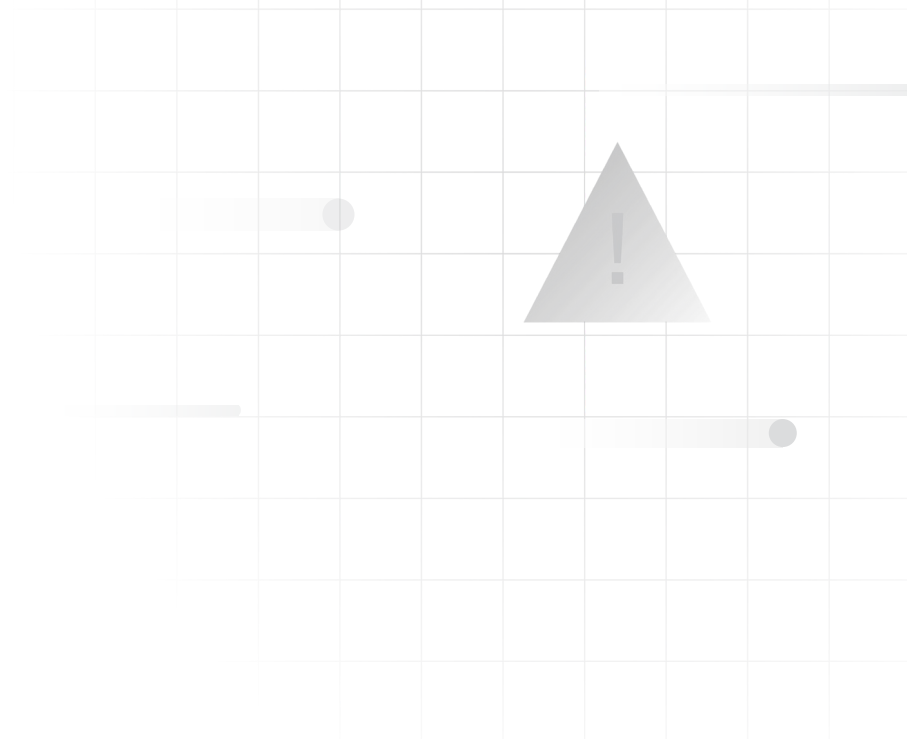




LaunchDarkly →

Release Assurance

Why innovative software delivery starts
with trust and psychological safety



Contents

Introduction	02
<hr/>	
Leadership nurtures innovation	04
<hr/>	
The processes fueling and fumbling innovation	11
<hr/>	
The impact of failing to support the psychological safety of developers	16
<hr/>	
Strengthening the last mile of software delivery	20



Release Assurance

Why innovative software delivery starts with trust and psychological safety

“Innovation is the ability to see change as an opportunity, not a threat,” said Steve Jobs. Without getting into Jobs’ leadership style, if you’ve ever worked on a software team where you dread your manager discovering that you deviated from standard release processes, you know it’s not always that simple.

Creating something truly innovative, by design, means doing something different. To get outside the status quo, you’ve got to take a risk. And if your company is too risk averse, it’s going to be very tough to do any of these things.

That’s one of the reasons trust and psychological safety, especially when practiced at the team and company levels, are so powerful: your team gains the ability to be fearless in attempting new features and processes.

Psychological safety is a broad concept that has implications for multiple aspects of an organization’s culture. However, as pressures increase to deliver more software updates faster, it’s clear our software development efforts need safety. That extra boost in confidence not only benefits developers and entire organizations, but also those on the receiving end of software updates.

Through this report—built around responses from 500 software developers across a range of industries and job titles—we explore:

- Factors that limit risk taking in the era of DevOps and continuous delivery
- How a fear of failure when releasing software dramatically impacts various areas of business, including retention; 67% of developers say they have either quit a job due to pressures around minimizing mistakes or know someone who has
- The three practices companies can leverage to remedy a fearful culture and give their teams some safer headspaces in the process

Ultimately, we all want better software for ourselves and others. And we're not going to get there without some room for error, intentional or otherwise.

What is psychological safety?

There are varying ways to describe psychological safety, but we like this definition from Amy C. Edmondson, Novartis Professor of Leadership and Management at the Harvard Business School and one of the pioneers on the topic:

“I define psychological safety as a belief that one will not be punished or humiliated for speaking up with ideas, questions, concerns or mistakes, and that the team is safe for inter-personal risk-taking.”



