

# Developer experience



## Takeaways

Software doesn't build itself. Even when assisted by AI, people build software, and their experiences at work are a foundational component of successful organizations.

In this year's report, we again found that alignment between what developers build and what users need allows employees and organizations to thrive. Developers are more productive, less prone to experiencing burnout, and more likely to build high quality products when they build software with a user-centered mindset.

Ultimately, software is built for people, so it's the organization's responsibility to foster environments that help developers focus on building software that will improve the user experience. We also find that stable environments, where priorities are not constantly shifting, lead to small but meaningful increases in productivity and important, meaningful decreases in employee burnout.

Environmental factors have substantial consequences in the quality of the products developed, and the overall experience of developers whose job is to build those products.

## **Put the user first, and (almost) everything else falls into place**

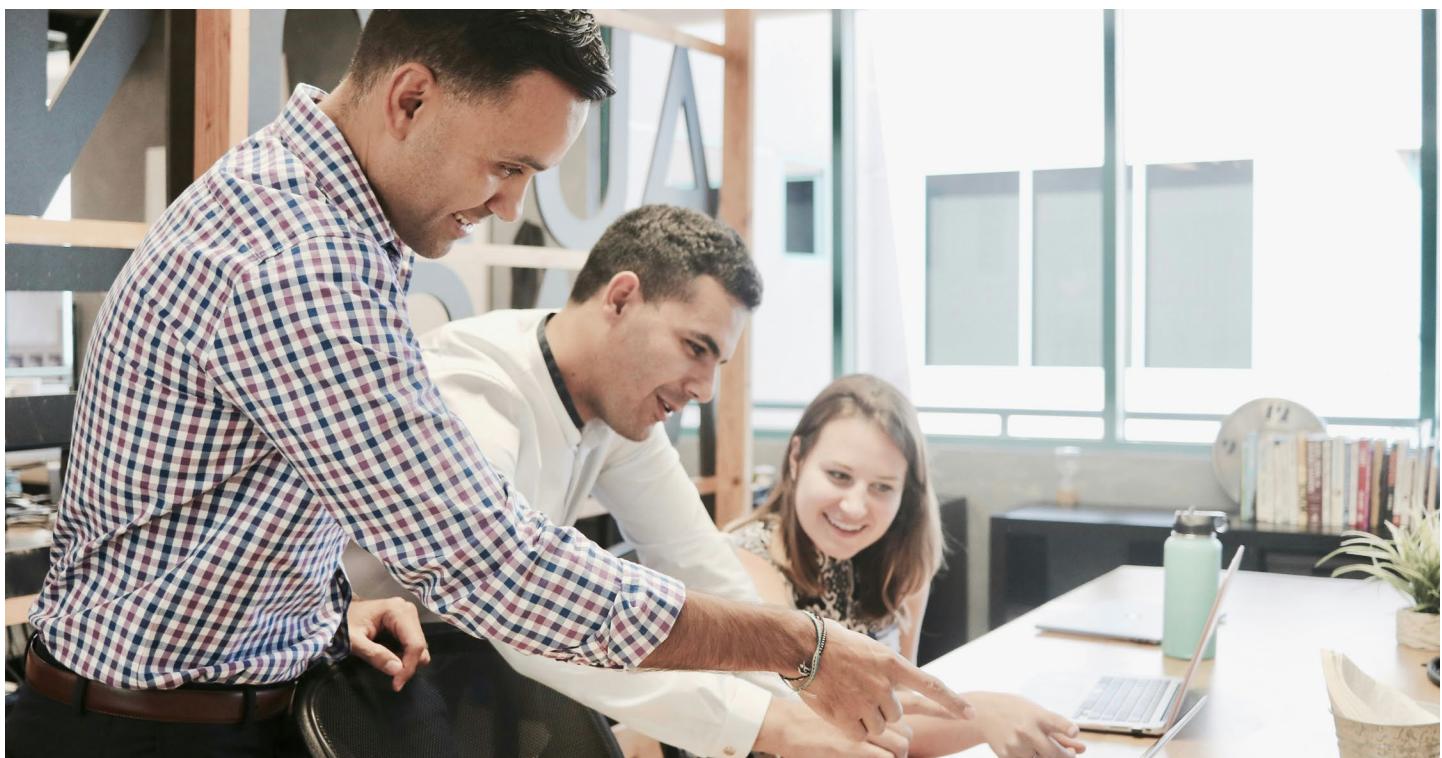
We think that the job of a developer is pretty cool. Developers are at the forefront of technological advancements and help shape how we live, work, and interact with the world.

