

# A Continuous Improvement Story



Nathen Harvey  
Developer Advocate  
@nathenharvey



Amanda Lewis  
Developer Advocate  
@swansama



Google Cloud



## **DORA Improvement Award Employee Productivity Team**

**4 years previous....**

# **Nathen starts at Fun Shirts**

**2 weeks after Nathen starts at  
Fun Shirts, he notices he is not  
in the Employee Directory**

# Fun Shirts Employee Directory



**John Goode**  
Employee Data Team  
[Email](#)  
[Chat](#)



**George Harvey**  
Employee Data Team  
[Email](#)  
[Chat](#)



**Amanda Lewis**  
Employee Data Team  
[Email](#)  
[Chat](#)

**4 weeks after Nathen starts at  
Fun Shirts, he notices he is still  
not in the Employee Directory**

# Fun Shirts Employee Directory



**John Goode**  
Employee Data Team  
[Email](#)  
[Chat](#)



**George Harvey**  
Employee Data Team  
[Email](#)  
[Chat](#)



**Amanda Lewis**  
Employee Data Team  
[Email](#)  
[Chat](#)

**Finally!! 5 weeks after Nathen  
starts at Fun Shirts, he is in the  
Employee Directory...but...**

# Fun Shirts Employee Directory



**Nathan Harvey**  
Director of Avocado Shirt Sales

[Email](#)  
[Chat](#)

# Fun Shirts Employee Directory



**Nathan Harvey**

Director of Avocado Shirt Sales

**Nathen**

[Email](#)

[Chat](#)

Name

Nathen Harvey

Email

nh@funshirts.yay

Describe Issue:

My name is misspelled in  
the Employee Directory.  
My name is spelled Nathen  
with an e, not an a.

Submit Ticket

**5 weeks after  
Nathen started,  
he opens a bug.**

**8 weeks after Nathen starts at  
Fun Shirts, he reaches out for  
an update on his bug report..**

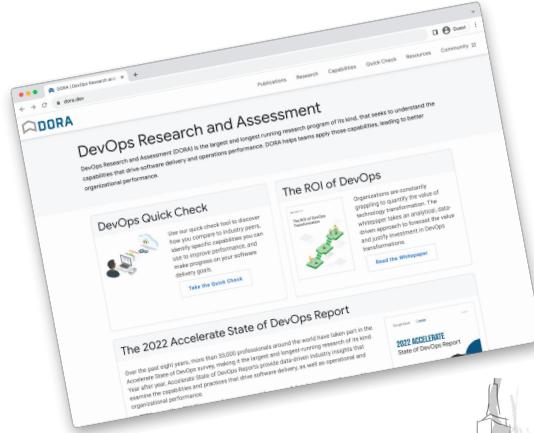


Nathen,

Hi! Thank you for reaching out for an update. We apologize for the misspelling of your name in the employee directory. We have submitted the change, and should be published with our next release, which is scheduled in 2 weeks.

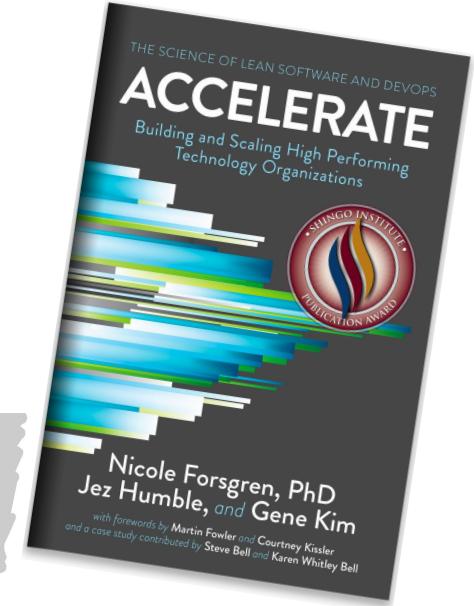
Smiles,

Amanda



# DORA

DEVOPS RESEARCH & ASSESSMENT



Google Cloud

## Take the DORA DevOps Quick Check

Measure your team's software delivery performance in less than a minute! Compare it to the rest of the industry by responding to five multiple-choice questions. Compare your team's performance to others, and discover which DevOps capabilities you should focus on improving. We don't store your answers or personal information.



QUESTION 1 OF 5

### Lead time



For the primary application or service you work on, what is your lead time for CI (that is, how long does it take to go from code committed to code successfully running in production)?

- More than six months
- One to six months
- One week to one month
- One day to one week
- Less than one day
- Less than one hour

QUESTION 2 OF 5

### Deploy frequency



For the primary application or service you work on, how often does your organization deploy code to production or release it to end users?

- Fewer than once per six months
- Between once per month and once every six months
- Between once per week and once per month
- Between once per day and once per week
- Between once per hour and once per day
- On demand (multiple deploys per day)

QUESTION 3 OF 5

### Change fail percentage



For the primary application or service you work on, what percentage of changes to production or releases to users result in degraded service (for example, lead to service impairment or service outage) and subsequently require remediation (for example, require a hotfix, rollback, fix forward, patch)?

- 0–15%
- 16–30%
- 31–45%
- 46–60%
- 61–75%
- 76–100%

QUESTION 4 OF 5

### Time to restore



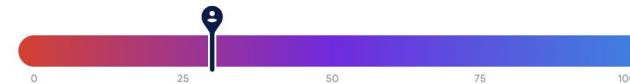
For the primary application or service you work on, how long does it generally take to restore service when a service incident or a defect that impacts users occurs (for example, unplanned outage, service impairment)?

- More than six months
- One to six months
- One week to one month
- One day to one week
- Less than one day
- Less than one hour

# Your software delivery performance

Your performance:

30



# We discovered:

- 6 weeks - average time from hire until listed in employee directory
- 6 weeks - average time for small changes to be deployed
- Less than a day - average time it takes for a employee record to be removed when they leave the company

# **Team Exploration:**

Investigate why it takes 5 weeks for adding new employees, but only 2 days for removing them from the directory.

# Finding:

Different process for approving employee removal  
deployments

# Streamlining change approval core

Most IT organizations have change management processes to manage the life cycle of changes to IT services, both internal and customer-facing. These processes are often the primary controls to reduce the operational and security risks of change.

Change management processes often include approvals by external reviewers or change approval boards (CABs) to promote changes through the system.