Leadership nurtures innovation

It's difficult to be your most creative at an organization where the people overseeing things can't spot the value in innovation. In our findings, the overwhelming majority of developers (84%) say improving their outcomes in terms of both velocity and quality is a "large priority," at the very least, for their organization. But it's the type of support given to developers by those in charge that translates into psychological safety and makes the difference.

The traditional view of leadership—c-suite, executives, and upper-level management, in particular—might involve setting budgets, measuring outcomes, and distilling priorities. For those looking to release better software, they should start thinking about whether or not they're building a culture of trust within their organization that lends itself to psychological safety.

From our findings, there's direct relationships between teams that are higher performing and feel entrusted to take risks, and leadership's actions to ensure those things are true.

Bosses can't just tell a team to be innovative, then go back to checking Twitter and hope for the best. Virtually all our respondents (99%) say they are looking to feel safe when taking risks in deployment, especially around internal buy-in, tools, and culture (94%).

How leadership's priorities impact software releases

We'll dive deeper into causes and effects later, but some points worth highlighting:

- When leadership places the improvement of developer outcomes as either a top or large priority, 91% of developers say they are very satisfied with their jobs. So if leadership is perceived as placing an emphasis on making developers more successful, that seems to positively impact areas like retention.
- When leadership places the improvement of developer outcomes as either a top or large priority, 67% of developers say there's a much stronger focus on releasing updates more quickly than avoiding rollbacks. In other words, teams that feel like their companies have their back are releasing more frequently.
- When developers are encouraged by leadership to take risks, the majority (54%) say either all or most of their releases are automatically approved (as opposed to manually approved, which can slow things down.) Organizations that assess the risk of automatic approvals to be less risky than the delays introduced by manual approvals signal that they trust developers, tools, and processes to take the right steps to manage risks appropriately.
- 86% of developers at companies that say they are more innovative than others agree that safely releasing code makes them feel empowered to experiment with new approaches. The momentum of an organization encouraging you to take risks can pay off through experimentation and, gradually, innovation.

In the next section, we'll look at the ways in which a company's overall priorities and approach, which, in many cases, are set directly by leadership, reflect back on the outcomes of software releases.

A focus on innovation starts with leadership championing developers

How innovative is your company compared to other companies?

Much more innovative	37%
Somewhat more innovative	48%
Neither more nor less innovative	11%
Somewhat less innovative	3%
Much less innovative	1%

Priority of improving developer outcomes for software releases:

	Top/large priority	Less than a large priority
Much more innovative	40%	17%
Somewhat more innovative	48%	49%
Neither more nor less innovative	9%	21%
Somewhat less innovative	2%	11%
Much less innovative	0%	3%

At 85%, the overwhelming majority of respondents feel their company is more innovative than others. That may speak to the competitive nature of the software industry, but those working at high-performing organizations seem to know it: 87% of teams releasing several times a week or more said their organization was more innovative than others.

Of those who say their company is more innovative than others, 88% say that improving developer outcomes for software releases is either a top or large priority at their organization. Those findings support the idea that innovation is driven partially by how high a value the organization, driven by leadership, places upon the support of its developers.

Most companies have some appetite for risk

Generally speaking, how much risk tolerance does your company have for unexpectedly negative outcomes in software releases?

A lot of risk	11%
Some risk	49%
A little risk	36%
No risk	4%

The majority of our respondents say their companies (60%) are game for either a lot or at least some risk, which is good because you can't have innovation without risk. And you can't take risks if you're feeling too worried about the outcome.

If you're working in a regulated industry, such as finance or government, it's understandable that your company would be more wary than others when it comes to something breaking. Among the 40% of those at organizations that can only stomach a little or virtually no risk, the split was almost even with those in regulated industries (42%) and unregulated (39%). Although, when it comes to those who answered that their company has a high tolerance for risk, those in non-regulated industries (12%) slightly edged out those (7%) in more regulated industries. The takeaway then is that

even within companies where managing risk is likely top of mind, developers still perceive an overall willingness to take on risk in today's dynamic, digital-first business environment.

Even if organizations want risk, the fear is still felt by developers

Which of the following best reflects your company leadership's view on deployments?

Developers are encouraged to take high levels of risks even if it results in rollbacks	11%
Developers are encouraged to take medium levels of risks even if it results in rollbacks	37%
Developers are encouraged to take low levels of risks even if it results in rollbacks	35%
Developers are pressured to work carefully to minimize the need for rollbacks	17%

In the previous section, we saw that most organizations are open to risk, but the concern around the consequences of errors is felt most by developers.