Their jobs are fundamentally tied to people—the users of the software and applications they create. Yet developers often work in environments that prioritize features and innovation. There's less emphasis on figuring out whether these features provide value to the people who use the products they make.

Here we provide compelling evidence showing that an approach to software development that prioritizes the end user positively impacts employees and organizations alike.

**This year, we asked questions focused on understanding whether developers:**

1. Incorporate user feedback to revisit and reprioritize features
2. Know what users want to accomplish with a specific application/service
3. Believe focusing on the user is key to the success of the business
4. Believe the user experience is a top business priority





## Our findings and what they mean

Our data strongly suggests that organizations that see users' needs and challenges as a guiding light make better products.

We find that focusing on the user increases productivity and job satisfaction, while reducing the risk of burnout.

Importantly, these benefits extend beyond the individual employee to the organization. In previous years, we've highlighted that high performing organizations deliver software quickly and reliably. The implication is that software-delivery performance is a requirement for success.

However, our data indicates there's another path that leads to success:

Developers and their employers, and organizations in general, can create a user-centered approach to software development.

We find that when organizations know and understand users' needs, stability and throughput of software delivery are not a requirement for product quality. Product quality will be high as long as the user experience is at the forefront.

When organizations don't focus on the user, don't incorporate user feedback into their development process, doubling down on stable and fast delivery is the only path to product quality (see Figure 15).

We understand the inclination that some organizations might have to focus on creating features and innovating on technologies. At face value, this approach makes sense. After all, developers most certainly know the ins and outs of the technology much better than their average user.

However, developing software based on assumptions about the user experience increases the likelihood of developers building features that are perhaps shiny but hardly used.<sup>1</sup>

When organizations and employees understand how their users experience the world, they increase the likelihood of building features that address the real needs of their users. Addressing real user needs increases the chances of those features being actually used.

**Focus on building for your user and you will create delightful products.**



Figure 15: Product performance and delivery throughput across 3 levels of user centricity

## Why is a user-centered approach to software development such a powerful philosophy and practice?

Academic research shows that deriving a sense of purpose from work benefits employees and organizations.<sup>2,3</sup>

For example, a recent survey showed that 93% of workers reported that it's important to have a job where they feel the work they do is meaningful.<sup>4</sup> In a similar vein, another survey found that on average, respondents were willing to relinquish 23% of their entire future earnings if it meant they could have a job that was always meaningful.<sup>5</sup>

That's an eye-popping trade-off employees are willing to make. It tells us something about what motivates people, and that people want to spend their time doing something that matters.

**“It would be grand if everybody could work at a company that affects individuals outside of the company, or [in] your local community in a positive way. That’s not always the case. That’s not always possible. A lot of the grand vision of autonomous driving is that it is going to enable people that can drive [to] sleep while they’re on a motorway. That’s not why I’m here. I want to help people that can’t drive to be able to get about, wherever they want, have the freedom to do whatever they want to do.” (P2)<sup>6</sup>**

## Provides a clear sense of direction:

A user-centered approach to software development can fundamentally alter how developers view their work. Instead of shipping arbitrary features and guessing whether users might use them, developers can rely on user feedback to help them prioritize what to build.

This approach gives developers confidence that the features they are working on have a reason for being. Suddenly, their work has meaning: to ensure people have a superb experience when using their products and services. There's no longer a disconnect between the software that's developed and the world in which it lives.

Developers can see the direct impact of their work through the software they create.

**“We are, as a company, under pressure to deliver. So, all of these, like, nice shiny things, or discussion points about how you want to improve, it’s kind of, like, with the recent change in how we’re structured, we’re focusing on delivery, not quality, and for me, personally, that’s kind of a big bugbear.” (P9)**

## **Increases cross-functional collaborations:**

Even the most talented developer doesn't build software on their own. Building high-quality products takes the collaboration of many people often with different yet complementary talents.

A user-centered approach to development allows developers to engage in cross-functional collaborations across the organization. In doing so, their responsibilities extend beyond simply shipping software. They are now part of a team driven to create incredible experiences for the people who use them.

