The state of feature management

2021

LaunchDarkly

How teams are using feature management.

Delivering software fast and frequently is a competitive advantage.

And yet, even in the era of Cloud and DevOps, the risk, complexity, and fear of launching new features slows innovation and engineering efforts.

Effective feature management has emerged as an antidote to some of these challenges, igniting developer productivity and creativity by fundamentally changing how software is delivered to customers.

We partnered with Wakefield Research, an independent research company, to learn how feature management is changing software development.

We surveyed

453

software professionals who currently use a dedicated, commercially-available platform for feature management.

We also compiled usage data from our user community to highlight general use patterns regarding feature releases. Here's what we found.

Including

203

LaunchDarkly customers

250

who use other feature management solutions

Feature flags are an industry standard.

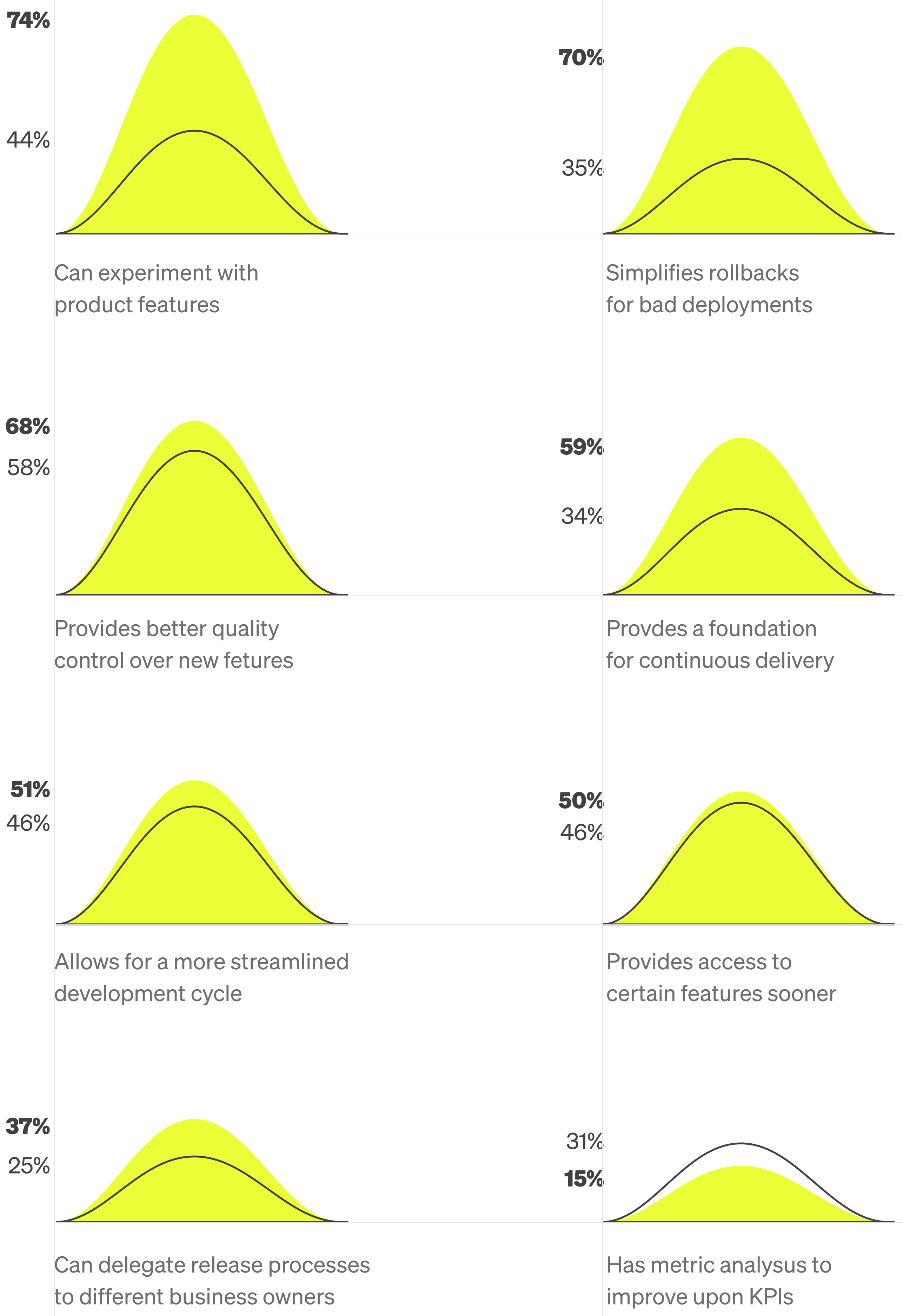
One of the biggest misconceptions about feature flags is that they're a new technology. In fact, feature flagging has been around since at least 2010.

In our survey, the majority of respondents say they've been flagging using a variety of methods before choosing a commercially-available platform like LaunchDarkly. While the reasons vary for why they're currently using flags, the majority of respondents say they're now flagging on almost every project.

Strongest initial motivations for using feature flags

LaunchDarkly customers are overwhelmingly more motivated to use feature flags to experiment with product features, simplify rollbacks, and pursue continuous delivery.

It’s notable that customers choose LaunchDarkly more for every initial motivation except for using metrics to hit KPIs. Our customers see us as a tool that can solve many different kinds of problems.



Can experiment with product features

Simplifies rollbacks for bad deployments

Provides better quality control over new fetures

Provdes a foundation for continuous delivery

Allows for a more streamlined development cycle

Provides access to certain features sooner

Can delegate release processes to different business owners

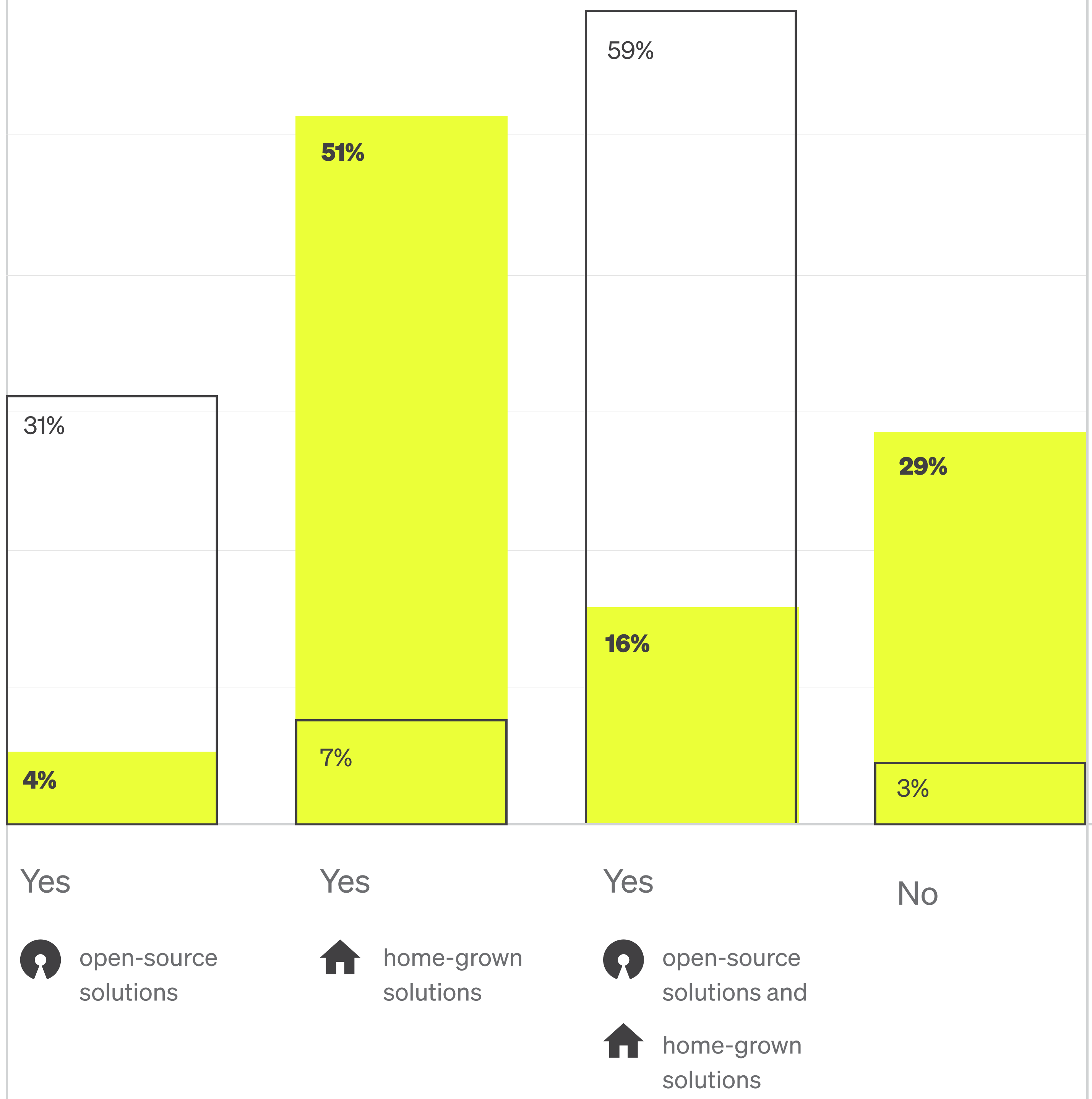
Has metric analysus to improve upon KPIs