Compliance managers and security managers rely on change management processes to validate compliance requirements, which typically require evidence that all changes are appropriately authorized.

Research by DevOps Research and Assessment (DORA), presented in the [2019 State of DevOps Report \(PDF\)](#), finds that change approvals are best implemented through peer review during the development process, supplemented by automation to detect, prevent, and correct bad changes early in the software delivery life cycle. Techniques such as [continuous testing](#), [continuous integration](#), and [comprehensive monitoring and observability](#) provide early and automated detection, visibility, and fast feedback.

Further, organizations can improve their performance by doing a better job of communicating the existing process and helping teams navigate it efficiently. When team members have a clear understanding of the change approval process, this drives higher performance.

## How to implement a change approval process

Two important goals of the change approval process are decreasing the risk of making changes, and satisfying regulatory requirements. One common regulatory requirement is segregation of duties, which states that changes must be approved by someone other than the author, thus ensuring that no individual has end-to-end control over a process.

Traditionally, these goals have been met through a heavyweight process involving approval by people external to the team proposing the change: a change advisory board (CAB) or a senior manager. However, DORA's research shows that these approaches have a negative impact on software delivery performance. Further, no evidence was found to support the hypothesis that a more formal, external review process was associated with lower change fail rates.

Such heavyweight approaches tend to slow down the delivery process leading to the release of larger batches less frequently, with an accompanying higher impact on the production system that is likely to be associated with higher levels of risk and thus higher change fail rates. DORA's research found this hypothesis was supported in the data.

## DevOps Capabilities

### TECHNICAL

[Code maintainability](#)

[Continuous delivery](#)

[Continuous integration](#)

[Database change management](#)

[Deployment automation](#)

[Empowering teams to choose tools](#)

[Flexible infrastructure](#)

[Loosely coupled architecture](#)

[Monitoring and observability](#)

[Shifting left on security](#)

[Test automation](#)

[Test data management](#)

[Trunk-based development](#)

[Version control](#)

### PROCESS

[Customer feedback](#)

[Documentation quality](#)

[Monitoring systems to inform business decisions](#)

[Proactive failure notification](#)

[Streamlining change approval](#)

[Team experimentation](#)

# Ways to measure change approval in your systems

Now your teams can list possible ways to measure change approval:

Factor to test	What to measure
Can changes be promoted to production without manual change approvals?	<p>The percentage of changes that do (or do not) require a manual change to be promoted to production.</p> <p><b>Tip:</b> You can also measure this factor based on risk profile what percentage of low-, medium-, and high-risk changes require a manual change to be promoted to production?</p>
Do production changes need to be approved by an external body before deployment or implementation?	<p>The amount of time changes spend waiting for approval from external bodies.</p> <p><b>Tip:</b> As you shift approvals closer to the work, measure the amount of time spent waiting for approval from local approval bodies or reviewers.</p>
Do you rely on peer review to manage changes?	You can also measure this factor by risk profile. Measure number or proportion of changes that require approval from external bodies, as well as the time spent waiting for those approvals
Do team members have a clear understanding of the process to get changes approved for implementation	The extent to which team members are confident they can get changes through the approval process in a timely manner and know the steps it takes to go from "submitted" to "accepted" every time for all the types of changes they typically make

While you consider your own environment, you will likely develop your own measures to understand and gain insight into your change approval processes. We suggest you use these to not only measure your process but also work to improve it.

# 2014 STATE OF **DEVOPS** REPORT



Presented by



IT REVOLUTION PRESS

ThoughtWorks

**Peer-reviewed** change approval process. We found that when external approval (e.g., change approval boards) was required in order to deploy to production, IT performance decreased. When the technical team held itself **accountable for the quality of its code** through peer review, performance **increased**.



**Automation** is a huge boon to organizations.

High performers automate significantly more of their configuration management, testing, deployments and change approval processes than other teams. The result is more time for **innovation** and a faster feedback cycle.



Begin shifting the change review process to an earlier phase of the development cycle, and rely on **peer review and automated testing** rather than a change review board. Ultimately, this will eliminate the need for a change review board altogether, and the team can move forward with a **faster and more reliable process** for reviewing changes.



High performers are doing significantly less manual work, and have **automated**:

- 33 % more of their configuration management
- 27 % more of their testing
- 30 % more of their deployments
- 27 % more of their **change approval processes**



Heavyweight change approval processes, such as change approval boards, **negatively impact speed and stability**. In contrast, having a **clearly understood process for changes drives speed and stability**, as well as reductions in burnout.

# Thinking about our users:



**Nathen Harvey**  
Director of Avocado Shirt Sales

[Email](#)  
[Chat](#)