This approach to software development can help developers break out of silos, seek alignment, foster teamwork, and create opportunities to learn more from others. Problem solving takes a different shape. It's not just about how to solve technical problems, but how to do so in ways that serve the user best.

This approach can help increase employee engagement and create an even more intellectually-stimulating environment that can stave off the feelings of stagnation that are associated with burnout.

# **What can organizations do?**

Based on our findings, we recommend organizations invest time and resources in getting to know their users. Focus on understanding who you are building for, and the challenges they experience. We strongly believe this is a worthy investment.

Resist the temptation to make assumptions about your users. Observe them in their environments, ask them questions, and be humble enough to pivot based on what they tell you. In doing so, developers will be more productive and be less prone to burnout while delivering higher quality products.

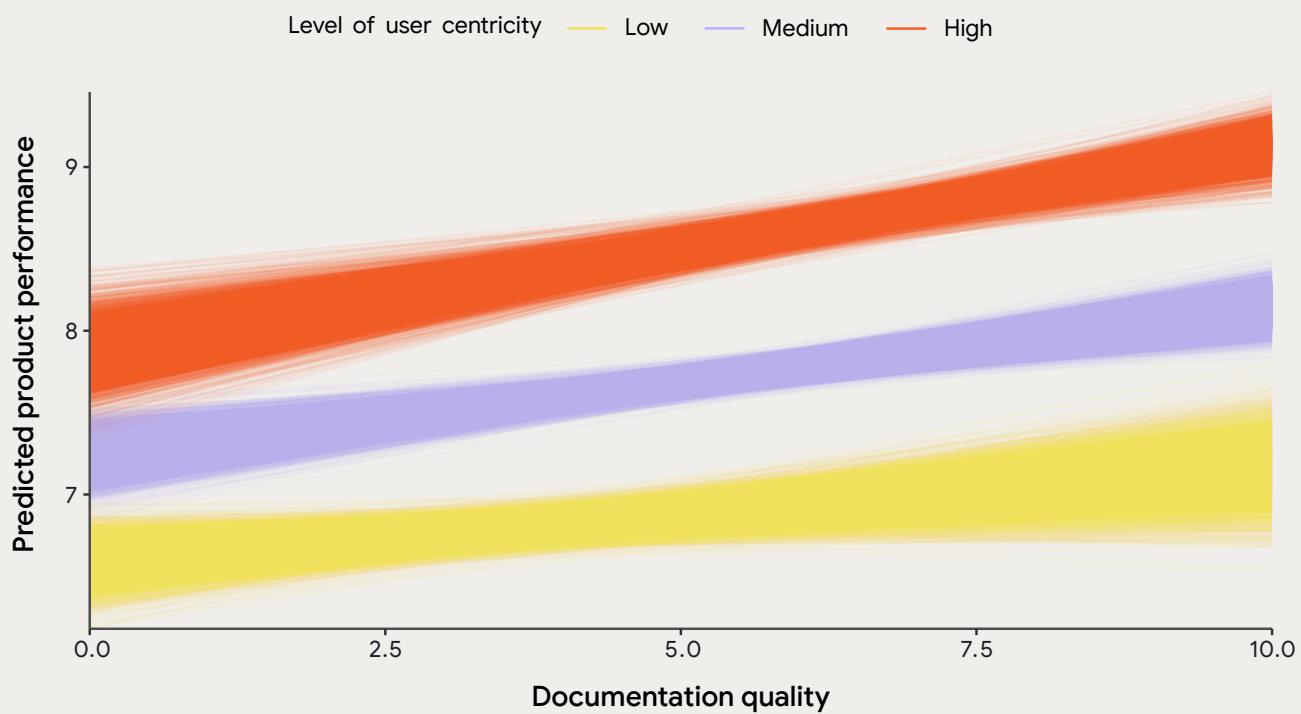
## The combination of good docs and a user-centered approach to software development is a powerful one.

Teams that focus on the user see an increase in product performance. When this focus on the user is combined with an environment of quality internal documentation, this increase in product performance is amplified (see Figure 16). This finding is similar to the behavior that we see where documentation amplifies a technical capability's impact on organizational performance.<sup>7</sup>

Documentation helps propagate user signals and feedback across the team and into the product itself.

We see that internal documentation doesn't meaningfully affect predicted product performance without user signals. However, if a team has a high quality internal documentation then user signals included in it will have a higher impact on product performance.

