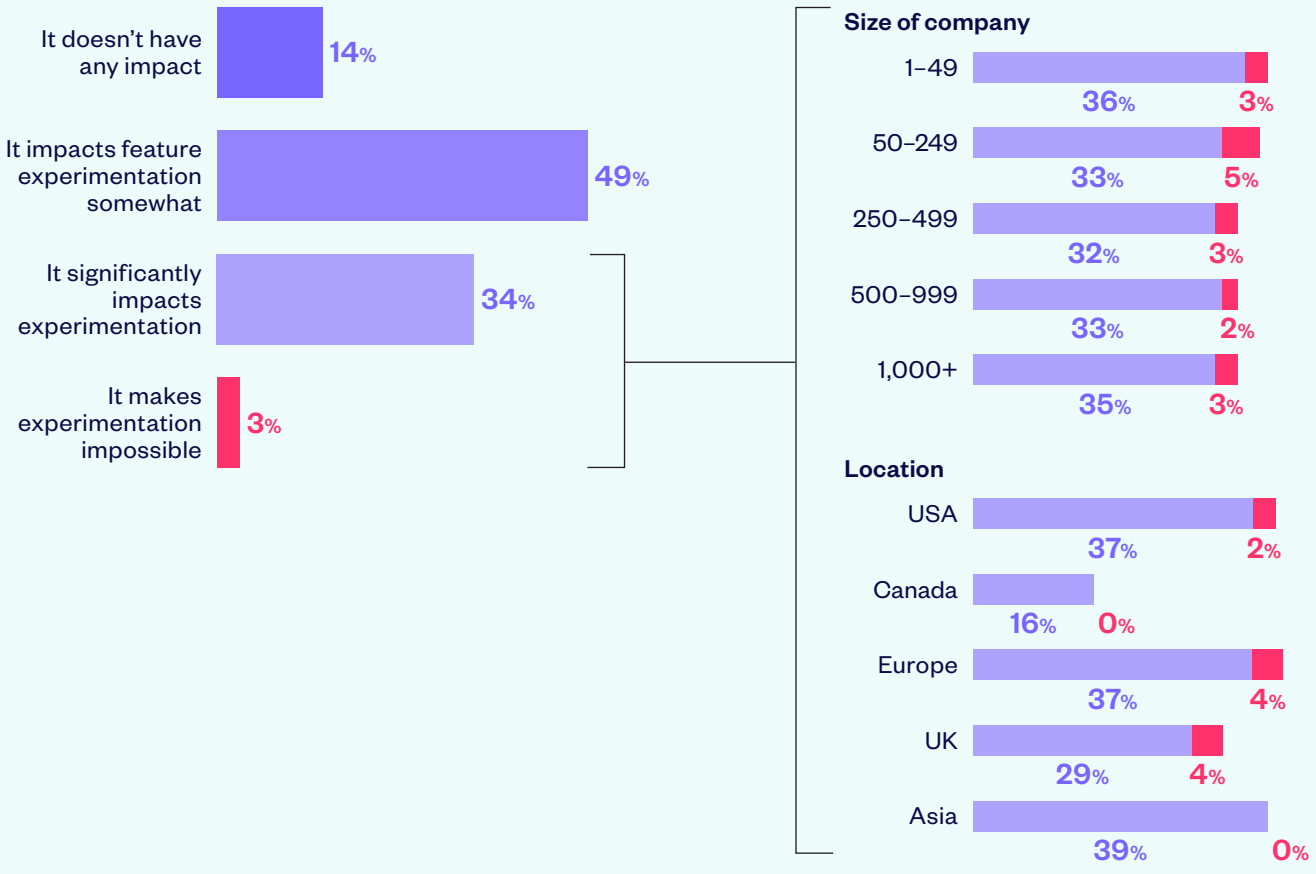
In terms of risks, the impact on UX was the biggest concern for 53% of respondents, outpacing scope creep (42%), and a reduction in code quality (36%). Security and data breaches was the biggest concern for 26%, and slowing down the release cadence for 21%.