Presented by



Google Cloud

# Accelerate State of DevOps Report 2023



## Premiere Sponsors



digital.ai



BROADCOM SOFTWARE  
Deloitte. SLEUTH  
Qarik

v. 2023-10



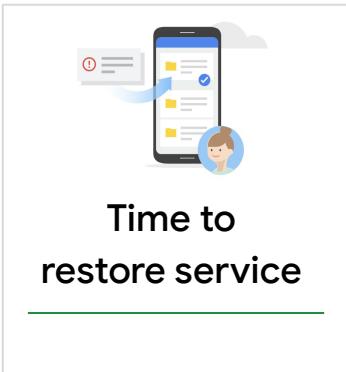
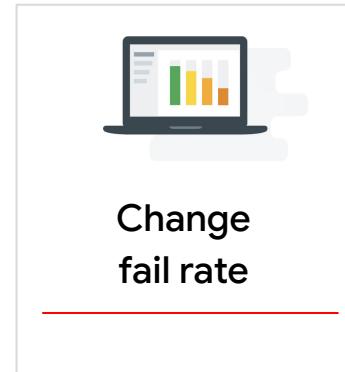
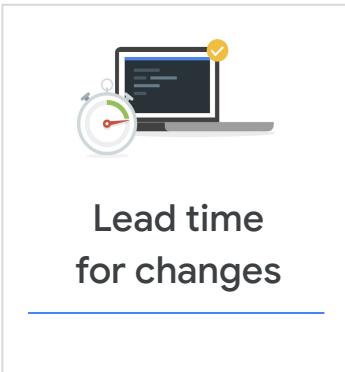
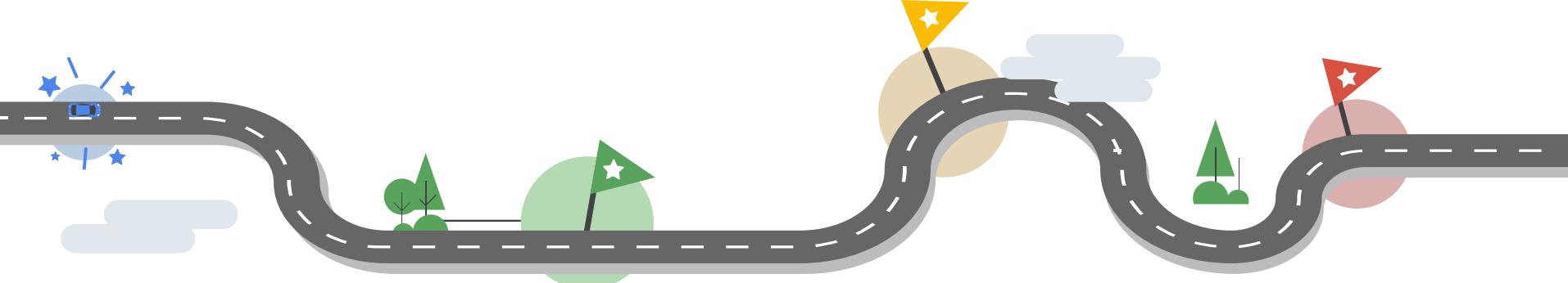
Teams that focus on the user  
have  
**40% higher** organizational  
performance

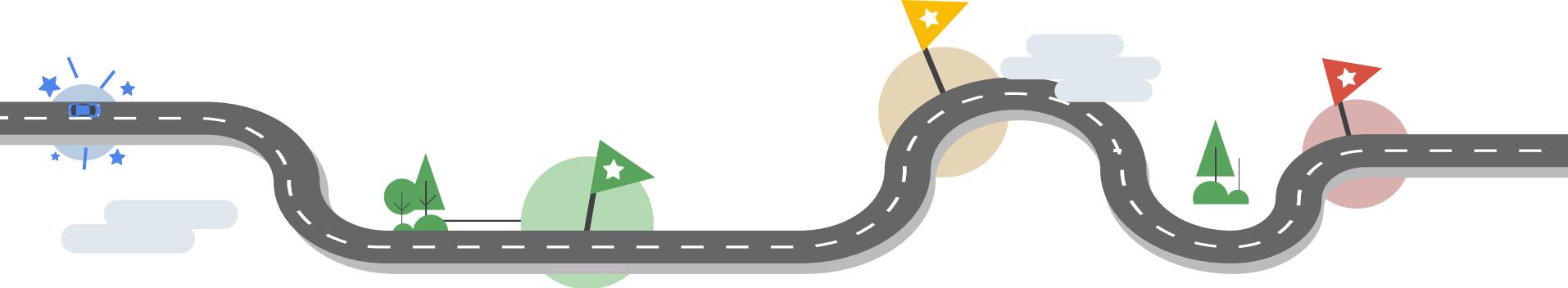
## Effect of user-centrism on ...

Organizational performance		Substantial increase
Team performance		Substantial increase
Software delivery performance		Minor increase
Operational performance		Substantial increase
Trunk-based development		Substantial increase
Reliability practices		Substantial increase
Continuous integration		Substantial increase
Continuous delivery		Substantial increase
Loosely coupled architecture		Substantial increase
Burnout		Minor decrease*
Job satisfaction		Substantial increase
Productivity		Substantial increase



User focus predicts a lot of good things and provides insights for all sorts of teams.





## Code Reviews



Lead time  
for changes



Deployment  
frequency



Change  
fail rate



Time to  
restore service



Committing code **sooner** is better

Working in **small batches** is better: Large “patch bombs” are harder and slower to merge into a project than smaller, more readable patchsets since maintainers need more time to review the changes.

Presented by



Google Cloud

Accelerate

# State of DevOps Report 2023



Premiere Sponsors



digital.ai



BROADCOM SOFTWARE  
Deloitte. SLEUTH  
Qarik

v. 2023-10



Teams with faster code reviews  
have **50% higher** higher  
software delivery performance



## Code Reviews



Lead time  
for changes

## Loosely Coupled Architecture



Deployment  
frequency



Change  
fail rate



Time to  
restore service

# Loosely coupled architecture

core

Research from the [DevOps Research and Assessment \(DORA\)](#) team shows that architecture is an important predictor for achieving continuous delivery. Whether you're using Kubernetes or mainframes, your architecture enables teams to adopt practices that foster higher levels of software delivery performance.

When teams adopt continuous delivery practices, adopting the following architectural practices drives successful outcomes:

- Teams can make large-scale changes to the design of their systems without the permission of somebody outside the team or depending on other teams.
- Teams are able to complete work without needing fine-grained communication and coordination with people outside the team.
- Teams deploy and release their product or service on demand, independently of the services it depends on or of other services that depend on it.
- Teams do most of their testing on demand, without requiring an integrated test environment.
- Teams can deploy during normal business hours with negligible downtime.

It's possible to achieve these outcomes with mainframe technologies. It's also possible to fail to achieve them even when using the latest, most trendy technologies. Many organizations invest lots of time and effort in adopting technologies, but fail to achieve critical software delivery outcomes, due to limitations imposed by architecture.

When the architecture of the system is designed to enable teams to test, deploy, and change systems without dependencies on other teams, teams require little communication to get work done. In other words, both the architecture and the teams are loosely coupled.

This connection between communication bandwidth and systems architecture was first discussed by Melvin Conway, who said, "organizations which design systems ... are constrained to produce designs which are copies of the communication structures of these organizations." To counteract tightly-coupled architectures and help support better communication patterns, teams and organizations can use the [Inverse Conway Maneuver](#), whereby team structures and patterns are designed to promote the expected architectural state. In this way, team communication patterns support and enforce the architectural patterns that are built.