We started to look at documentation in 2021, and every year we continue to find extensive impact of quality documentation. This year's findings adds internal documentation's impact on predicted product performance to the list.



The graph is a composite of 12000 lines from simulations trying to estimate the most plausible pattern  
Figure 16: Product performance and documentation quality across 3 levels of user centricity

## Culture of documentation

The Agile manifesto advocates for “working software over comprehensive documentation”.<sup>7</sup> We continue to find, however, that quality documentation is a key component of working software.

“Comprehensive documentation” may be a phrase standing in for unhealthy practices, which might include documentation. Problematic documentation includes documentation that is created only for bureaucratic purposes, or to paper over mistrust between management and employees. An unhealthy documentation culture can also include writing documentation, but not maintaining or consolidating the documentation.

In these cases, our measure of quality documentation would likely score low. This type of content is written for the wrong audience so doesn’t perform as well when you try to use it while doing your work. And too much documentation can be as problematic as not enough.

Our measure of quality documentation includes attributes like findability and reliability of the documentation. Remember, for internal documentation, the primary audience is your colleagues or even your future self trying to accomplish specific tasks.<sup>8</sup> Teams with a healthy documentation culture have a focus on serving these readers. This is another way that focusing on your users matters.

**You can create a healthy culture of documentation on your own teams by following the practices we’ve identified to create quality documentation, such as:**

---

Documenting critical use cases.

---

Taking training in technical writing.

---

Defining ownership and processes to update the documentation.

---

Distributing documentation work within the team.

---

Maintaining documentation as part of the software development lifecycle.

---

Deleting out-of-date or redundant documentation.

---

Recognizing documentation work in performance reviews and promotions.

# The perils of ever-shifting priorities

We all know the feeling. You've spent the last few months working on a new feature. You know it's the right thing to build for your users, you are focused and motivated. Suddenly, or seemingly so, the leadership team decides to change the organization's priorities. Now it's unclear whether your project will be paused, scrapped, Frankensteined, or mutated.

This common experience can have profound implications for employees and organizations. Here we examine what happens when organizations constantly shift their priorities.

## Our findings and what they mean

Overall, our findings show small but meaningful decreases in productivity and substantial increases in burnout when organizations have unstable priorities.

Our data indicates it is challenging to mitigate this increase in burnout. We examined whether having strong leaders, good internal documents, and a user-centered approach to software development can help counteract the effect of shifting priorities on burnout.

The answer is: They can't. An organization can have all these positive traits and, if priorities are unstable, employees will still be at risk of experiencing burnout.

## Why are unstable organizational priorities bad for employees' well-being?

We hypothesize that unstable organizational priorities increase employee burnout by creating unclear expectations, decreasing employees' sense of control, and increasing the size of their workloads.

To be clear, we believe that the problem is not with changing priorities themselves. Business goals and product direction shift all the time. It can be good for organizational priorities to be malleable.

We believe it is the frequency with which priorities change that has a negative impact on employees' well-being. The uncertainty that accompanies unstable priorities implies something chronic about the frequency with which priorities change.

Decades of academic research have shown the detrimental effects of chronic stress on health and well-being.<sup>9</sup> We see parallels between research on chronic stress and our findings. Chronic instability increases uncertainty and decreases perceived control. This combination is an excellent recipe for burnout.

## What happens when priorities stabilize?

Our findings here are a little puzzling. We find that when priorities are stabilized, software delivery performance declines. It becomes slow and less stable in its delivery.

We hypothesize that this might be because organizations with stable priorities might have products and services that are generally in good shape so changes are made less frequently. It is also possible that stability of priorities leads to shipping less and in larger batches than recommended.

Nevertheless, we find this to be an unexpected finding. Why do you think stabilizing organizational priorities decreases the speed and stability of software delivery?

## **Building AI for end users creates stability in priorities, but not stability in delivery.**

Incorporating AI-powered experiences for end users stabilizes organizational priorities. This sounds like a flashy endorsement for AI. However, we do not interpret this finding as telling us something meaningful about AI itself.