Q: In your opinion, what are the biggest risks of experimentation?

Technical debt

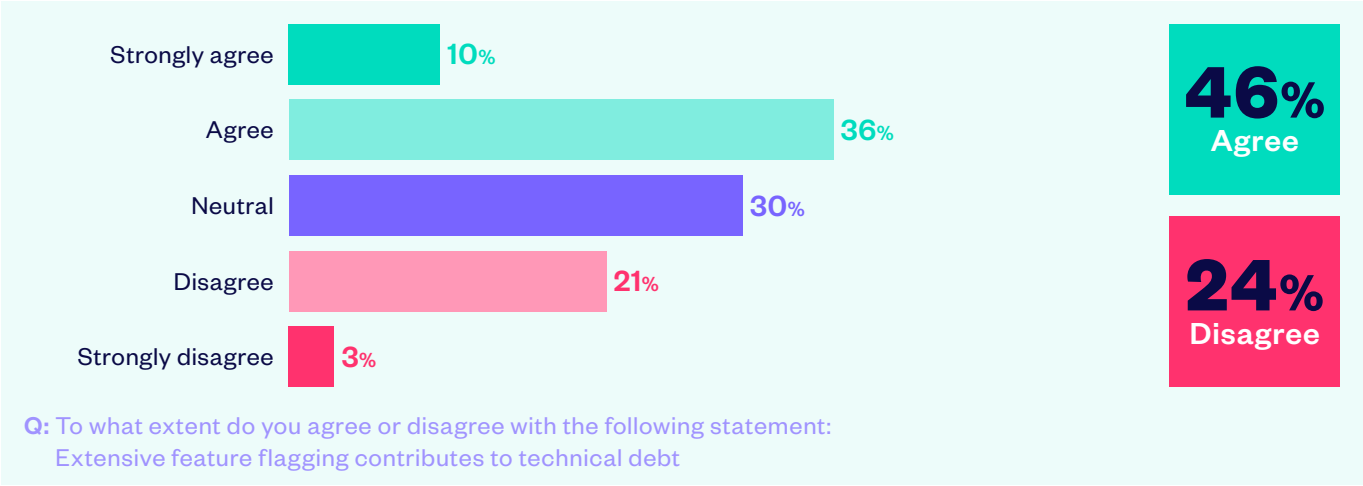
Interestingly, the specter of technical debt isn't a significant barrier for 63% of respondents, but does make feature experimentation significantly more difficult to implement for 34% of respondents and "impossible" for 3%.



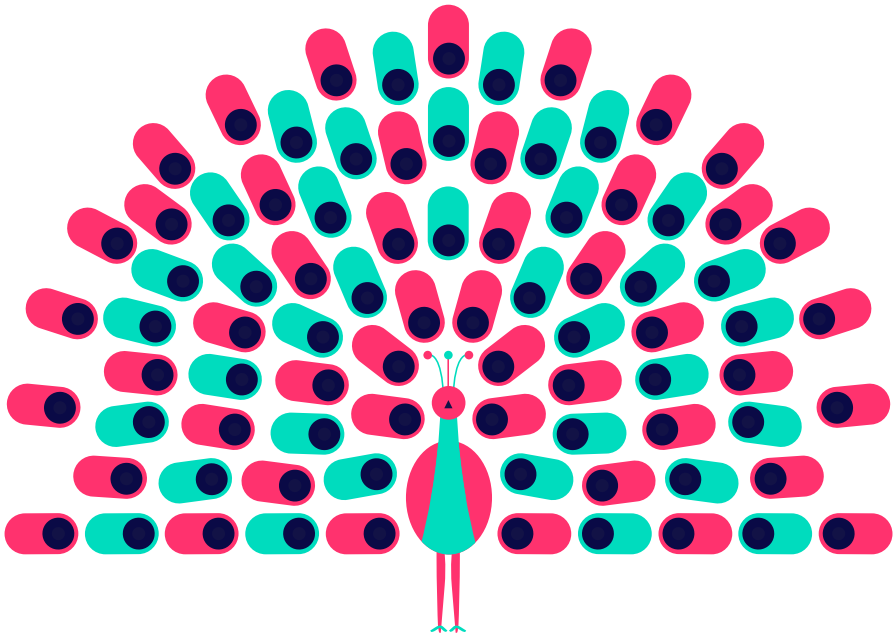
Q: To what extent does existing technical debt impact your team's ability to perform experiments?