## DevOps Capabilities

### TECHNICAL

Code maintainability

Continuous delivery

Continuous integration

Database change management

Deployment automation

Empowering teams to choose tools

Flexible infrastructure

Loosely coupled architecture

Monitoring and observability

Shifting left on security

Test automation

Test data management

Trunk-based development

Version control

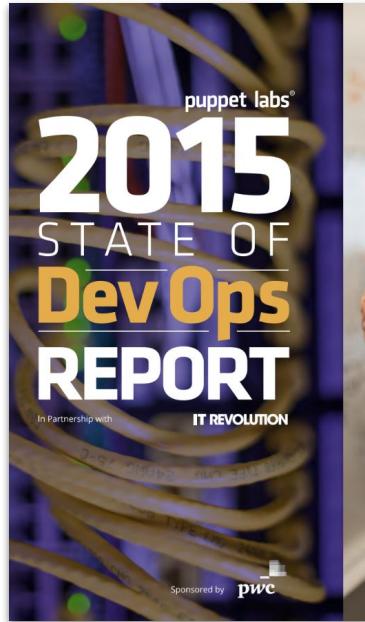
### PROCESS

Customer feedback

Documentation quality

Monitoring systems to inform business decisions

Proactive failure notification



High-performing teams are using **loosely coupled architectures** for both **new and brownfield systems**, including packaged software (COTS), embedded systems, user-installed and server-side systems.



If you want to achieve **higher IT performance**, start shifting to loosely coupled services — services that can be developed and released independently of each other — and **loosely coupled teams**, which are **empowered** to make changes.



Architectural approaches that enable loosely coupled architecture include the use of **bounded contexts** and **APIs** as a way to decouple large domains, resulting in smaller, more loosely coupled units.



A loosely coupled architecture is when delivery teams can **independently test, deploy, and change their systems on demand without depending on other teams** for additional support, services, resources, or approvals, and with less back-and-forth communication. This allows teams to **quickly deliver value**, but it requires orchestration at a higher level.

# 2022 ACCELERATE

## State of DevOps Report



Sponsored by



In the real world, loose coupling is **not restricted** to one architectural style; fundamentally, it's the ability to make a change in one part of the system, without that change **impacting other parts**. This allows organizations to divide up their work, so that **individual teams can make progress** without having to coordinate with other teams.

# 2022 ACCELERATE

## State of DevOps Report



Sponsored by



Teams that focus on building software with loosely-coupled architectures are in a **better position to perform strongly** across stability, reliability, and throughput. These teams are also more likely to **recommend their workplace** to a friend or colleague.

# 2022 ACCELERATE

## State of DevOps Report

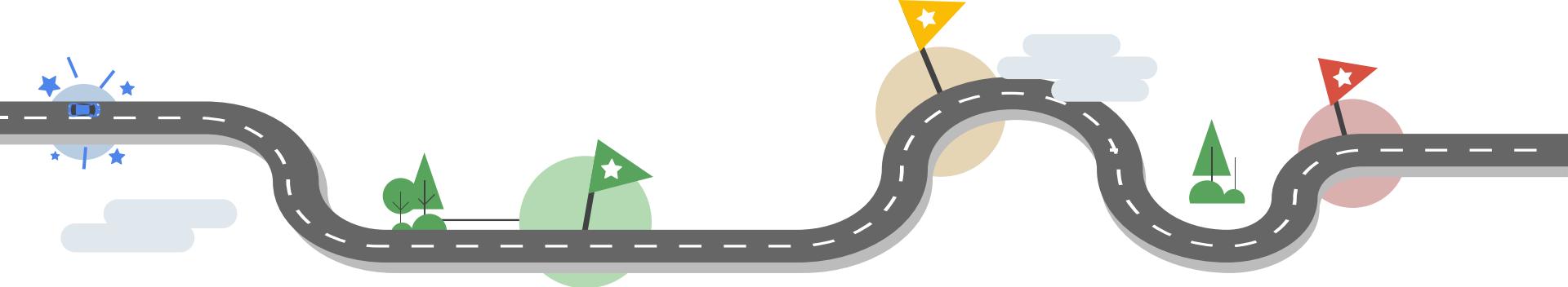


Sponsored by



**High performers** who meet **reliability targets** are:

**40% more likely** to have systems based on a **loosely-coupled architecture**



## Code Reviews



Lead time  
for changes

## Loosely Coupled Architecture



Deployment  
frequency



Change  
fail rate



Time to  
restore service

# 2014 STATE OF **DEVOPS** REPORT



Presented by



IT REVOLUTION PRESS

ThoughtWorks

Build **trust** with your counterparts on other teams. Building trust between teams is the **most important** thing you can do, and it must be built over time. Trust is built on kept promises, **open communication**, and behaving predictably even in **stressful** situations.

# 2014 STATE OF **DEVOPS** REPORT



Presented by



IT REVOLUTION PRESS

ThoughtWorks

Building **bridges** between teams will increase your understanding of the **challenges** at every point in the life cycle.



The benefit of **loosely coupled teams and services**:  
higher throughput and  
higher quality and stability.

Presented by



Google Cloud

Accelerate

# State of DevOps Report 2023



Premiere Sponsors



digital.ai



BROADCOM SOFTWARE  
Deloitte. SLEUTH  
Qarik

v. 2023-10

Improving all the technology,  
processes, and capabilities  
won't help you succeed  
without a **healthy culture**

---

# Culture



## Westrum

How an organization tends to respond to problems and opportunities. There are three types of culture: generative, bureaucratic, and pathological

## Flexibility

How, where, and when a person works on tasks

## Knowledge sharing

How ideas and information spread across an organization. Team members answer questions once, and make the information available to others. People don't have to wait for answers

# Culture



## User-centrism

Understanding and incorporating users' needs and goals to make products and services better



## Stability & security

Providing a stable environment for employees and minimizing worries about job security



## Work distribution

Formal processes to help employees distribute tasks equitably within a team

# Culture helps individuals get where they want to be

Aspect of culture	Key outcomes		
	Burnout*	Job satisfaction	Productivity
Generative culture	↔	↑↑	↑↑
Organization stability	↔	↑↑	↑
Job security	↔	↑	↑
Flexibility	↓	↑	↑
Knowledge sharing	↔	↑	↑
User-Centrism	↓	↑↑	↑↑
Work distribution	⊖	↑	↑

\* Notice how the color scheme is flipped for burnout (here, green means down). That's because reducing burnout is a good thing!