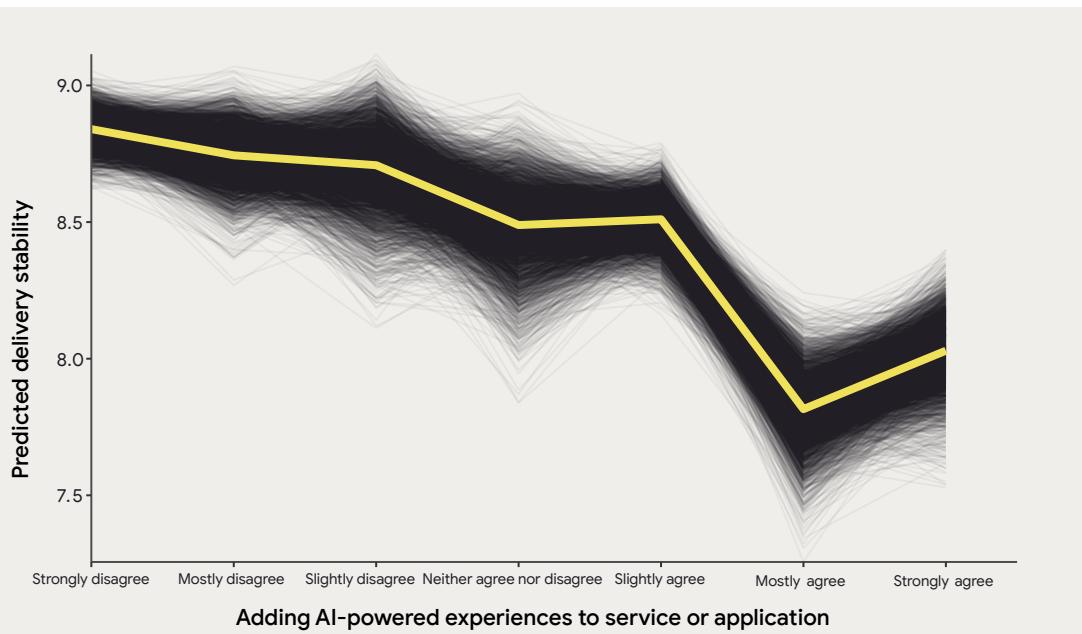
Instead, we believe that shifting efforts towards building AI provides clarity and a northstar for organizations to follow. This clarity, and not AI, is what leads to a stabilization of organizational priorities.

This is worth highlighting because it tells us something about what happens to organizations when new technologies emerge. New technologies bring change and organizations need time to adapt.

This period likely leads to a destabilization of priorities as leaders try to figure the best move for the organization. As the dust settles, and organizations clarify their next steps, priorities begin to stabilize.

Priorities stabilizing, however, doesn't immediately translate into the software delivery process stabilizing. Our analyses show that a shift to adding AI-powered experiences into your service or application comes with challenges and growing pains.

We find that teams that have shifted have a significant 10% decrease in software delivery stability relative to teams who have not. Here is a visualization depicting the challenge.



\*Each line is one of 4000 simulations trying to estimate the most plausible pattern

Figure 17: Software delivery stability as a function of adding AI-powered experiences to service or application

## What can organizations do?

The answer, while easy, might not be so simple. Based on our findings, we recommend organizations focus on stabilizing their priorities. This is one sure way to counteract the negative effects of unstable priorities on employee burnout.

Our findings show the negative effects of unstable priorities are resistant to having good leaders, good documentation, and a user-centered approach to software development. This leads us to believe that, aside from creating stability, there's not much organizations can do to avoid burnout aside from finding ways to (1) stabilize priorities and (2) shield employees from having their day-to-day be impacted by the constant shift in priorities.

1. <https://www.nngroup.com/articles/bridging-the-designer-user-gap/>
2. <https://executiveeducation.wharton.upenn.edu/thought-leadership/wharton-at-work/2024/03/creating-meaning-at-work/>
3. <https://www.apa.org/pubs/reports/work-in-america/2023-workplace-health-well-being>
4. <https://bigthink.com/the-present/harvard-business-review-americans-meaningful-work/>
5. <https://hbr.org/2018/11/9-out-of-10-people-are-willing-to-earn-less-money-to-do-more-meaningful-work>
6. (P[N]), for example (P1), indicates pseudonym of interview participants.
7. [https://cloud.google.com/blog/products/devops-sre/deep-dive-into-2022-state-of-devops-report-on-documentation-and Accelerate State of DevOps Report 2023 - https://dora.dev/research/2023/dora-report](https://cloud.google.com/blog/products/devops-sre/deep-dive-into-2022-state-of-devops-report-on-documentation-and-accelerate-state-of-devops-report-2023)
8. <https://agilemanifesto.org/>
9. Other audiences exist, such as management, regulators, or auditors.
10. Cohen S, Janicki-Deverts D, Miller GE. Psychological Stress and Disease. JAMA. 2007;298(14):1685–1687.doi:10.1001/jama.298.14.1685

# Leading transformations



A lot needs to be in place for transformation to work. This year, we've found high-performing teams are ones that prioritize stability, focus on their users, have good leaders, and craft quality documentation. Our research points to some useful paths in helping you plot a course towards successful transformation.