🟨 **Customers** □ Non-Customers

Tools used prior to investing in a commercial feature management service

The majority of LaunchDarkly customers were using home-grown systems prior to choosing our platform.

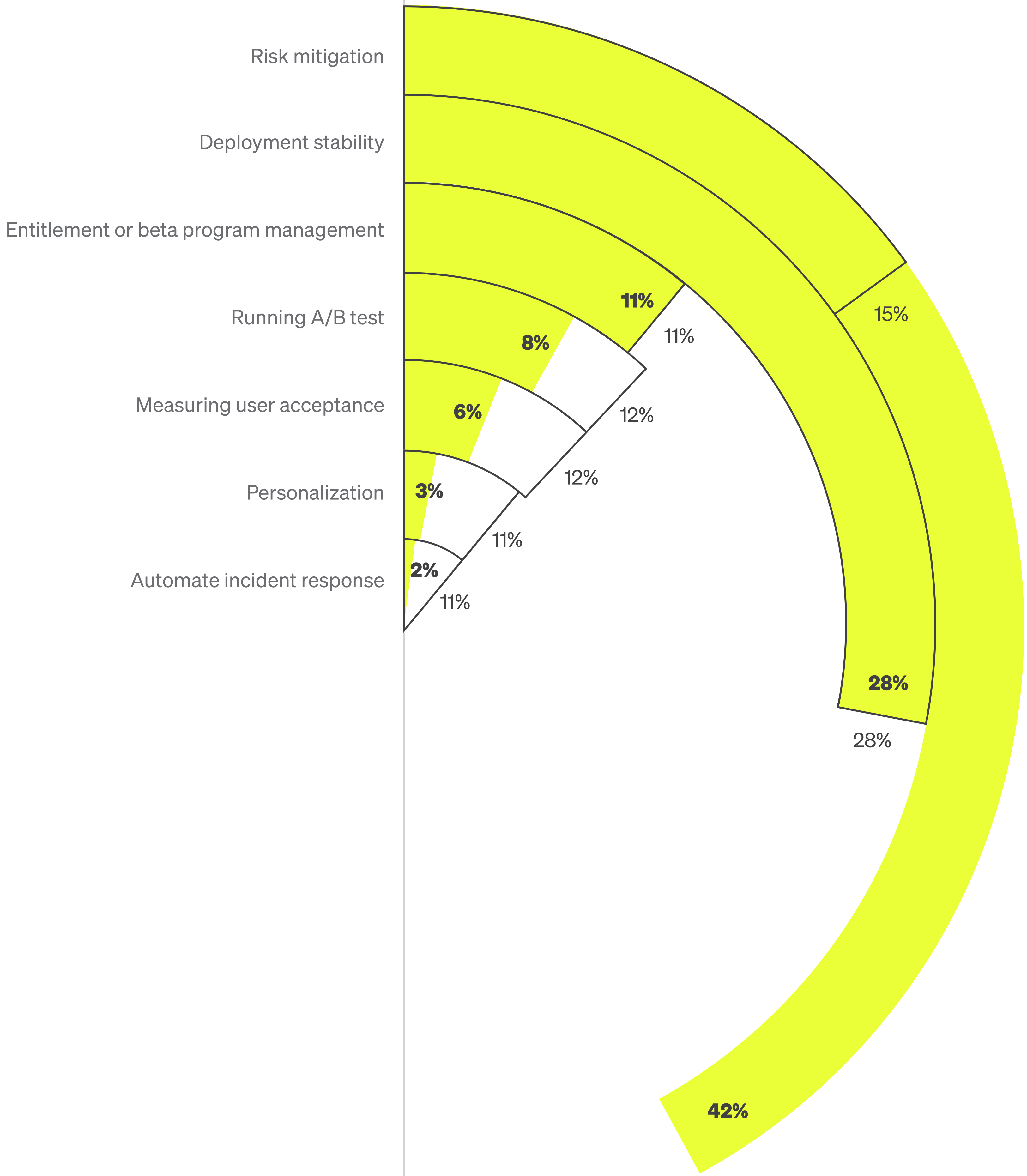
This shows that the ability to manage features has been important for a while, even before commercial solutions were widely available. Many organizations have also used open-source feature flagging solutions, but we find that those work best for single-language systems instead of polyglots.



🟨 **Customers** □ Non-Customers

Top reasons for feature flagging

Risk mitigation is the top reason LaunchDarkly customers are using feature flags, while deployment stability remains the main priority for non-users.



LaunchDarkly customers appear to trust us heavily for de-risking, and the distribution of other uses may increase as we add education around other use cases. In the future, we'll investigate which risks people find specifically concerning, because we think it's an area that warrants more research.

🟨 **Customers** □ Non-Customers

How often feature flags are being used

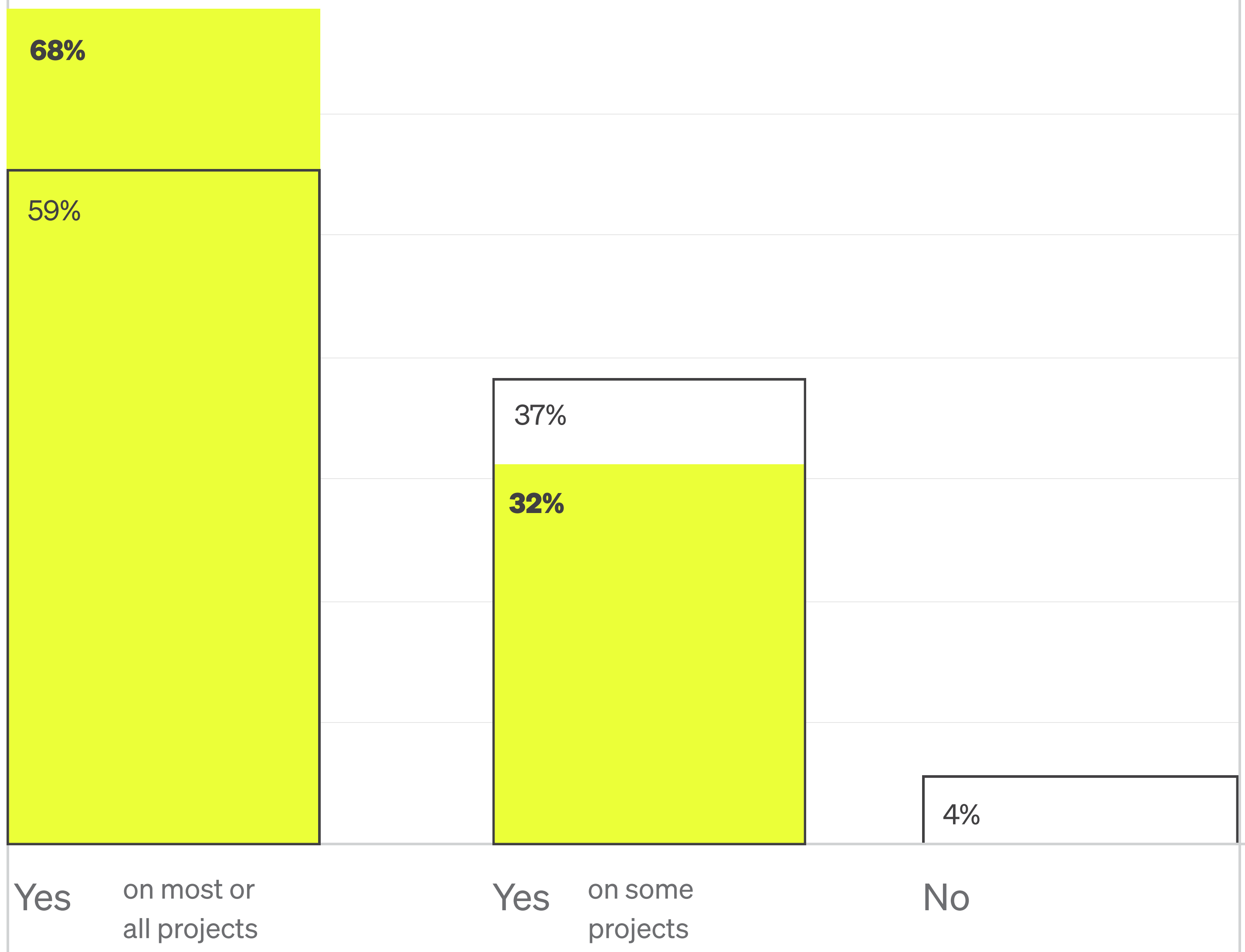
Most respondents are relying heavily on feature flags for almost every project.