# Culture gets orgs and teams to key outcomes

Aspect of culture	Key outcomes			
	Team performance	Organizational performance	Software delivery performance	Operational performance
Generative culture	↑↑	↑↑	↑↑	↑↑
Organization stability	↑	↑↑	↓	⊖
Job security	↑	⊖	↑	↑
Flexibility	↑	↑	↑↑	↑
Knowledge sharing	↑	↑	↑↑	↑↑
User-Centrism	↑↑	↑↑	↑	↑↑
Work distribution	↑↑	↑↑	↓↓	⊖



# Transformational Leadership

Establishing & supporting generative & **high-trust** cultural norms.

- Supporting **team experimentation** and innovation, to create and implement better products faster.
- Working across organizational silos to achieve **strategic alignment**.



**Transformational leaders** encourage their teams to work towards a **common goal** through their vision, values, communication, example-setting, and their evident caring about their followers' personal needs.

Transformational leaders focus on getting followers to **identify with the organization and engage in support of organizational objectives**.



The importance of **future value creation** should not be understated. Using the time recovered from excess rework and downtime to develop new products and features or implement process improvements will continue to **pay off for years to come.**

## The ROI of DevOps Transformation



Retaining existing talent is more **cost-effective**, preserves institutional knowledge, and gives organizations an **advantage** by having a **strong technical workforce** that is engaged and continuing to learn.

## The ROI of DevOps Transformation



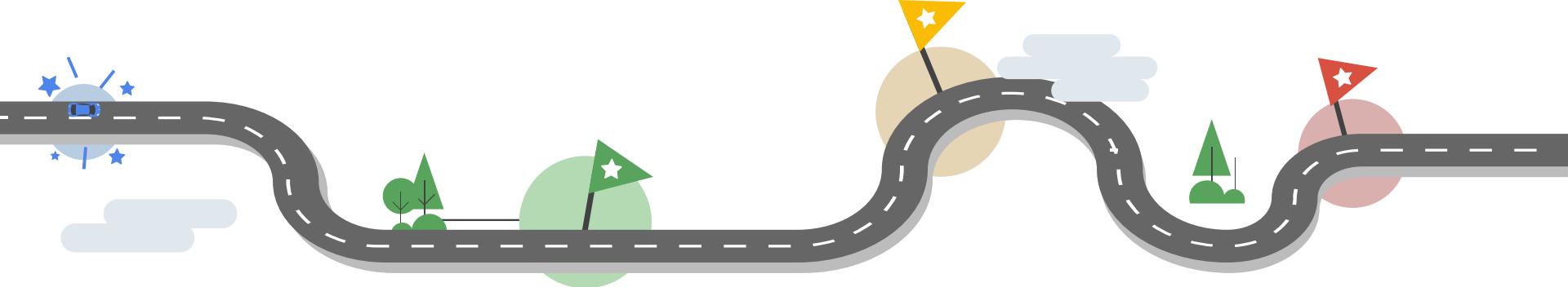
The best, most innovative companies undertake their technology transformations with an eye to the **value they can deliver to their customers** and the business in addition to the cost savings and efficiencies they can realize.

# The ROI of DevOps Transformation



\$100M product portfolio business size	Elite IT performers	High IT performers	Medium IT performers	Low IT performers
<b>Large organization that relies on in-house software</b> (8,500 engineers)	\$18.2M value of rework recovered + \$48.7M value lost from new features + \$13.7M cost of downtime = \$80.6M return	\$27.3M value of rework recovered + \$5.2M value lost from new features + \$31.4M cost of downtime = \$63.9M return	\$36.5M value of rework recovered + \$1.6M value lost from new features + \$9.6M cost of downtime = \$47.7M return	\$36.5M value of rework recovered + \$267K value lost from new features + \$222.6M cost of downtime = \$259.3M return
<b>Medium-to-large technical organization</b> (2,000 engineers)	\$4.3M value of rework recovered + \$19.5M value lost from new features + \$13.7M cost of downtime = \$37.4M return	\$6.4 cost of rework + \$2M value lost from new features + \$31.4M cost of downtime = \$39.9M return	\$8.6M cost of rework + \$640K value lost from new features + \$9.6M cost of downtime = \$18.8M return	\$8.6M cost of rework + \$107K value lost from new features + \$222.6M cost of downtime = \$231.3M return
<b>Small to medium businesses and non-technical enterprises</b> (250 engineers)	\$536K value of rework recovered + \$2.4M value lost from new features + \$13.7M cost of downtime = \$16.7M return	\$804K value of rework recovered + \$260K value lost from new features + \$31.4M cost of downtime = \$32.4M return	\$1M value of rework recovered + \$80K value lost from new features + \$9.6M cost of downtime = \$10.8M return	\$1M value of rework recovered + \$13.3K value lost from new features + \$222.6M cost of downtime = \$223.7M return





## Code Reviews



Lead time  
for changes

## Loosely Coupled Architecture



Deployment  
frequency

## Documentation



Change  
fail rate



Time to  
restore service

# State of DevOps 2021



Google Cloud

Teams with quality documentation are:

3.8x

More likely to implement security best practices

3.4x

More likely to implement SRE best practices

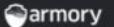
2.4x

More likely to reach reliability targets

2.5x

More likely to fully leverage the cloud

Sponsored by



**Deloitte.**



PagerDuty



Google Cloud

# State of DevOps 2021



Google Cloud

Sponsored by

armory

circleci

redgate

Deloitte.

sysdig

GitLab

Liquibase

PagerDuty

CD.FOUNDATION

## Quality Documentation

helps readers **accomplish** their goals

is **accurate, up-to-date,** and **comprehensive**

is **findable, well organized,** and **clear**

# State of DevOps 2021



Google Cloud

Sponsored by

armory

circleci

redgate

Deloitte.

sysdig

GitLab

Liquibase

PagerDuty

CD.FOUNDATION

Teams with high quality documentation are **better able to implement technical best practices** and perform better as a whole.

# 2022 ACCELERATE

## State of DevOps Report



Sponsored by



Employees at organizations with a **generative culture** are more likely to belong to stable teams, produce **higher-quality documentation**, and spend most of their time **engaged in meaningful work**.

Presented by



Google Cloud

Accelerate

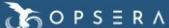
# State of DevOps Report 2023



## Premiere Sponsors



digital.ai



BROADCOM  
SOFTWARE

SLEUTH

Deloitte.