In fact, the majority of respondents (52%) say they are encouraged to either take low levels of risk or are pressured to work to minimize rollbacks altogether. The hypothesis, then, is that the actual process of releasing updates to customers naturally produces higher levels of organizational stress and urgency for developers to minimize errors.

Of the 35% of developers who say their company encourages low levels of risk, you can't help but feel there are some missed opportunities. Fully 100% of developers say applying new development approaches and solutions can positively impact business outcomes, especially by encouraging greater innovation among staff (53%), increasing adaptability (52%), and improving the bottom line (49%).

And yet it's difficult to imagine a team taking high levels of risk without feeling psychologically safe and having the support of the rest of the organization.. For those who say they are encouraged to take high levels of risk, a whopping 94% said their organization views improving developer outcomes as a large or top priority.

One other thing: 6% of senior executives said improving developer outcomes for software releases was only a small priority. If you're a developer at one of those companies, maybe it's time to start looking around for another job.

Process will help your innovation soar or stumble

How much of an obstacle to innovation are your company's processes?

A significant obstacle	15%
Somewhat of an obstacle	46%
A slight obstacle	33%
Not an obstacle at all	7%

The majority of respondents (61%) feel their company's processes hinder their ability to innovate and try new things. What's interesting is that even at companies where developers say improving outcomes is a top or large priority, 61% also describe their process as either significant or somewhat of an obstacle. That number, then, holds steady even if a company views developer outcomes as a priority.

So your organization can simultaneously be fighting to improve software releases and somewhat tolerant of risk, while also inadvertently making things harder for developers with its overall processes.

It's worth noting, too, that 45% of teams that are releasing more often—either several times a day or week—say their process is not really an obstacle to innovation.

We'll dig into some of the issues with processes in our next section.



The processes fueling and fumbling innovation

Every team has a different way of doing things. Various tools, team members, approvals, priorities, hacks, and more all play a role in how a process unfolds. That diversity of modes of operation is one of the most fascinating aspects of software development, but we see some broad trends that you should note if your team is striving for greater innovation.

For starters, releasing new code is a fact of life for developers: 85% deploy new code to production several times a month or more, and more than half (55%) do so at least several times a week. But many of these deployments come with red tape.

A near majority (49%) say releasing code changes to production involves manual approvals most or all of the time. Sometimes manual approvals can be a necessary step, but excessively heavyweight approval processes can also slow down velocity and encourage low-risk policies—ultimately hindering psychological safety as a result.

In this section, we'll dive into some of the way processes can help to make or break your innovation.

Weekly or more is the new standard for release cadence

How often does your team release new code changes to production?

Several times a day	12%
Several times a week	43%
Several times a month	30%
About once a month	9%
Several times a year	6%
Once a year or less	0%

Big bang releases are on their way out. Today's top-performing teams are releasing around the clock. In our study, 55% of developers say they're releasing new code to production either multiple times per day or week. It's not to say that you have to constantly be pushing out code, but if you're not at least moving in that direction, you can guarantee that a competitor is doing exactly that.

Almost all companies are focused on holding the line between the speed and quality of releases

How much of a company priority is improving developer outcomes for software releases with regards to balancing velocity and quality?

The top priority	36%
A large priority	48%
A moderate priority	12%
A small priority	3%
Not a priority at all	-

While companies are releasing more often, they're not looking to do it by sacrificing the overall quality of the product. The overwhelming majority (84%) of respondents say the balance between quick and safe is crucial. This is great news, because nobody wants quantity over quality when it comes to new software updates. Reaching some sort of equilibrium is the goal.

With so much focus on speed, developers are feeling the stress

Which puts more pressure on you as a developer?

Deploying updates more quickly	65%
Avoiding rollbacks for deployments	35%

Most developers are feeling the pressure of delivering updates faster than ever, but only around a third feel avoiding rollbacks is

actually a bigger stressor. The concern here is that organizations are becoming almost too focused on velocity, and potentially losing focus on quality.

Manual approvals are still common, despite increased push for automation

Which of the following best describes the approvals needed to release a new code change to production for you or your team?