68%

of LaunchDarkly customers use feature flags on most or all projects

/ / as well as 59% of non-customers

This means that feature flags are a part of the default workflows for creating features and adding business value. As feature management becomes more culturally widespread in a company, use cases that match user needs are typically added to the feature management system.



🟨 **Customers** ▭ Non-Customers

Key takeaways

Feature flags are an industry standard.

Feature management is not a new technology. Until viable commercial options became available, teams were using homegrown or open-source solutions.

Homegrown and open-source tools are a stepping-stone to feature flagging culture, but organizational growth and complexity demand more robust and accessible tooling.

Feature flags are a critical part of the majority of our respondents' projects. Feature flagging culture is a tried and tested software pattern.

Feature management is improving people's lives.

Our survey respondents show that feature management, when done correctly, is actively making their team's performance and quality of life better.

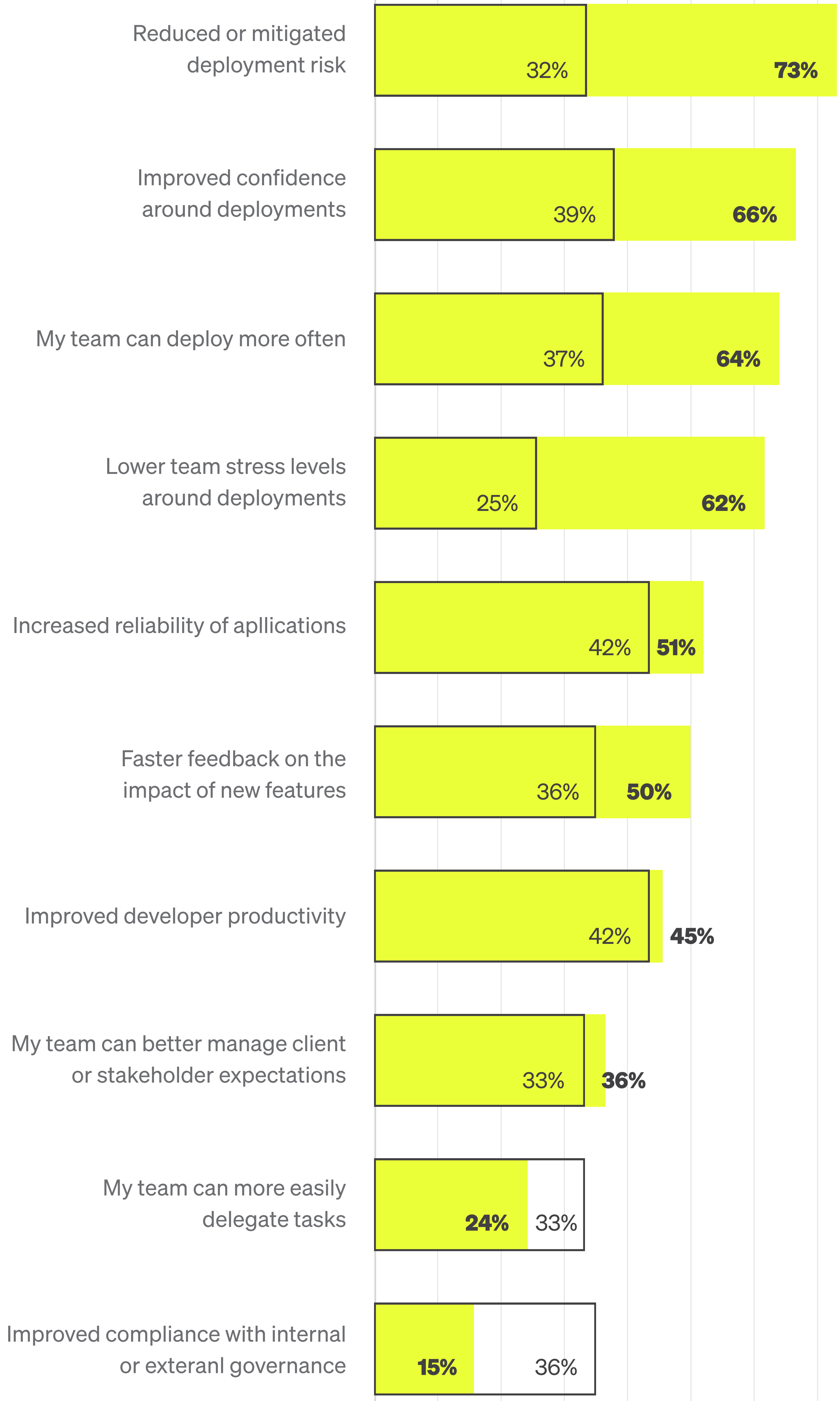
Without such a platform in place, the potential consequences range from system outages to lost customers and productivity.

Easing deployment fears

While reduced risk stays top-of-mind for LaunchDarkly customers, everyone seems to agree that the impact of quality feature management can be felt teamwide.

We know from the DORA report that a team's ability to deliver value is affected by their comfort with deployment and risk. Easing deployment fears frees up team energy to focus on improving use and experience.

Feature management also allows teams to hand off the crucial moments of testing and launch to the appropriate business teams. Changing the decision point to the team closer to the user reduces the time for feedback to make a difference.