Qarik

v. 2023-10

**Quality documentation** not only helps establish technical capabilities, but helps them matter



The quality of written content that people in the organization create and use in their daily work

Presented by



Google Cloud

Accelerate

# State of DevOps Report 2023



Premiere Sponsors



digital.ai



BROADCOM SOFTWARE  
Deloitte. SLEUTH  
Qarik

v. 2023-10

Documentation predicts  
**productive** and **happy**  
**individuals** and  
**organizations**



## Code Reviews



Lead time  
for changes

## Loosely Coupled Architecture



Deployment  
frequency

## Documentation



Change  
fail rate

## Flexible Architecture



Time to  
restore service

# State of DevOps 2021



Google Cloud

What really matters is **how teams implement** their cloud services, not just that they are using cloud technologies.

- On-demand self-service
- Broad network access
- Resource pooling
- Rapid elasticity
- Measured service

2018

## Accelerate: State of DevOps

Strategies for a New Economy

Presented by  
**DORA**  
DEVOPS RESEARCH & ASSESSMENT

Diamond Sponsor  
**Google Cloud**

Gold Sponsors



Teams that leverage all of cloud computing's essential characteristics are **23 times** more likely to be high performers.

Presented by



Google Cloud

# Accelerate State of DevOps Report 2023



## Premiere Sponsors



digital.ai



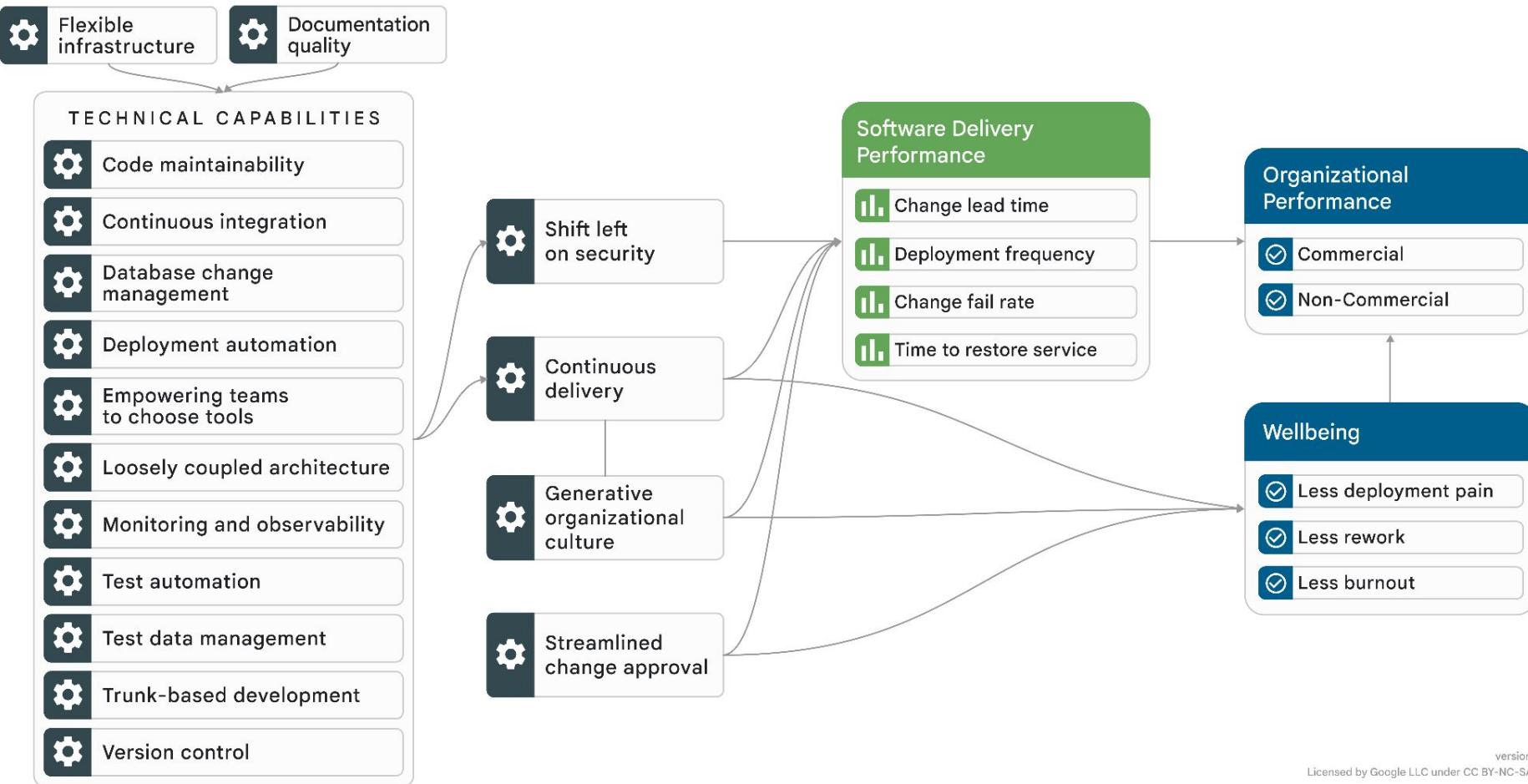
BROADCOM SOFTWARE  
Deloitte. SLEUTH  
Qarik

v. 2023-10



---

Flexible infrastructures lead to  
**30% higher** organizational  
performance





To scale deployments per day per developer, we need to **focus on all the factors** that predict high IT performance: a goal-oriented generative culture, a modular architecture, the engineering practices that enable continuous delivery, and effective leadership.



Teams implementing **continuous delivery** practices and taking an **experimental** approach to product development will **build better products**, and also feel more connected to the rest of their organization.