Feature management can also contribute toward technical debt if flags and experiments aren't properly managed and deprecated over time. 46% of respondents agree that extensive feature flagging contributes to technical debt and 30% are neutral on the topic, leaving just 24% who **don't** think it's an issue.

A handful of respondents cited the 80/20 rule of dedicating 20% of engineering time to help keep a lid on technical debt, and establish practices to regularly review experiments and remove old flags after a set period of time, say 90 days.



Q: To what extent do you agree or disagree with the following statement: Extensive feature flagging contributes to technical debt

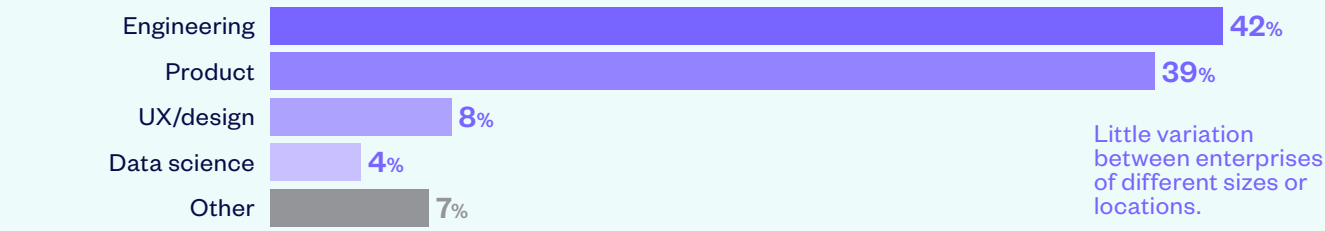


4. How are you managing features?

Summary

- Engineering or product are equally likely to take the lead on feature management.
- The buy vs build debate is yet to be settled in this space, with **33% buying** and **31% building** their own.
- In terms of key measures of success, error rates are the most common metric, used by **74%** of respondents.

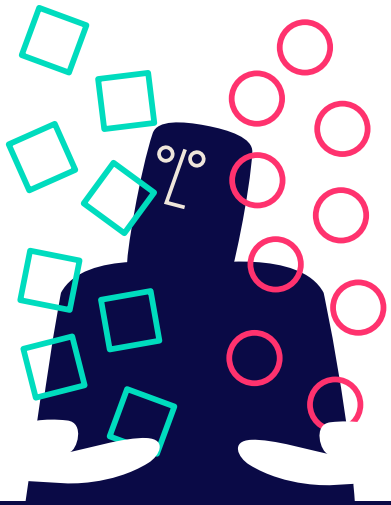
Despite our research primarily targeting engineers, 42% of organizations have the **engineering team leading** on feature management and experimentation, marginally ahead of **product** at 39%. Another 8% leave this in the hands of the design or UX group, and 4% the data science team.



Little variation between enterprises of different sizes or locations.

Q: Which teams lead on designing and executing experiments/tests?