We have found the key to success is to approach transformation from a mindset of continuous improvement. High performers in our study understand the variables holding them back, and methodically and continuously improve using the DORA metrics as a baseline. While long-term success requires excellence in all pillars, a decade of DORA research has pointed us to four specific, impactful ways to get started on driving transformation in your own organization.

# Transformational leadership

Transformational leadership is a model in which leaders inspire and motivate employees to achieve higher performance by appealing to their values and sense of purpose, facilitating wide-scale organizational change.

These leaders encourage their teams to work towards a common goal through the following dimensions:<sup>1</sup>

## Vision

They have a clear vision of where their team and the organization are going.

## Inspirational communication

They say positive things about the team; make employees proud to be a part of their organization; encourage people to see changing conditions as situations full of opportunities.

## Intellectual stimulation

They challenge team members to think about old problems in new ways and to rethink some of their basic assumptions about their work.

## Supportive leadership

They consider others' personal feelings before acting; behave in a manner which is thoughtful of others' personal needs.

## Personal recognition

They commend team members when they do a better-than-average job; acknowledge improvement in quality of team members' work.

This year, we saw that transformational leadership leads to a boost in employee productivity. We see that increasing transformational leadership by 25% leads to a 9% increase in employee productivity.

Transformational leadership can help improve more than just productivity. Having good leaders can also lead to:

- A decrease in employee burnout
- An increase in job satisfaction
- An increase in team performance
- An improved product performance
- An improved organizational performance

Our research found a statistically significant relationship between the above qualities of leadership and IT performance in 2017. High-performing teams had leaders with strong scores across all five characteristics and low-performing teams had the lowest scores. Additionally, we saw that there's a strong correlation between transformative leadership and Employee Net Promoter Score (eNPS), the likelihood to recommend working at a company.

That said, transformative leadership by itself does not lead to high performance, but should be seen as an enabler.

Transformative leadership plays a key role in enabling the adoption of technical and product-management capabilities and practices. This is enabled by (1) delegating authority and autonomy to teams; (2) providing them the metrics and business intelligence needed to solve problems; and (3) creating incentive structures around value delivery as opposed to feature delivery.