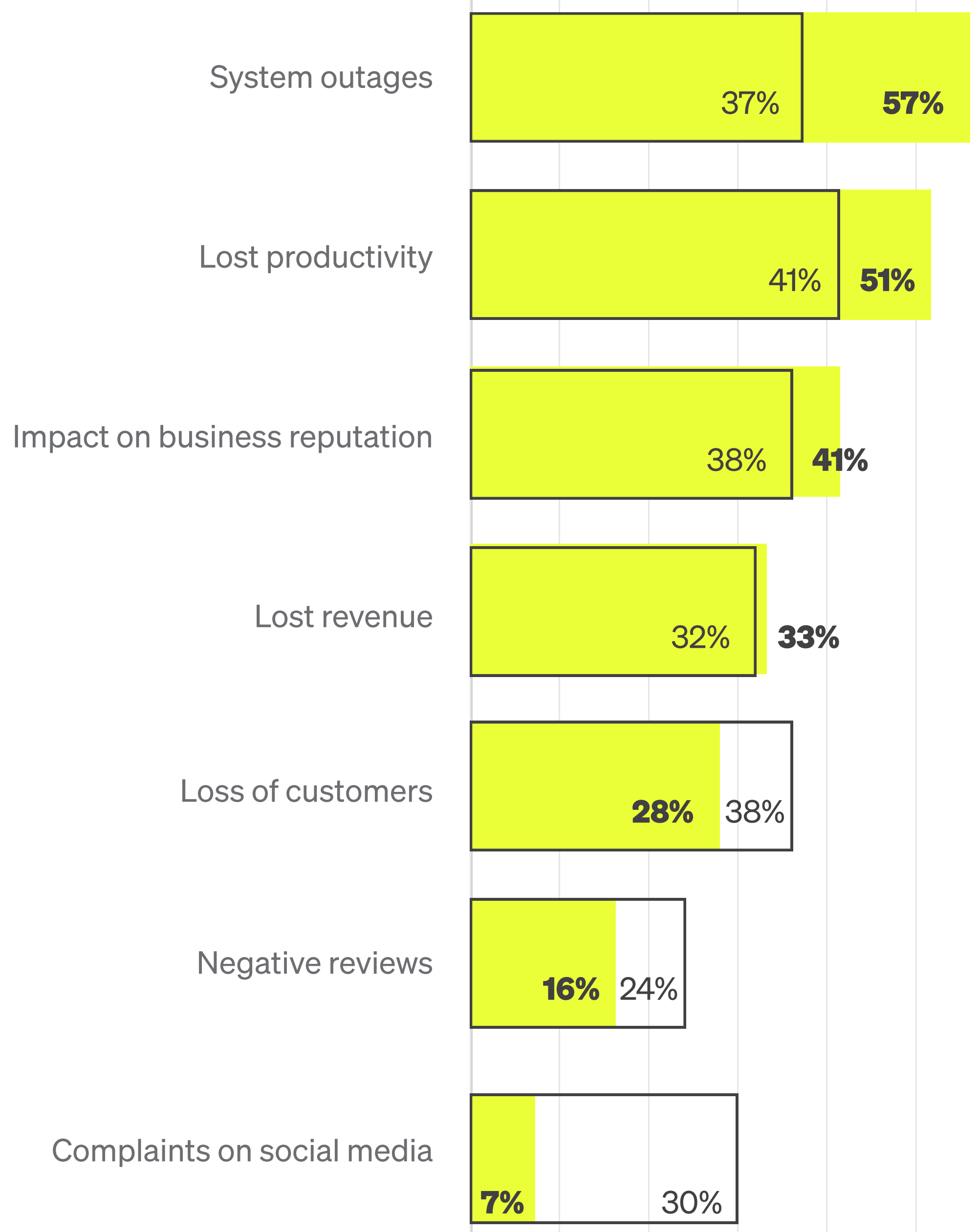


🟨 **Customers**

□ Non-Customers

Feature management failure scenarios

Everyone who deploys and releases has a story about a time when it went wrong, when the release broke user experience or affected the business's reputation.



🟨 **Customers** □ Non-Customers

Both customers and non-customers share a range of concerns around what would happen if they did not have an adequate way to handle feature management.

When we asked our respondents what they worry most about, we can see that LaunchDarkly customers are more concerned about system outages, which might include systems that are beyond the control of feature flags.

One of the interesting results from this question was that LaunchDarkly customers are significantly less worried about social media complaints and negative reviews than their peers. This may be because they have experienced how quickly feature flags can restore a system's state if something does go wrong, or it may just be that the LaunchDarkly respondents are working further away from the user experience in general.

Key takeaways

Feature management is improving people's lives.

➤ LaunchDarkly customers overwhelmingly feel that a good feature management platform reduces risk, improves confidence, and results in more deployments.

⚠ The consequences of not handling feature management correctly are real and potentially severe.

☺ Organizations have seen the positive impacts of effective feature management and aren't going back.

Everyone wins when better software is delivered faster and safer.

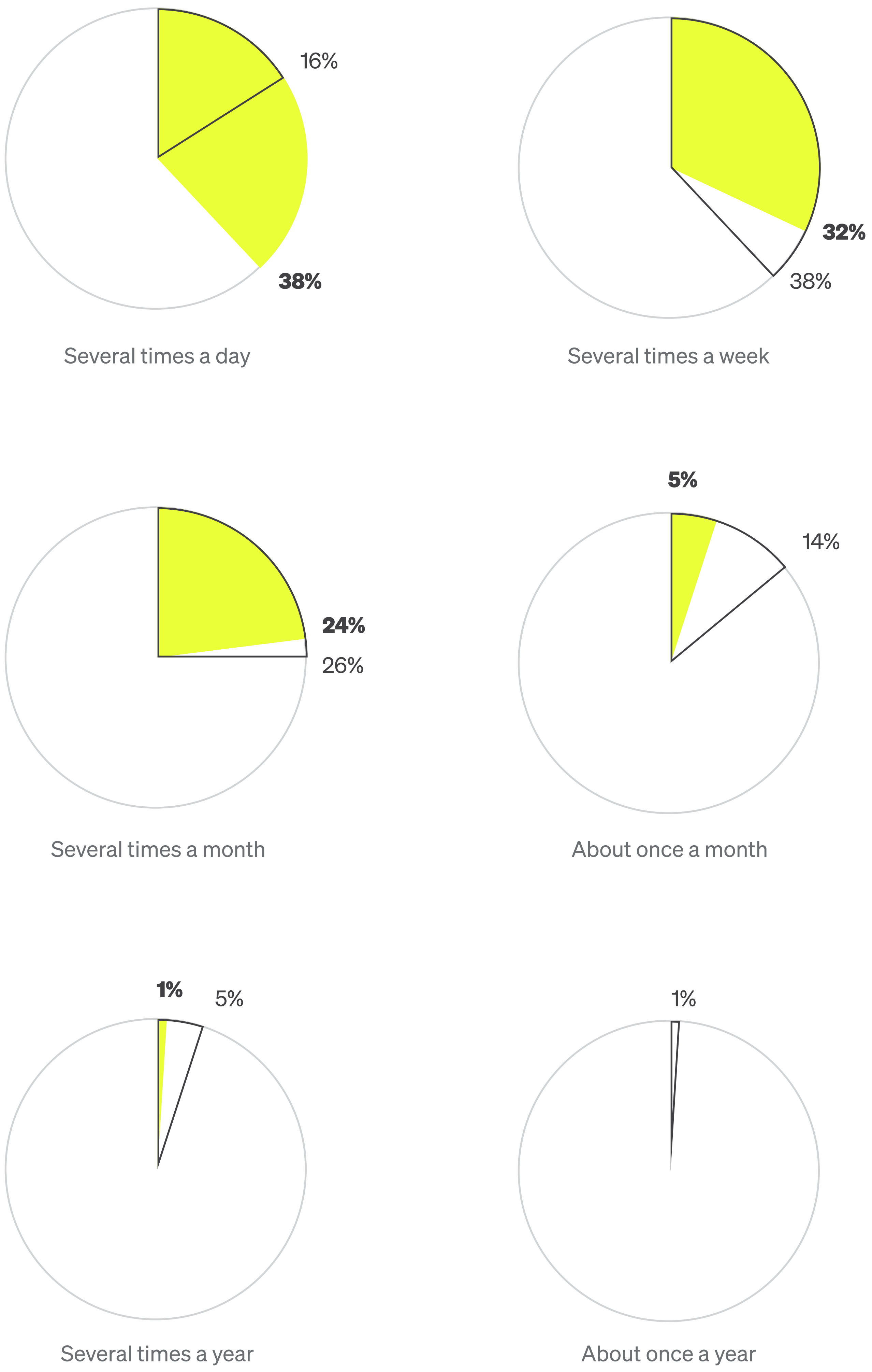
In this section, we'll look at how quickly teams are delivering software, their most common types of deployments, frequency of rollbacks, and more.