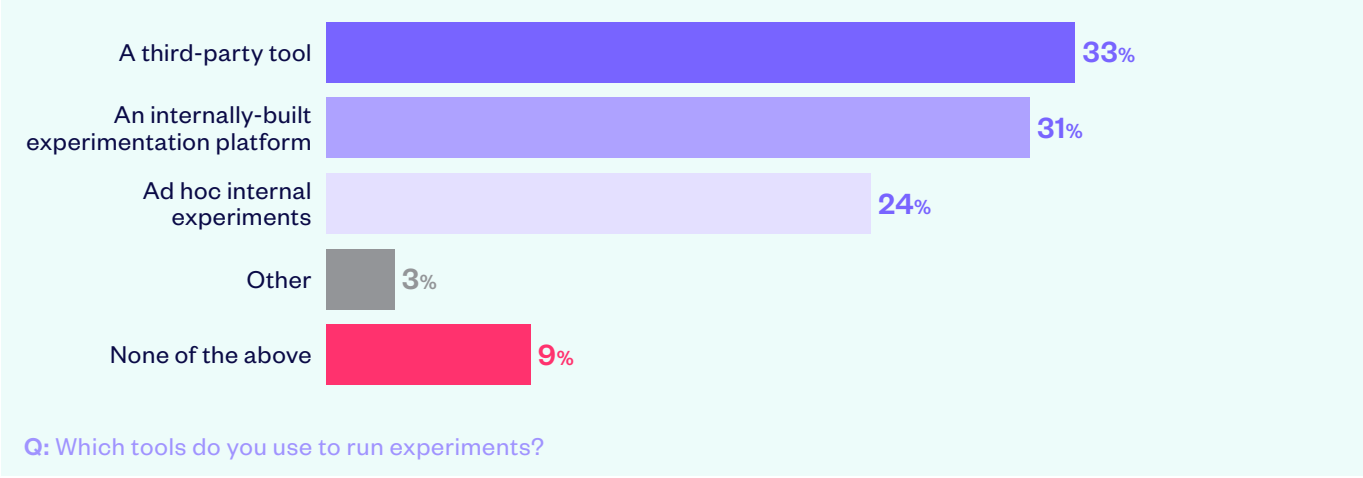
In terms of how this translates to perceptions of success, engineering-led efforts are more likely to feel they are **not** having any success (49%) than product-led efforts (35%). On the other side of the coin, product-led teams are more likely to feel they are doing feature management and experimentation **extremely well** (42%) than their counterparts in engineering (36%).



What tools are you using?

In terms of tooling, 33% are using best-of-breed third-party tools to run experiments, and 31% have opted to build their own experimentation platforms.

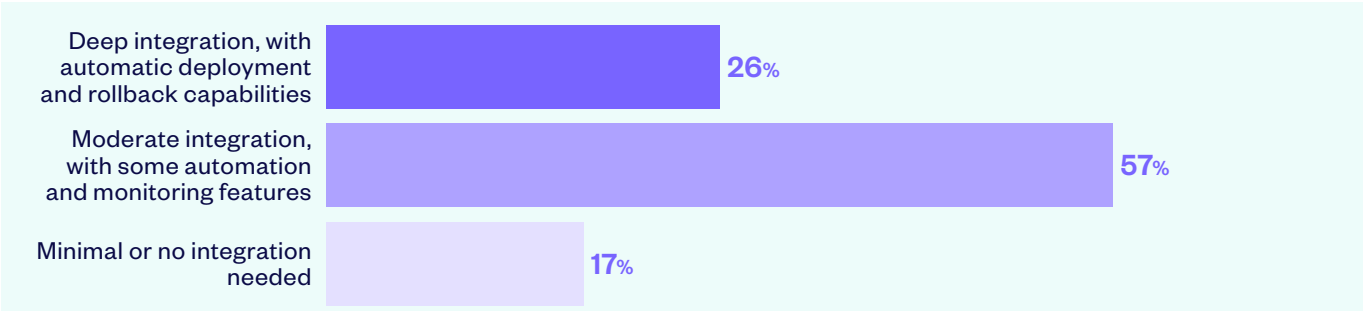
Somewhat surprisingly, large organizations of more than 1,000 employees are the most likely to roll their own, at 42%, with mid-sized companies preferring to buy a solution in 46% of instances.



Q: Which tools do you use to run experiments?