# 2022 ACCELERATE

## State of DevOps Report



Sponsored by



Technical capabilities **build on one another**

# Software delivery performance at a glance

Aspect of Software delivery performance	Elite	High	Medium	Low
<b>Change lead time</b>  For the primary application or service you work on, what is your lead time for changes (i.e., how long does it take to go from code committed to code successfully running in production)?	Less than one day	Between one day and one week	Between one week and one month	Between one week and one month
<b>Deployment frequency</b>  For the primary application or service you work on, how often does your organization deploy code to production or release it to end users?	On-demand (multiple deploys per day)	Between once per day and once per week	Between once per week and once per month	Between once per week and once per month
<b>Change failure rate</b>  For the primary application or service you work on, what percentage of changes to production or released to users result in degraded service (e.g., lead to service impairment or service outage) and subsequently require remediation (e.g., require a hotfix, rollback, fix forward, patch)?	5%	10%	15%	64%
<b>Failed deployment recovery time</b>  For the primary application or service you work on, how long does it generally take to restore service after a change to production or release to users results in degraded service (for example, lead to service impairment or service outage) and subsequently require remediation (for example, require a hotfix, rollback, fix forward, or patch)?	Less than one hour	Less than one day	Between one day and one week	Between one month and six months



# Elite



Lead time  
for changes



Deployment  
frequency



Change  
fail rate



Time to  
restore service



Elite



Lead time  
for changes



Deployment  
frequency



Change  
fail rate



Time to  
restore service



Loosely coupled architecture

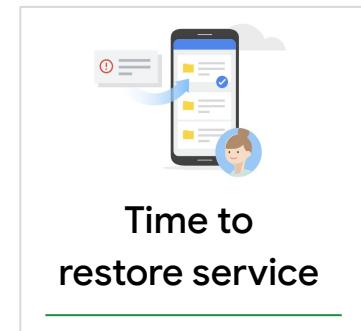
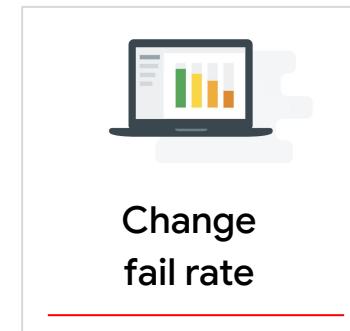
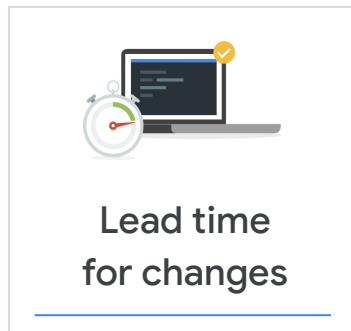
Faster Code Reviews

Flexible Infrastructure

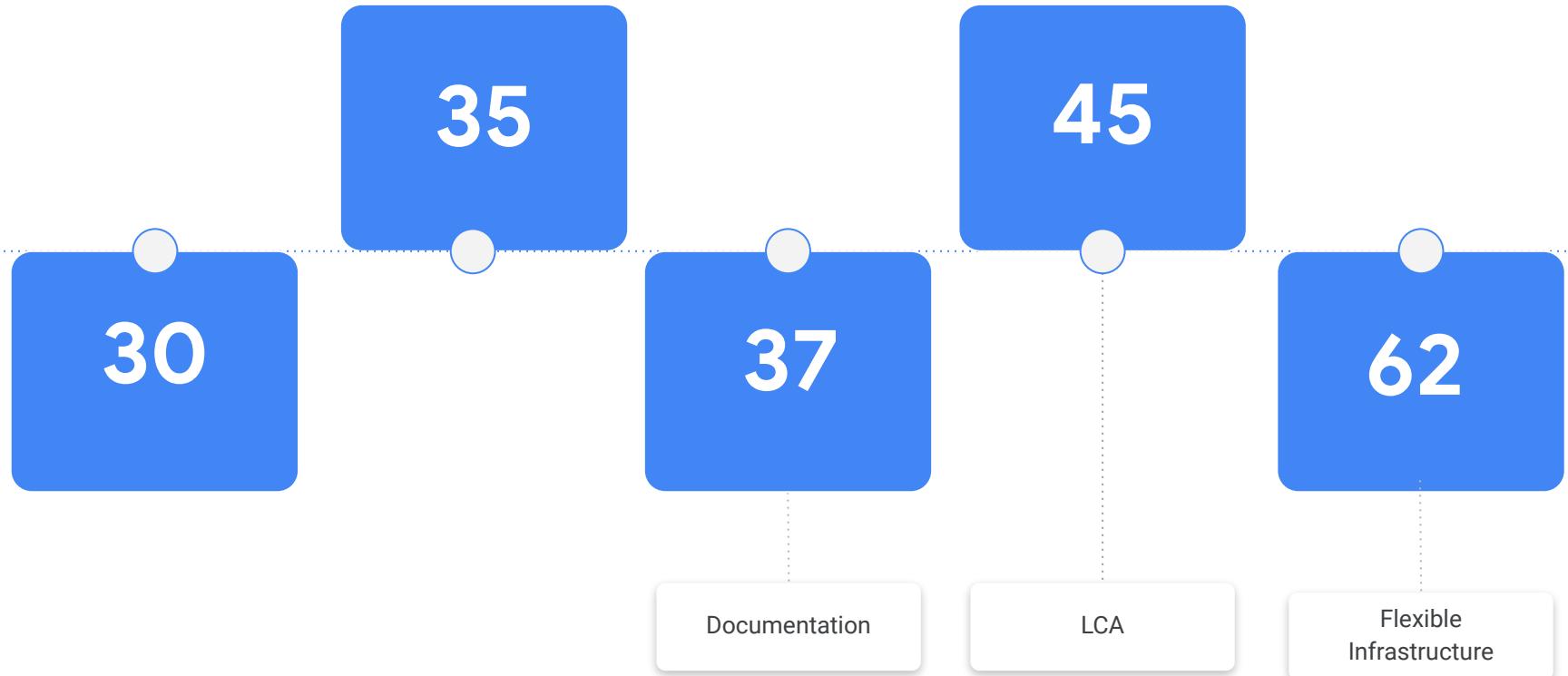
Documentation



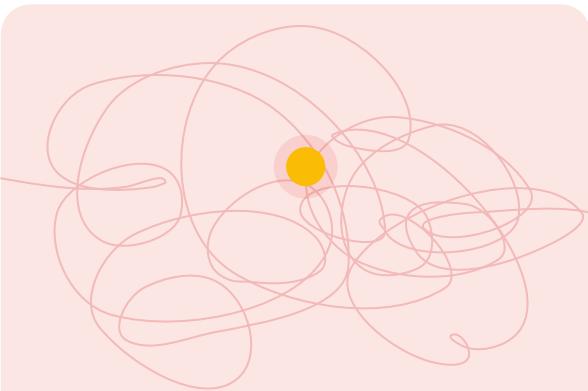
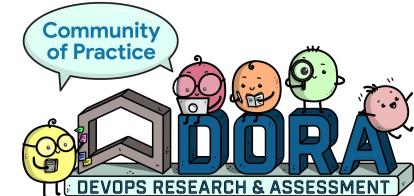
**Employees can trust that  
the employee directory is  
accurate**