Diving into deployment rates

70%

of LaunchDarkly customers say they are releasing **several times a week or more.**

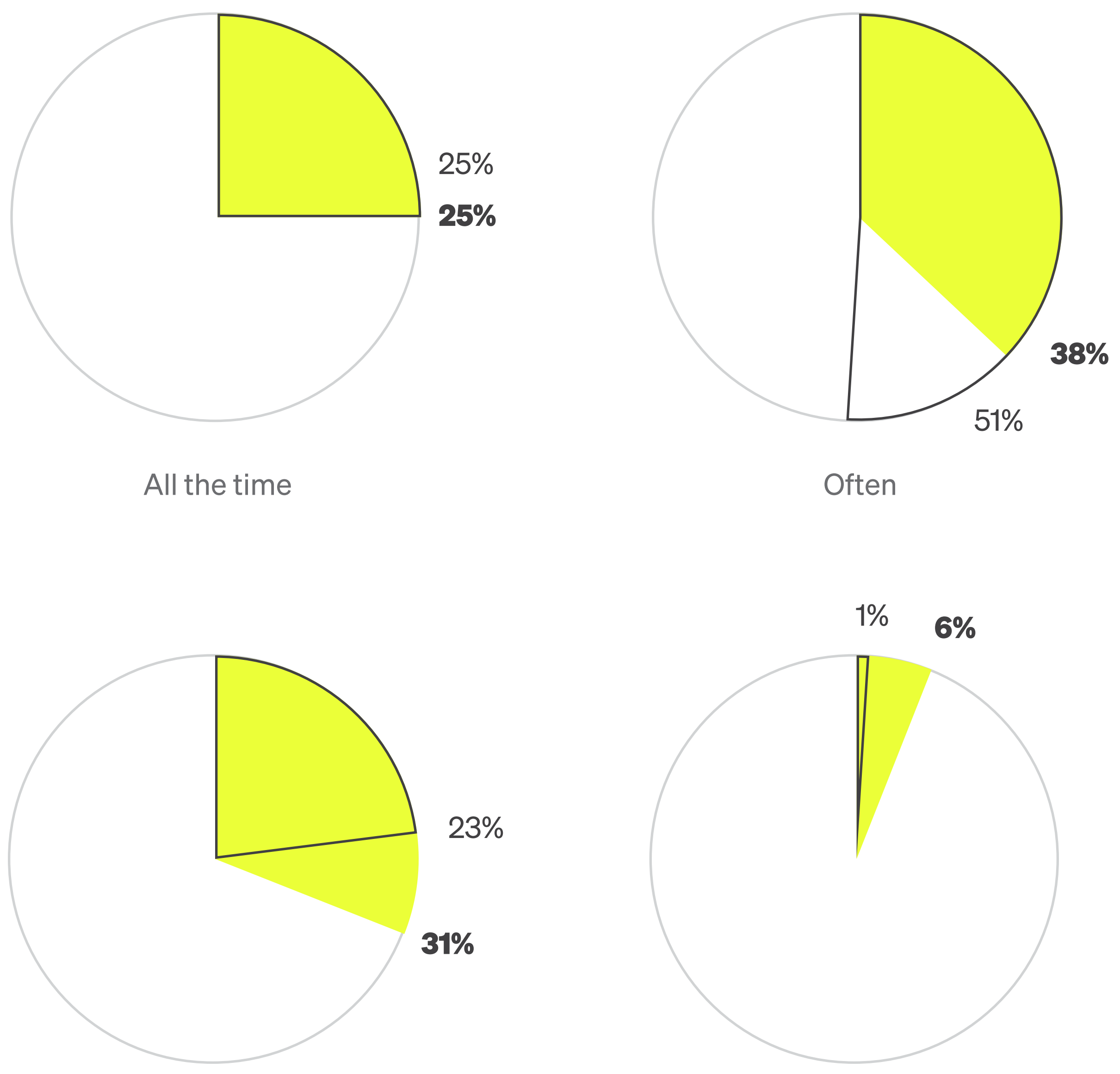
/ / compared to 55% of non-customers



🟨 **Customers** □ Non-Customers

Controlling features following a deployment

Pulling a feature from production once it's been deployed is a popular tactic with both sets of respondents.



🟨 **Customers** □ Non-Customers

After using LaunchDarkly feature flags, our customers achieved:

Source

9x ↑

increase in deployment frequency

76% ↓

decrease in time from commit to deploy

27% ↓

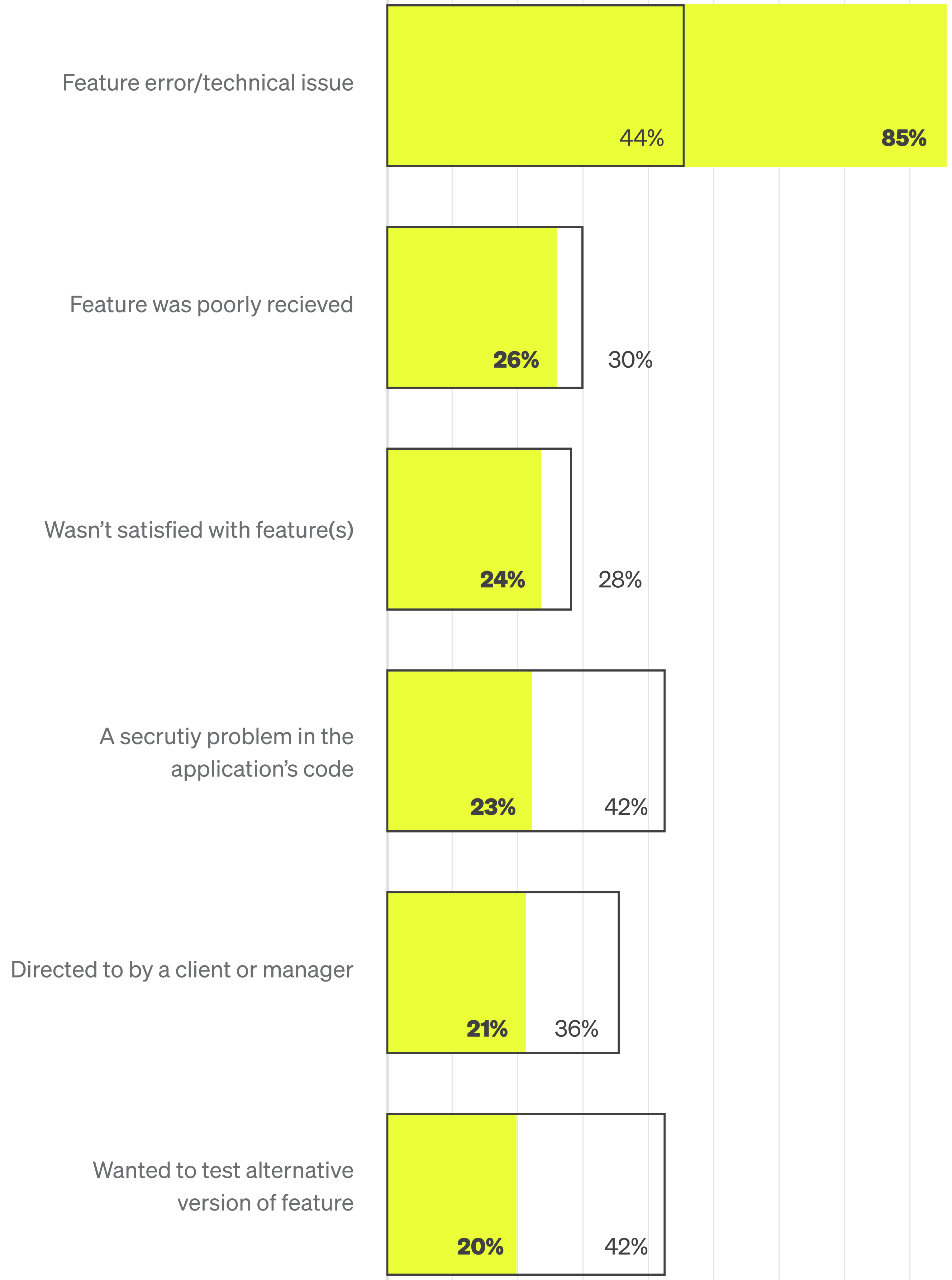
decrease in time to release new features