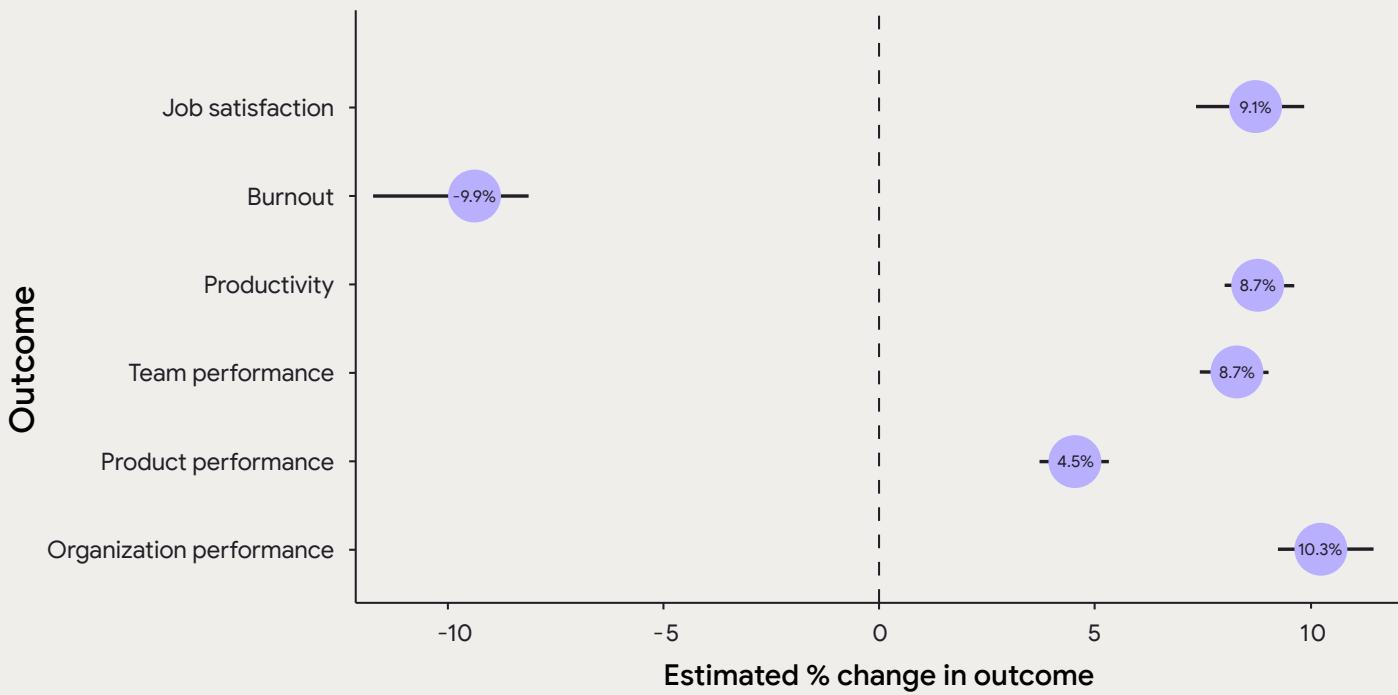
Transformation takes time and requires tools. Resources must be allocated by leadership specifically for the task of improvement. Good leaders play a key role in providing teams with the time and funding necessary to improve. Engineers should not be expected to learn new things and automate on their off time, this should be baked into their schedule.

Our research has helped to flip the narrative of IT being a cost-center to IT being an investment that drives business success. In 2020, we wrote the ROI of DevOps whitepaper,<sup>2</sup> which contains calculations you can use to help articulate potential value created by investing in IT improvement.

Monetary return is only one of the returns you can expect from this investment. Our research in 2015 showed that, “organizational investment in DevOps is strongly correlated with organizational culture; the ability of development, operations, and infosec

teams to achieve win-win outcomes; lower levels of burnout; more effective leadership; and effective implementation of both continuous delivery and lean management practices.”<sup>3</sup> We recommend dedicating a certain amount of capacity specifically for improvement.

## If transformational leadership increases by 25%...



Point = estimated value

Error bar = 89% uncertainty interval

Figure 18: Impacts of transformational leadership on various outcomes.

# Be relentlessly user-centric

This year's research shows that organizations with strong leaders and a focus on building software that addresses user needs leads to the development of better products: It's a powerful combination. When the user is at the center of software development, leaders have a clear vision to articulate.

The ultimate goal is for users to love the products we create. As we discuss in the [Developer experience](#) chapter, focusing on the user gives product capabilities a reason to exist. Developers can confidently build these features knowing they'll help improve the user experience.

We see that teams that have a deep desire to understand and align to their users' needs and the mechanisms to collect, track, and respond to user feedback have the highest levels of organizational performance. In fact, organizations can be successful even without high levels of software velocity and stability, as long as they are user-focused. In 2023 we saw user-centered teams have a 40% higher level of organizational performance compared to those that did not,<sup>4</sup> and in 2016 we also saw that user-centered teams had better organizational performance.

This year's research echoes previous findings. Teams that focus on the user make better products.

Not only do products improve, but employees are more satisfied with their jobs and less likely to experience burnout.

Fast, stable software delivery enables organizations more frequent opportunities to experiment and learn. Ideally, these experiments and iterations are based on user feedback. Fast and stable software delivery allows you to experiment, better understand user needs, and quickly respond if those needs are not being met.

Having speed and stability baked into your delivery also allows you to more easily adjust to market changes or competition.

It is important to remember that your internal developers are also users. Internal Developer Platforms (IDPs) are a way your organization can deliver value to developers that in turn deliver value to external users or other internal users.

Our research shows that successful IDPs are developed as a product and focus on user centricity to deliver an experience that allows developers to work independently. An IDP deployed in this way leads to higher individual productivity, higher team productivity, and higher organizational performance.