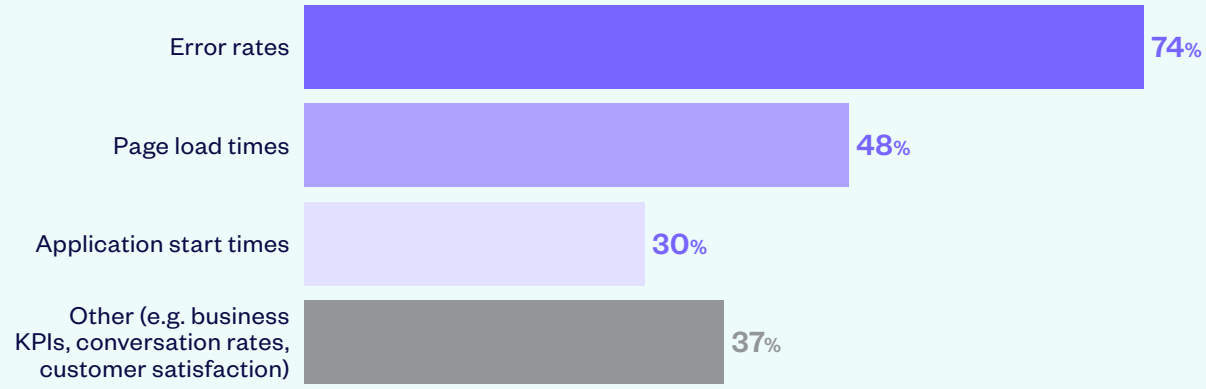
How does this choice impact perceptions of success? Well, just 37% of organizations that choose a third-party tool consider themselves to be either successful or extremely successful in their feature management and experimentation efforts, whereas 50% of those using an internally-built platform feel they have been successful.

When it comes to capabilities, 83% of respondents expect at least a moderate level of integration between their feature management tool and existing CI/CD pipelines, with 26% expecting deep integration with automatic deployment and rollback capabilities. This is important because if feature management is not connected to the CI/CD pipeline, you may fall back into manual coordination, with less ability to automate and enforce policy, and gather end-to-end metrics.



Q: What level of integration do you expect between your feature management tool and CI/CD pipeline?

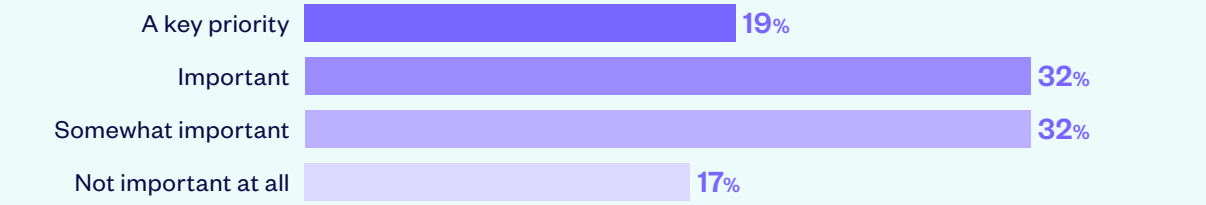
In terms of key measures of success, **error rates are the most common metric** used by 74% of respondents. Page load times are next at 48%, and application start times at 30%.



Q: Which metrics do you use to measure this?