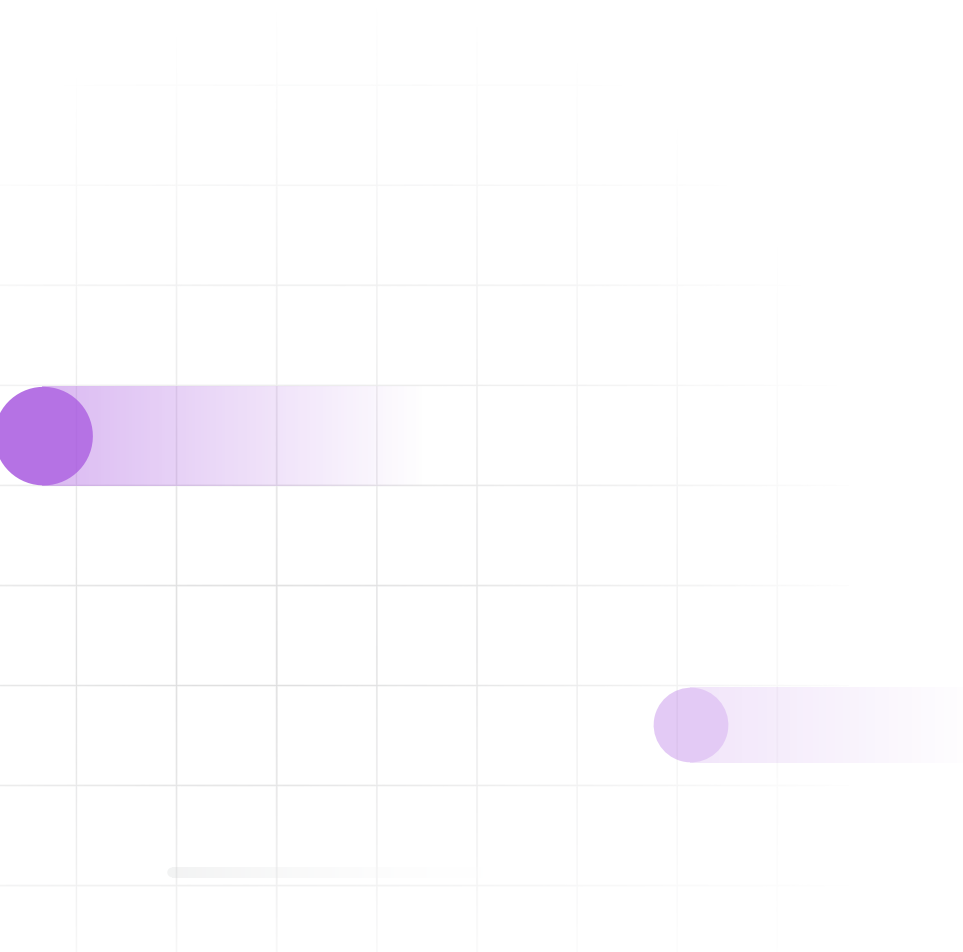
All changes require manual approval	12%
Most changes require a manual approval, other lower risk changes are automatically approved	37%
Most changes are automatically approved, only some high-risk changes require manual approval	40%
All changes are automatically approved, no manual approvals required	10%
Other	-

The split is fairly evenly distributed between those requiring manual approvals and those who opt for a lighter, more automatically-approved process. However, 89% of developers say releasing code changes to production involves some kind of manual approval, including 49% who say it's required most or all of the time.

In spite of research such as the [DevOps Research and Assessment \(DORA\)](#) and from [analysts](#) suggesting more automated approvals—which equates with more trust in developers—we still see that for a large portion of teams, manual approvals are common.

And, interestingly, the majority (63%) of those working at companies with fewer manual approvals say their processes are still either a significant or somewhat of an obstacle to innovation.

Organizations should consider the emergence of new techniques that can help alleviate the overuse of manual approvals including: better risk assessment and scoring through analytics, automated methods to mitigate risk areas such as automated testing and kill switches, and release techniques such as canary, blue/green, and progressive rollouts.





The impact of failing to support the psychological safety of developers

We see a strong correlation between leadership championing the success of developers, and those teams actually feeling that support.

To be clear, most developers still seem relatively happy with their jobs. We do, however, see some red flags in teams that aren't feeling supported when it comes to innovation and improving outcomes.

Too much focus on avoiding risk introduces more risk

Overall, a majority of developers (53%) say leadership pressures them to minimize the need for rollbacks or take only low levels of risk—a pressure that comes from official policy. And 40% of developers say their company has little to no risk tolerance for unexpectedly negative outcomes in software releases (such as rollbacks or failed deployments).

What we also see is that too much focus on avoiding mistakes will cost you talent. The majority of developers (67%) say they or someone they know has quit over pressure to minimize deployment errors, including 36% who have quit themselves.