The most common reasons for performing rollbacks

85%

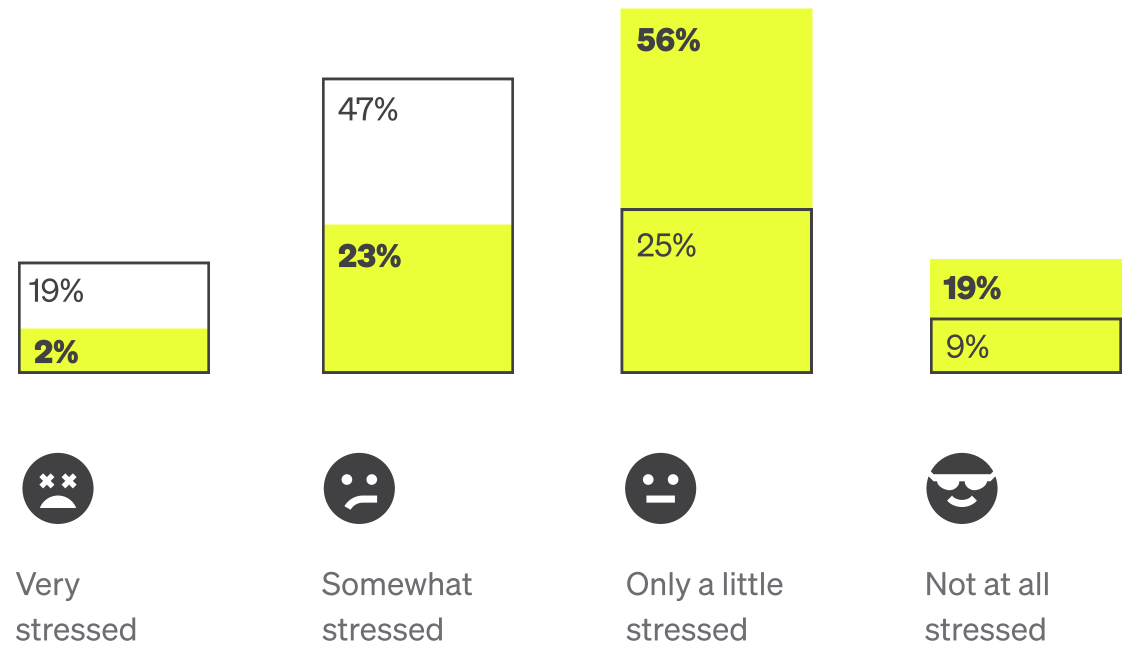
of LaunchDarkly customers say when rollbacks occur, they are due to an error or technical issue

/ / non-users are more evenly split between technical errors, security problems, and testing alternative versions



🟨 **Customers** □ Non-Customers

Managing stress during deployments



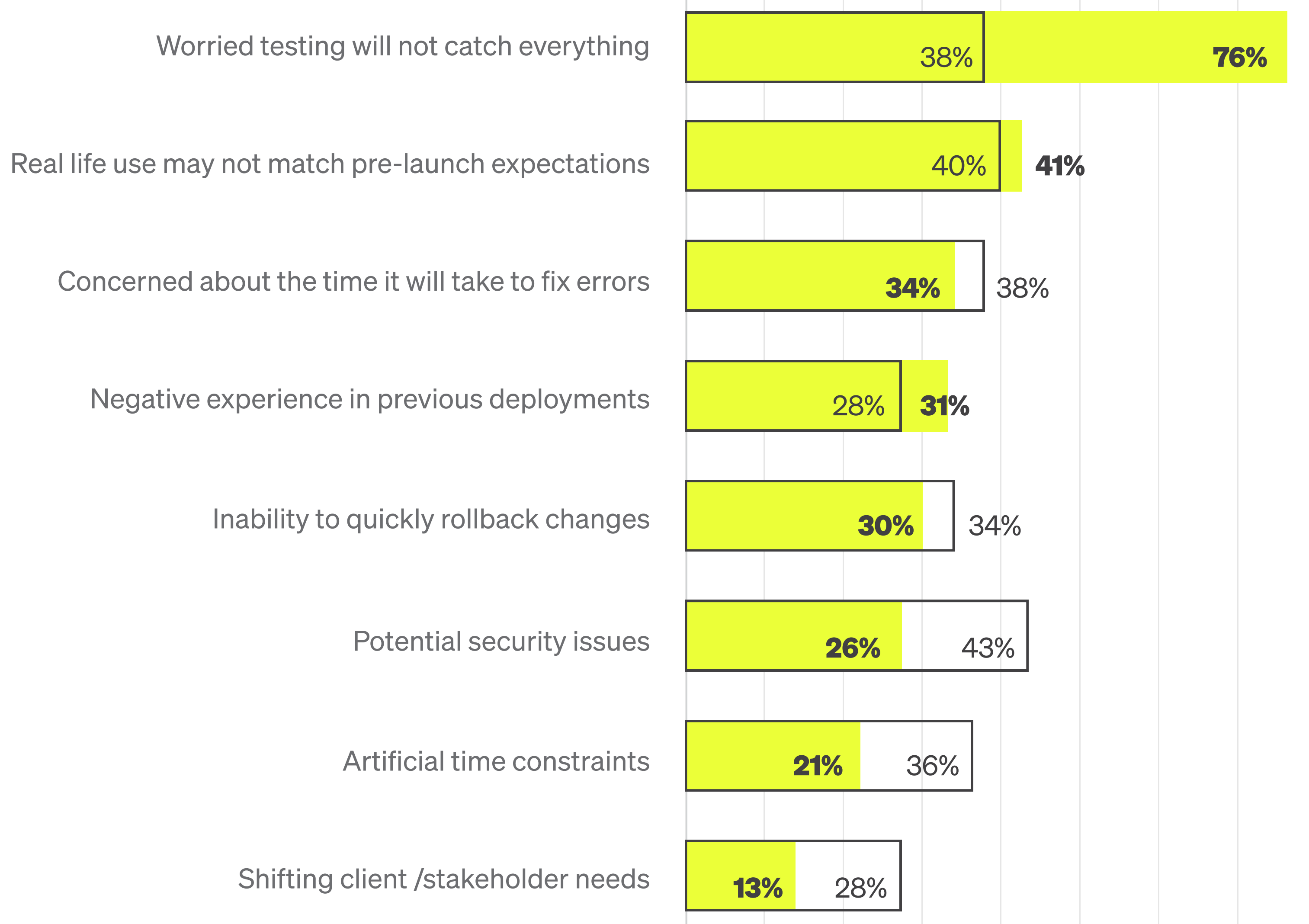
Only

2%

of LaunchDarkly users are very stressed

// compared to 19% of nonusers

Causes of your stress



LaunchDarkly users experience stress over QA, not bugs, when releasing new features. Non-customers, meanwhile, say their biggest concern is security, followed by the launch not meeting expectations.

🟨 **Customers** □ Non-Customers

Key takeaways

Everyone wins when software is delivered faster and safer.

- The majority of all respondents are releasing multiple times a week.
- Only 2% of LaunchDarkly customers report feeling “very stressed” about releases, even though almost everyone performs rollbacks.
- There’s no single cause of deployment stress; everyone is feeling it for different reasons.

How companies are using LaunchDarkly.

We have more than 20 clients in the Fortune 100, and we serve thousands of organizations from five-person startups to multi-faceted enterprises.

In the next two sections, we'll review some of the findings we learned from working with our user community. Although this information—the majority of which was compiled separately from our survey findings—is not a perfect reflection of the economy as a whole, we think it does indicate our ability to support organizations at many sizes and maturity levels.

Industries that create the most feature flags

It’s not surprising that web-first companies have adopted feature management as part of their workflows.

The very basis of web delivery means that companies have a lot of options around how people use their products. However, we also see a lot of more traditional industries using feature management, either as part of their web presence or their value delivery. For example, feature flags can power the behavior of rural betting machines, car dashboards, streaming media, food delivery, and logistics routing.

Internet software & services	147,590
Software	76,842
Media & Internet	50,025
Finance	24,273
Retail	19,697
Business services	15,400
Internet	11,746
Transportation	10,705
Insurance	10,180

Most popular customer integrations

LaunchDarkly is increasingly part of a seamless control system. By offering accessible endpoints and integrations, we can help customers build feature management into their existing daily systems.

Now users don't have to leave their context to request or approve flag changes, and LaunchDarkly serves as a force multiplier for software they already use, even if they aren't aware that their actions are invoking a feature flag. Slack is certainly something of a superpower in the current distributed workforce, and we are continuing to increase the utility and usability of the integration for users at all technical levels.

Slack	39.8%
Jira	14.5%
Generic Webhook	10.2%
GitHub	9%
Datadog	8.4%
Code references	5.2%
Microsoft teams	4.6%
Custom trigger	2.2%
New Relic APM	2%

Other: 4.1%

Top software development toolkits (SDKs)

One of the first questions developers and operators ask about a tool is whether it works with their existing system.