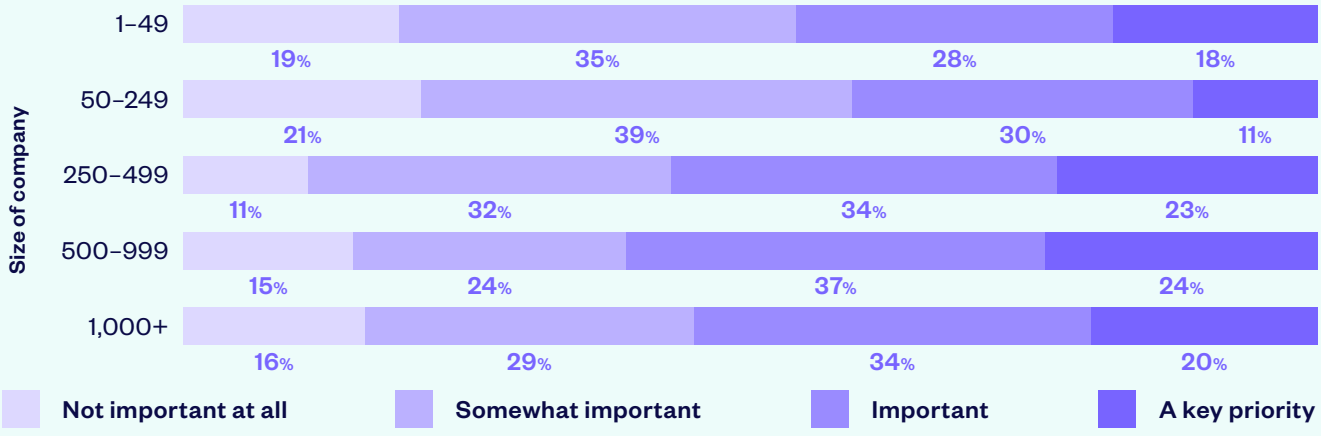
Leadership buy-in

While feature management is clearly a vital tool for engineering, product, and UX teams, does senior leadership see the value? While only 19% think it is seen as vital by senior leadership, 83% see it as at least somewhat important, with just 17% of senior leadership teams still to be convinced.



Q: How important is feature management and experimentation to business leadership in your org?

This trend gets worse at smaller organizations, where leadership teams appear to have bigger fish to fry than feature management and experimentation. At companies with less than 250 people, it is not important to 20% of respondents' leadership teams and only a key priority for 15%.



Q: How important is feature management and experimentation to business leadership in your org?

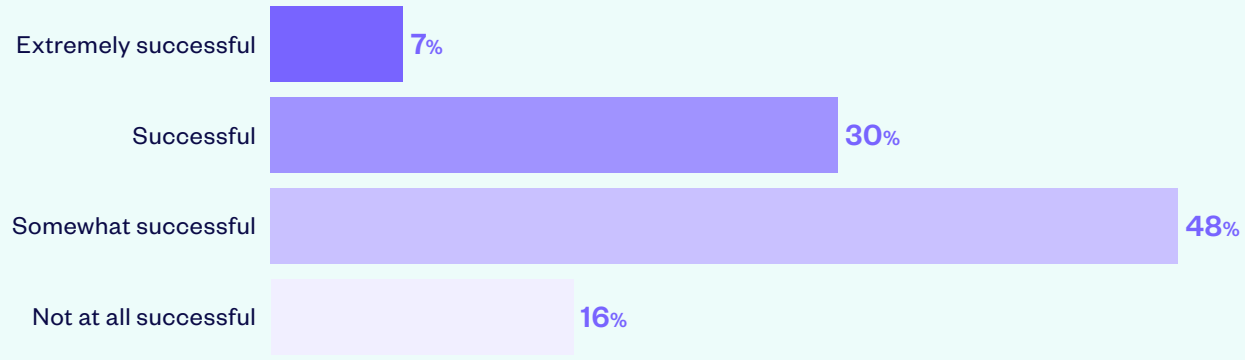
5. How are we doing?

Summary

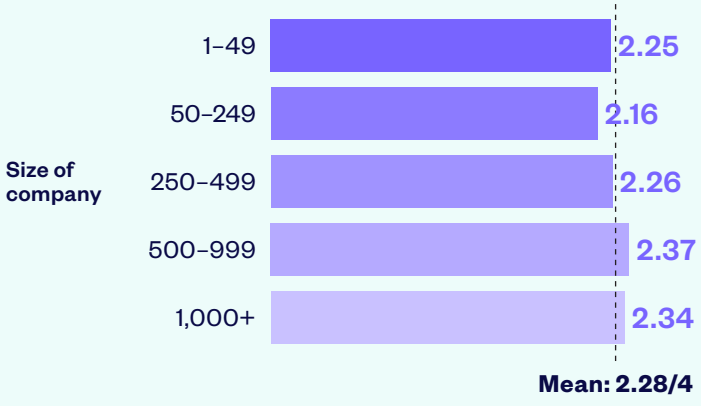
- **84%** of organizations feel like they have been at least somewhat successful with their feature management and experimentation efforts to date.
- But only **7%** feel they have been "extremely successful."
- Organizations that have fully committed to managing and experimenting with features typically feel like it was worthwhile.