Change needs to be championed from the top down, and developers who feel supported by leadership to take risks often feel

more confident in their work. And those incremental risks can lead to larger, organizational success.

Virtually all developers say applying new development approaches and solutions can positively affect business outcomes, especially by encouraging greater innovation among staff (53%), increasing adaptability (52%), and even improving the bottom line (49%).

There's more job satisfaction at innovative organizations

How would you rate the general job satisfaction of developers in your organization?

Very satisfied	53%
Somewhat satisfied	39%
Somewhat dissatisfied	6%
Very dissatisfied	2%

There's been a lot of efforts to keep developers happy in recent years, and some of it appears to be working. Overall, 92% of developers say their teams are either very satisfied or somewhat satisfied with their current roles.

“Somewhat satisfied” doesn't mean those developers are extremely happy with the way things are going. It could be more akin to a shrug. And the way the company is perceived to be performing makes a difference.

For instance, 86% of devs who describe their company as prioritizing developer outcomes for software releases as either a top or high priority also say they are satisfied with their roles. Contrast that with just 14% of developers who say they are satisfied with their position even though outcomes are not a top priority for their organization. If you don't feel like leadership has your back, you're probably not as happy at your job.

People will quit over too much pressure to avoid risks

Have you or someone you worked with ever left a job due to pressures from over minimizing mistakes (i.e. avoiding rollbacks or failed deployments)?

Yes, I have	36%
Yes, someone I've worked with has	35%
No	33%

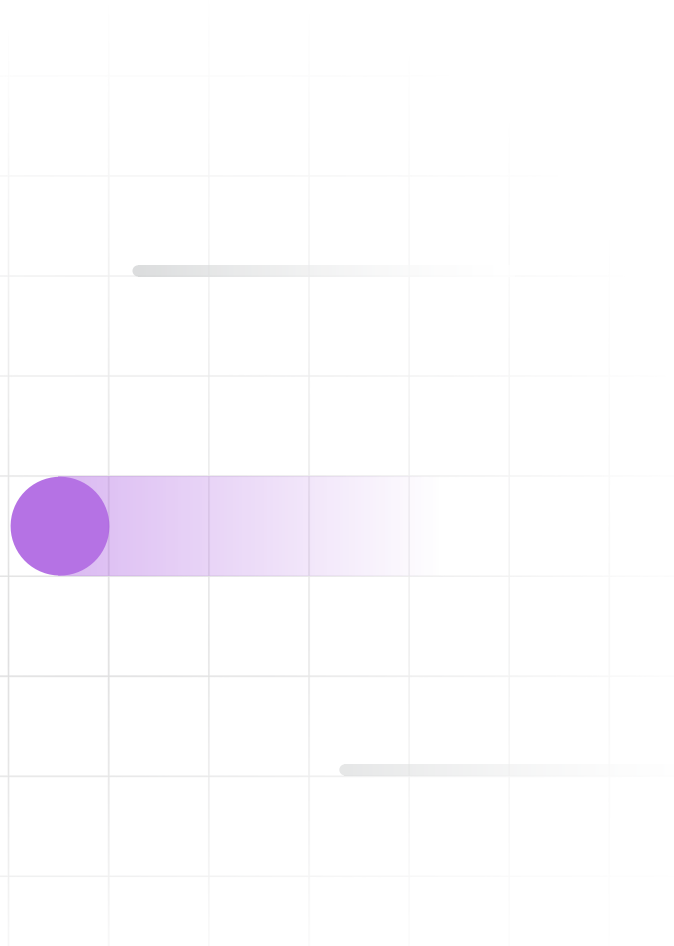
Note for chart above: Multi-choice question and reflects all respondents who answered "yes" to either or both "yes" categories.


Nobody wants to work in a culture of fear. A boss who is a micromanager can destroy your morale, as can overly burdensome processes.

The majority of developers in our study (67%) have either quit a job due to pressures from trying to avoid rollbacks or botched deployments, or know someone who has. To this point, there is some danger in being too careful, and that can impact retention.

Of those who say their process is at least somewhat of an obstacle to innovation, 74% say they've either left an organization or know someone who has due to pressure to avoid mistakes. So we see an over-reliance on a painstaking process can cause problems with retention over time.

And, in organizations where developer outcomes are considered less than a large priority, 78% have quit or know someone who has. That speaks to the fact that if management isn't trying to actively make the lives of developers better, attrition could follow.