This internal data about what SDKs are called most is a useful snapshot of where people are integrating LaunchDarkly flags. Browser SDKs are an obvious choice for web-based delivery, but we also see the .Net, Java, and Javascript/NodeJS ecosystems participating meaningfully. Overall, there are 26 actively-updated SDKs supporting languages as uncommon as Erlang, Lua, and Haskell.

Browser	32.4%
NET	17%
Java	12.3%
Node.js	10.3%
Python	5.3%
iOS	5.3%
Android	4.7%
Go	4%
Ruby	3.9%
PHP	2%

Other: 2.8%

Key takeaways

How companies are using LaunchDarkly

- Internet-first companies are feature flagging the most.
- The use of our integrations shows a trend toward asynchronous and/or automated feature management.
- SDK usage indicates that we add a lot of value to web features, but .Net, Java, and NodeJS are also significant parts of the SDK ecosystem.

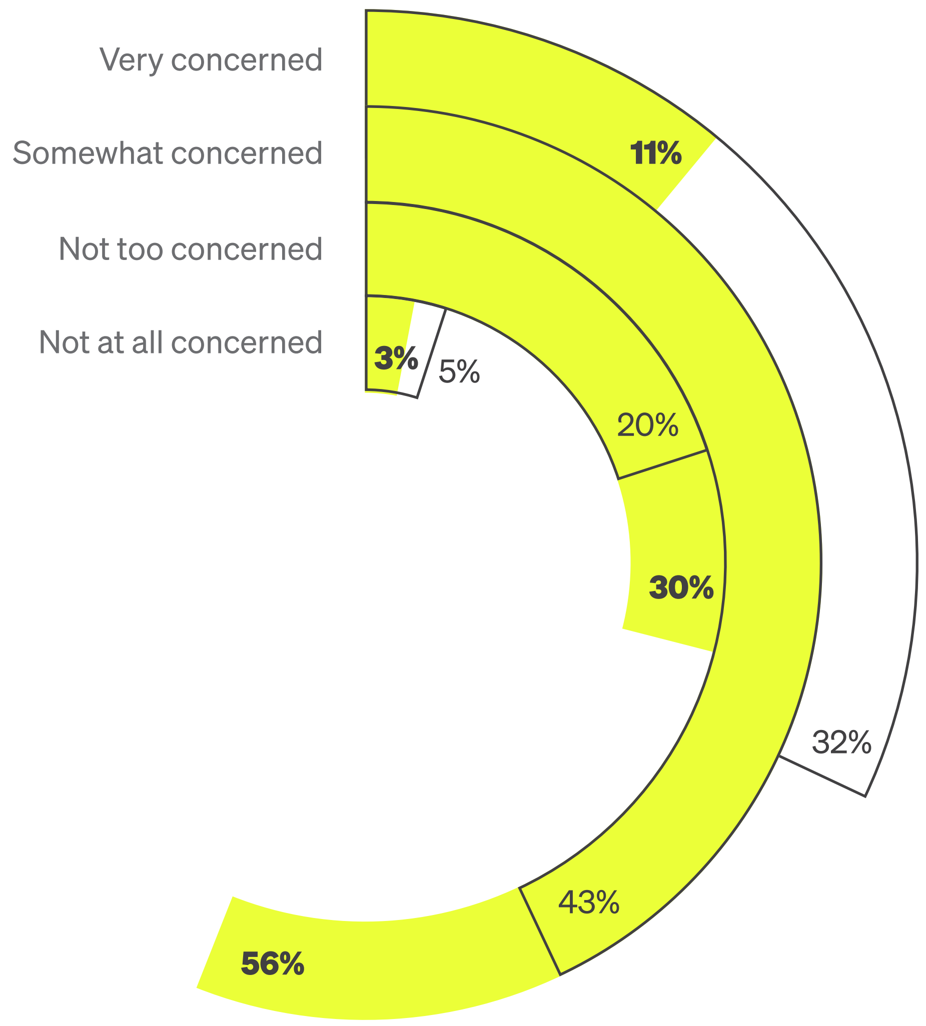
Technical debt is a threat for everyone.

Technical debt is inevitable.

If left ignored, however, your once seemingly manageable technical debt can grow into an obstructing behemoth, increasing development costs, delaying innovation, and elevating security risks—all factors that negatively impact revenue growth. In this section, we'll look at how our user community is managing technical debt, as well as how our survey respondents feel about it.

Levels of concern with technical debt

In our survey, everyone is concerned about technical debt, but the degree to which they're worried varies.



■ Customers □ Non-Customers

The responses we saw indicate that everyone who uses feature management spends time worrying about the technical debt that they are carrying, but LaunchDarkly customers aren't "very concerned".

We know from The Forrester New Wave™: Feature Management And Experimentation, Q2 2021 report that LaunchDarkly is the feature management system that scores highest in handling technical debt.

> LaunchDarkly is a Leader in the feature management space, offering highly performant and scalable feature flagging capabilities. Setting it apart is its strong enterprise governance capabilities, which include an integrated workflow and approval engine. It also scored highest in managing feature flag technical debt.

The Forrester New Wave™ : Feature Management And Experimentation, Q2 2021

Best practices can mitigate some of the technical debt implications of feature flagging, but every organization needs to plan for how they will stay aware of this new dimension of code management.

Flag types and technical debt impact

Experimental flags

// Temporary flags that are associated with a LaunchDarkly experiment. These might control A/B/N tests or other experimental states. They should be removed when the experiment has concluded.

Archived flags

// Flags that are no longer being evaluated, but can be retained, either to keep data that are associated with them or to keep targeting rules that may be useful to copy and reuse.

Permanent flag

// A flag that is not meant to be removed or tied to a particular release. Permanent flags are also called operational flags or long-lived flags. Organizations might use permanent flags for managing entitlements, configuring circuit-breaker patterns, or managing traffic load. Because these flags are designed to remain with the software long-term, they are not considered technical debt.

Temporary flag

// A flag intended to be removed once release is completed. Temporary flags are often used for progressive delivery, gradual rollouts, A/B/N testing, or canary launches. Temporary flags should be removed once a release is fully operational and stable. Leaving temporary flags in your codebase introduces risk, since they are not meant to be activated again.