The good news is that of those that have tried feature management, **84% of organizations feel like they have been at least somewhat successful** to date. Just 16% report having no success at all.

However, of that 84%, only 7% feel they have been "extremely successful," suggesting there is plenty of room for improvement for many organizations yet, as best practice, skills, and tools in this space continue to mature.

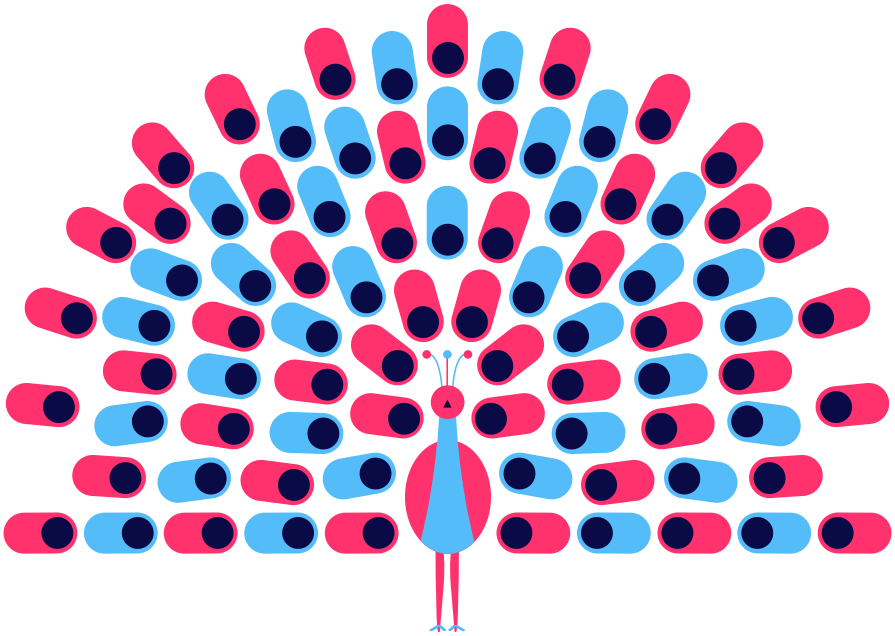


Mean success – where 1 is not at all successful and 4 is extremely successful

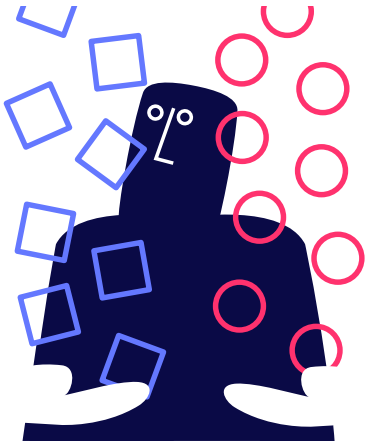


Q: In your opinion, how successful is your org at feature management and experimentation?

This suggests a clear maturity curve when it comes to managing features and experiments frequently compared to dabbling with ad hoc experiments. For organizations that have made experimentation core to their ways of working, only 1% think they aren't doing well, and 17% feel like they are extremely successful. **Put another way: Once an organization fully commits to these ways of working, they are more likely to feel like it was a success.**



Final thoughts



It's clear from this research that feature management and experimentation is still a maturing concept, with far too many organizations dabbling at the edges and coming away feeling unsatisfied with the results.

As with any software development practice, there is a clear maturity curve at play. A handful of organizations feel like they have cracked the code and a great deal more have either had some success, or are just starting to see sustained results from their efforts to manage and experiment with features in a robust andprofessional way.

What is clear is that organizations that have put feature management and experimentation at the core of their software development processes feel they are continuously improving their products, which is why we tinker in the first place, right?

Engineer Without Fear

If you're not embracing feature management & experimentation already, you're falling behind. If you're doing it the hard way, let us remove the toil so you can move faster with more confidence.

Feature Management & Experimentation

Release up to 50x faster

Triage issues in less than 1 minute

Boldly experiment without risk

Learn more at harness.io/fme

split by harness®