Strengthening the last mile of software delivery

There are three essential practices that can help your organization increase confidence, risk-taking, and overall performance:

- Smaller, more frequent deployments that help to improve confidence
- Leaders fostering a culture of healthy risk-taking with psychological safety
- Developers feeling rewarded for experimenting

Adopting these practices can enable your teams to deliver more value to the user experience and company bottom line, while sparking innovation through different ways of doing things such as experimentation.

Overall, developers know that rollbacks and risks are part of their job, which is why they want deployment approaches that minimize consequences of rollbacks and disruptions while paradoxically moving more quickly than ever. Developers want psychological safety.

More than half (56%) would feel more confident in the deployment process with smaller, more frequent deployments rather than larger, less frequent deployments. This allows for less consequential rollbacks, as well as provides the space and safety for innovative approaches.

Psychological safety spells huge gains for retention and morale, especially around keeping companies on the cutting edge. Nearly all developers (93%) agree that when they gain confidence in their team's ability to safely release new code changes to production, they feel empowered to innovate more. The majority (52%) agree with this sentiment strongly.

Nearly half of developers (47%) say applying new development and solutions can lead to improved job satisfaction, and 45% say it improves user experiences.

More frequent, smaller deployments boost confidence

Which of the following would make you feel more confident in the deployment process

Smaller, more frequent deployments	56%
Larger, less frequent deployments	44%

Smaller releases are more popular with developers, and improve their confidence. To this point, 59% of developers releasing several times a week agree. Sixty percent of the developers in favor of smaller, more frequent deployments also say deploying updates faster puts additional pressure on them than avoiding rollbacks.

How do we square the fact that most developers think more frequent deployments gives them more confidence, but they also rank deploying faster as the main stressor? It could be that the size of the deployments those developers are being asked to release

are too large, or that they're getting pushed to release too fast. It's tough to say, but there is definitely a balance to be found there.

Another point worth mentioning is that around half of all developers who say their process is at least somewhat of an obstacle to innovation also say larger, less frequent deployments would make them feel more confident. That would make sense, because if your process is already convoluted and taxing, you wouldn't want to try and do it even more.

Developers want to experiment, and should be encouraged to do so

How important is it for your career to explore new development approaches and solutions?

Essential	30%
Very important	56%
Somewhat important	12%
Not too important	1%
Not important at all	-

86% of developers feel it's a major priority for their careers to feel free to try new ways of working. Experimenting helps foster innovation, and when there's clearly so much appetite for this, leadership at organizations should consider feeding it.

**How much do you agree or disagree with the following statement:
As I gain confidence in my team's ability to safely release new code
changes to production, I feel empowered to innovate more.**

Strongly agree	52%
Somewhat agree	41%
Somewhat disagree	5%
Strongly disagree	2%

Still, confidence is needed to take that next step into experimentation and innovation. According to 93% of developers, one of the ways they gain confidence is through their team's ability to safely release new code.

This seems accurate, because it would be difficult to do a job confidently while knowing the rest of your team isn't on the same level. It's up to leaders, then, to hire the right people and put the right processes in place that allows some sense of safety amongst the team to experiment and, hopefully, innovate in the process.

Experimentation fosters innovation and retention

How can applying new development approaches and solutions positively impact business outcomes?

Encourages greater innovation among staff	53%
Increases adaptability	52%
Improves the company bottom line	49%
Improves job satisfaction from developers	47%
Improves user experiences	45%
Other	-
Applying new development approaches and solutions cannot positively impact business outcomes	-

Note: This question was multi-response.

In today's job market, innovation isn't just resume filler: it's an essential differentiator. Virtually all developers (99%) say exploring new development approaches and solutions is at least somewhat important for their career development—including 30% who say it's essential. Another 56% say it's very important.

Experimenting and trying new ways of working in safe ways has positive ripple effects that touch almost all aspects of a business.

Wrapping up

Teams feeling empowered through the power of psychological safety to take new approaches, embrace risk, and become fearless will help drive innovation at your organization. And that, in turn, will result in more satisfied leaders, end users, and business results. There's clearly a disparity among organizations that are prioritizing successful outcomes for their developers versus those that aren't. And companies that place a high emphasis on improving software quality are viewed as more innovative. But it's difficult to be innovative when teams are terrified at stepping out of line or bogged down by outdated processes.

Nobody's saying it's an easy balancing act releasing quality software while staying hyper-competitive with the speed of releases. You'll get a lot farther, though, when teams are supported, encouraged, and trusted throughout the process.



Learn more at LaunchDarkly.com.