Total active flags

Permanent Flags

5,672,924

Temporary Flags

244,598

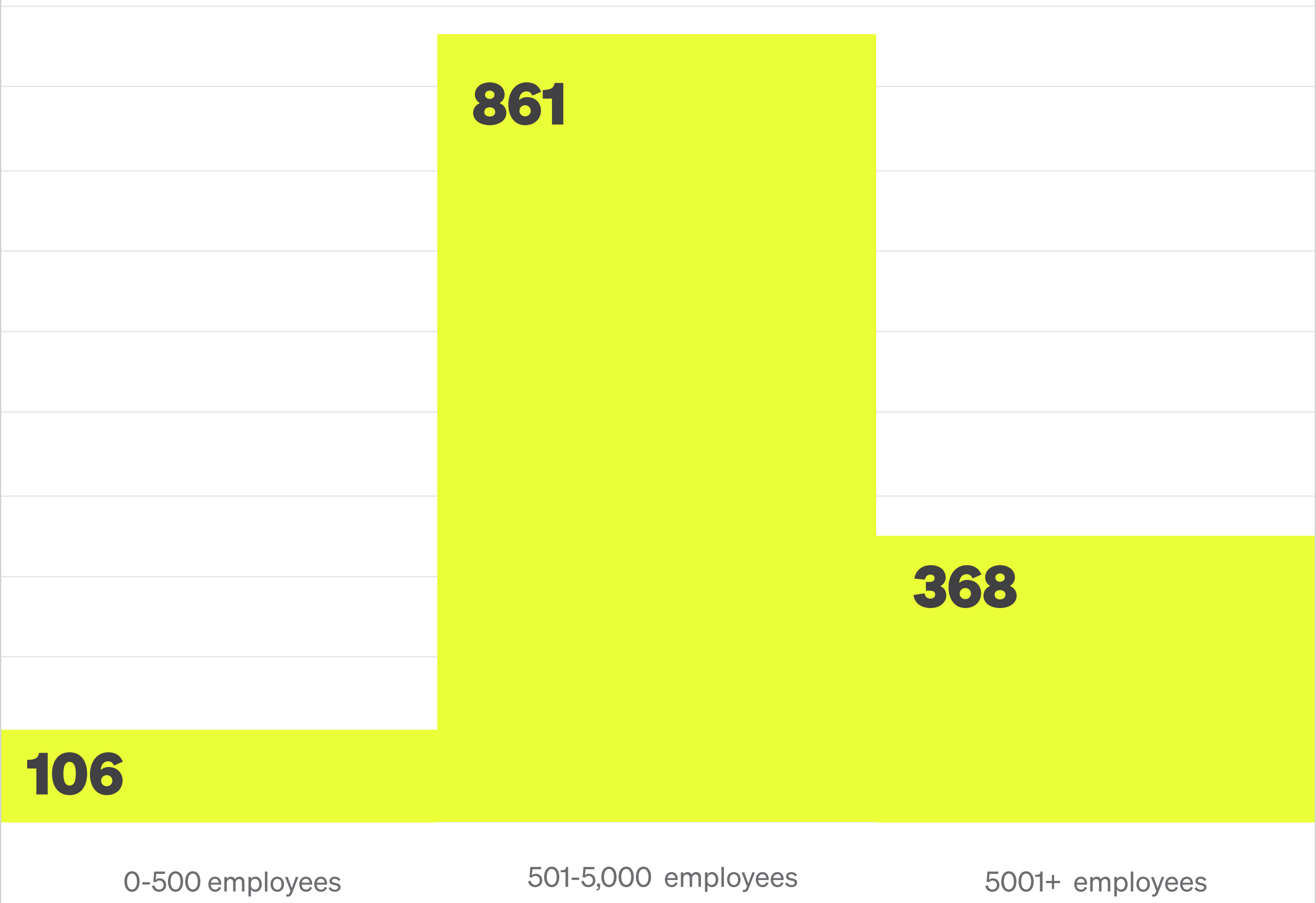
Experiment & Temporary Flags

58,042

Experimental Flags

80,924

Average active flags by business size



A huge majority of the active flags we see when analyzing user systems are permanent flags. Most organizations use permanent flags for operational or business logic purposes. It's not surprising that there are so many flags in this category, since a lot of LaunchDarkly adoption has come through DevOps groups.

The next largest category are the temporary and experimental flags. Temporary flags manage releases and help de-risk the act of turning software on. Experimental flags are tied to an experiment, allowing users to do hypothesis-based development with real-time results and testing. Finally, we have archived flags, which are flags that have been removed from active service but are retained to keep track of audit history, serve as templates for future flags, or just as insurance in case the feature needs to be flagged again.

Key takeaways

Technical debt is a threat for everyone.

⚠ Technical debt is on everyone's minds.

Organizations are using permanent/long-lived flags 10x more often than temporary or experimental flags, which may indicate operations or automation use cases.

Medium-sized organizations tend to have more flags, but they don't last as long, so they may be at a different level of flag use maturity.

The wrap up: unexpected observations.

While compiling this report, some unexpected things jumped out at us.

This data demonstrates how feature flags are currently being used in the software development lifecycle. We also wanted to give teams the data points of the larger community to compare their own experiences against.

For comparative purposes, we noted there are...

5.7 million

permanent flags to roughly

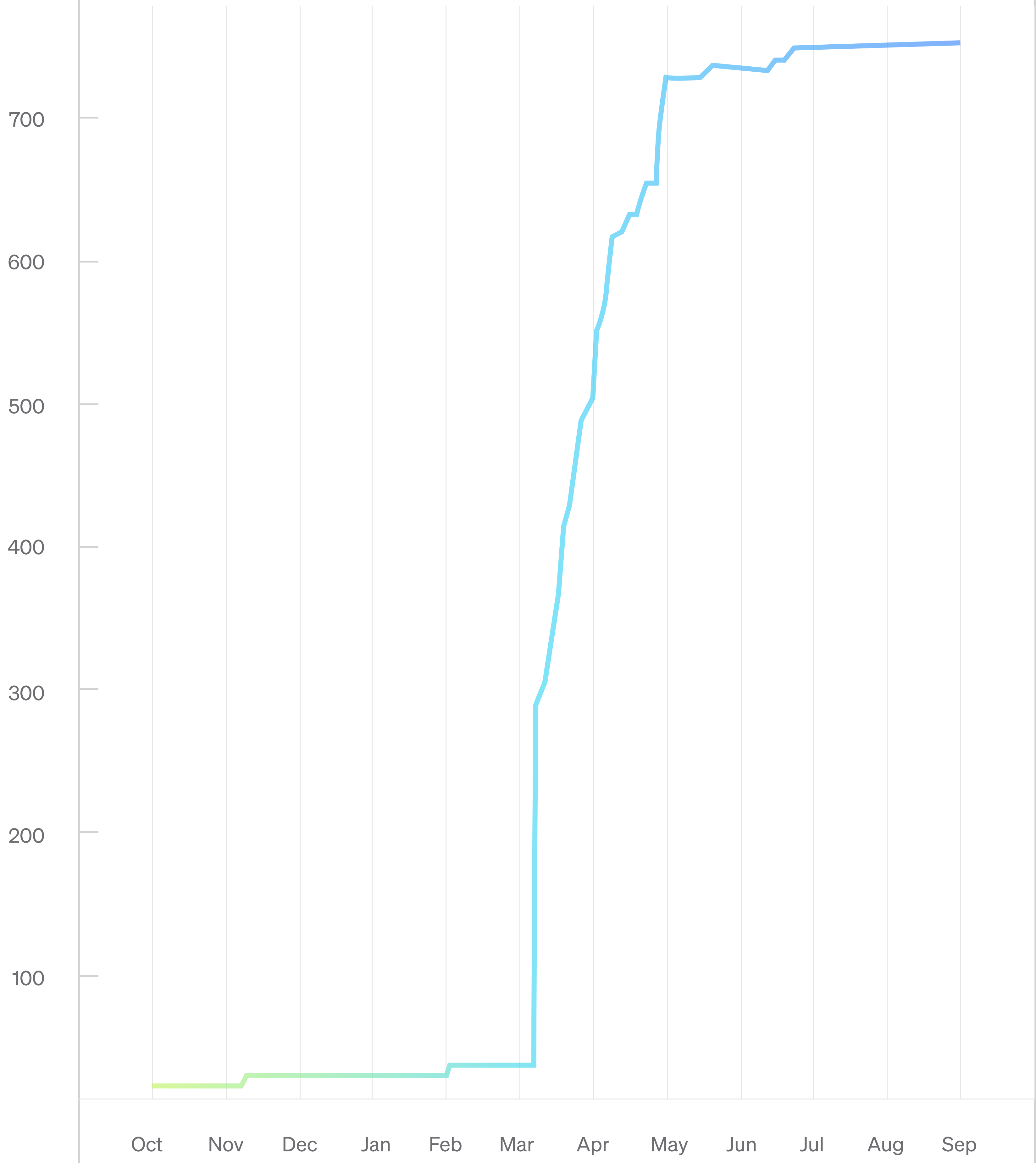
244,598

temporary flags.

We think this may relate to a lot of flags being used to manage entitlements or infrastructure needs, such as circuit-breakers and kill switches.

After we made a change for automated workflows in March 2020, we saw a significant uptick in automated code changes. Increasing the ease of automation allows teams to split responsibility and delegate flag activation to non-development teams. As our ServiceNow, Apex, and Salesforce integrations join our Slack integration, we expect to see this number increase even more.

Flags triggered by account integration with action - turnFlagOn (total executions)



When looking at all survey participants, we see that...

40%

of LaunchDarkly customers have been using feature flags for **3-5 years**

This is a testament to the value our customers have found in managing their features with flags. The data also indicates that our users are generally more familiar with feature flags as a concept, and as a part of their overall software development lifecycle.

Another interesting finding was that...

75%

of LaunchDarkly customers are only a little stressed or not stressed at all when deploying new code

Of course we want every deployment to be as free of stress as possible, and we'll continue to work to ensure that's the case. Reducing deployment size, being able to release in increments, and having the ability to roll back a deployment instantly significantly reduce the anxiety many people feel about deployment and release.

Thank you.

Thanks again to everyone who participated. 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.

We empower all teams to deliver and control their software.