# Become a data-informed organization

The ability to visualize your progress toward success is critical. Over the last 10 years we have made the case for becoming a data-informed organization. DORA's four key metrics<sup>5</sup> have become a global standard for measuring software delivery performance, but this is only part of the story. We have identified more than 30 capabilities and processes<sup>6</sup> that can be used to drive organizational improvement.

The value in the metrics lies in their ability to tell you if you are improving. The four key metrics should be used at the application and service levels, and not at the organization or line-of-business level. The metrics should be used to visualize your efforts in continuous improvement and not to compare teams — and certainly not to compare individuals.

The metrics should also not be used as a maturity model for your application or service teams. Being a low, medium, high, or elite performer is interesting, but we urge caution as these monikers have little value in the context of your transformation journey.

As our research progresses and evolves, we encourage you to think beyond the four keys. It has become clear that user feedback metrics are as important as the four key metrics. We believe this is because most teams have devised workable solutions for improving

speed and stability. As a result, the benefits gained by speed and stability are diminished as higher performance becomes ubiquitous.

Thinking about transformation holistically, we recommend creating dashboards and visualizations that combine both technical metrics (such as our four keys and reliability metrics) and business metrics. This helps bridge the gap between the top-down and bottom-up transformation efforts. This also helps connect your northstar, OKRs, and employee goals with the investments made in IT. They can help quantify the ROI.

We believe metrics are a requirement for excellence. Metrics facilitate decision making. The more metrics you collect, quantitative and qualitative, the better and more informed decisions you can make. People will always have opinions on the value of the data or the meaning of the data, but using data as the basis by which to make a decision is often preferable to relying on opinion or intuition.



## Be all-in on cloud or stay in the data center

We have been investigating the relationship between the NIST defined-5 characteristics of cloud computing<sup>7</sup> (on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service also known as flexible infrastructure) and organizational performance since 2018. We see that successful teams are more likely to take advantage of flexible infrastructure than less successful teams.

Last year, our research led us to the most striking bit of information on this topic to date: Using the cloud without taking advantage of the five characteristics can be detrimental and predicts decreased organizational performance.

Organizations may be better off staying in the data center if they are not willing to radically transform their application or service. Of course, to accomplish this, it is not simply adopting tools or technologies, but often an entire new paradigm in designing, building, deploying, and running applications. Making large-scale changes is easier when starting with a small number of services, we recommend an iterative approach that helps teams and organizations to learn and improve as they move forward.

# Summary

What we've seen consistently over the last 10 years is that transformation is a requirement for success. What many organizations misunderstand is that transformation isn't a destination, but a journey of continuous improvement.<sup>8</sup> Our research is clear: Companies that are not continuously improving are actually falling behind. Conversely, companies that adopt a mindset of continuous improvement see the highest levels of success.

On this journey, be aware that you will likely hit a little bit of pain and discomfort along the way. Our research has shown an initial drop in performance followed by big gains (also known as the “j-curve”) with DevOps,<sup>9</sup> SRE,<sup>10</sup> and this year with Platform Engineering. This is normal, and if you are continuously improving, things will get better and you will come out the other end in much better shape than when you started.

The idea of a never-ending journey can seem daunting. It's easy to get stuck in planning or designing the perfect transformation. The key to success is rolling up your sleeves and just getting to work. The goal of the organization and your teams should be to simply be a little better than you were yesterday. The goal of our last 10 years of research and into the future is to help you get better at getting better.

- 
1. Dimensions of transformational leadership: Conceptual and empirical extensions - Rafferty, A. E., & Griffin, M. A.
  2. The ROI of DevOps Transformation - <https://dora.dev/research/2020/>
  3. 2015 State of DevOps Report <https://dora.dev/research/2015/2015-state-of-devops-report.pdf#page=25>
  4. 2023 Accelerate State of DevOps Report - <https://dora.dev/research/2023/dora-report/2023-dora-accelerate-state-of-devops-report.pdf#page=17>
  5. DORA's Four Key Metrics <https://dora.dev/guides/dora-metrics-four-keys/>
  6. DORA's capabilities and processes <https://dora.dev/capabilities/>
  7. NIST defined-5 characteristics of cloud computing <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>
  8. Journey of continuous improvement <https://cloud.google.com/transform/moving-shields-into-position-organizing-security-for-digital-transformation>
  9. 2018 Accelerate State of DevOps Report <https://dora.dev/research/2018/dora-report/>
  10. 2022 State of DevOps Report <https://dora.dev/research/2022/dora-report/>