# Employee Productivity Team DORA Improvements

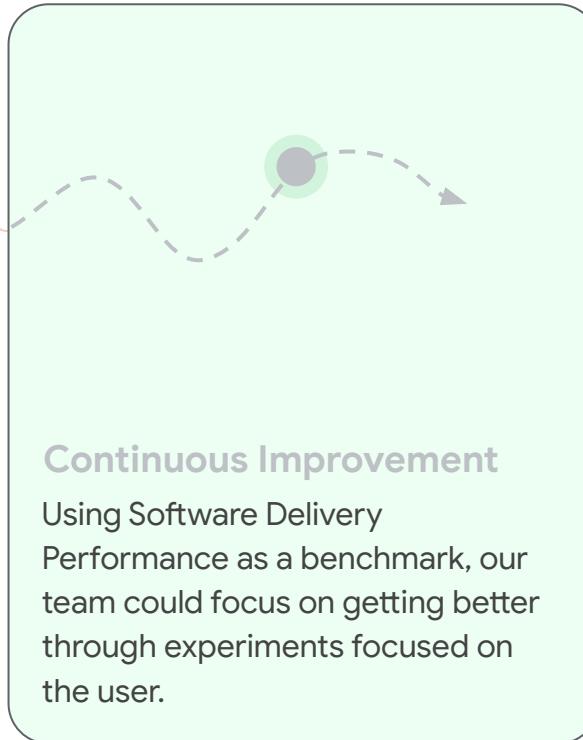


# Getting Better at Getting Better



## Before Continuous Improvement Culture

Team struggled to align, and learn from each other. Our users felt the pain from our chaos.



## Continuous Improvement

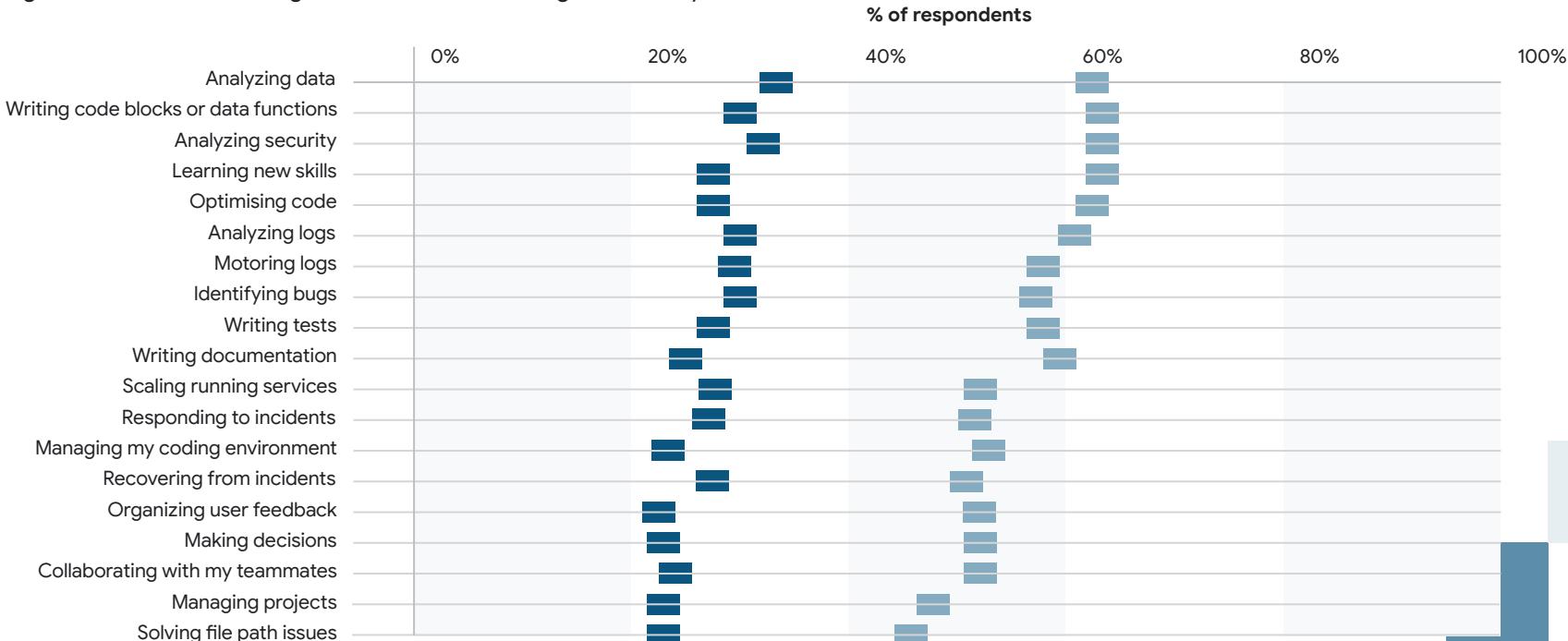
Using Software Delivery Performance as a benchmark, our team could focus on getting better through experiments focused on the user.

# AI contribution to technical tasks

For the primary application or service you work on, how important is the role of Artificial Intelligence (AI) in contributing to each of the following tasks today?"

Did NOT select  
"Not at all important"

Selected  
"Extremely Important"



\*Interval provided for each graph represents 89% credibility interval.  
Provided to show inherent uncertainty in our estimates.

Presented by



Google Cloud

Accelerate

# State of DevOps Report 2023



## Premiere Sponsors



digital.ai



BROADCOM SOFTWARE  
Deloitte. SLEUTH  
Qarik

v. 2023-10

---

Anticipate that it will take some time for **AI-powered** tools to come into widespread and coordinated use in industry

# Key insights from 2023



## Healthy culture

All the tech, process, and capabilities wouldn't get you far with unhealthy culture



## User centricity

Fast, stable, and reliable is great. But if you don't know who you build for the org, the team, and the employees will struggle



## Quality documentation

Quality documentation not only leads to favorable outcomes and the establishment of technical capabilities, but provides the grounds for technical capabilities to have an impact



## Flexible infrastructure

Creating a flexible infrastructures how cloud computing differentiates itself

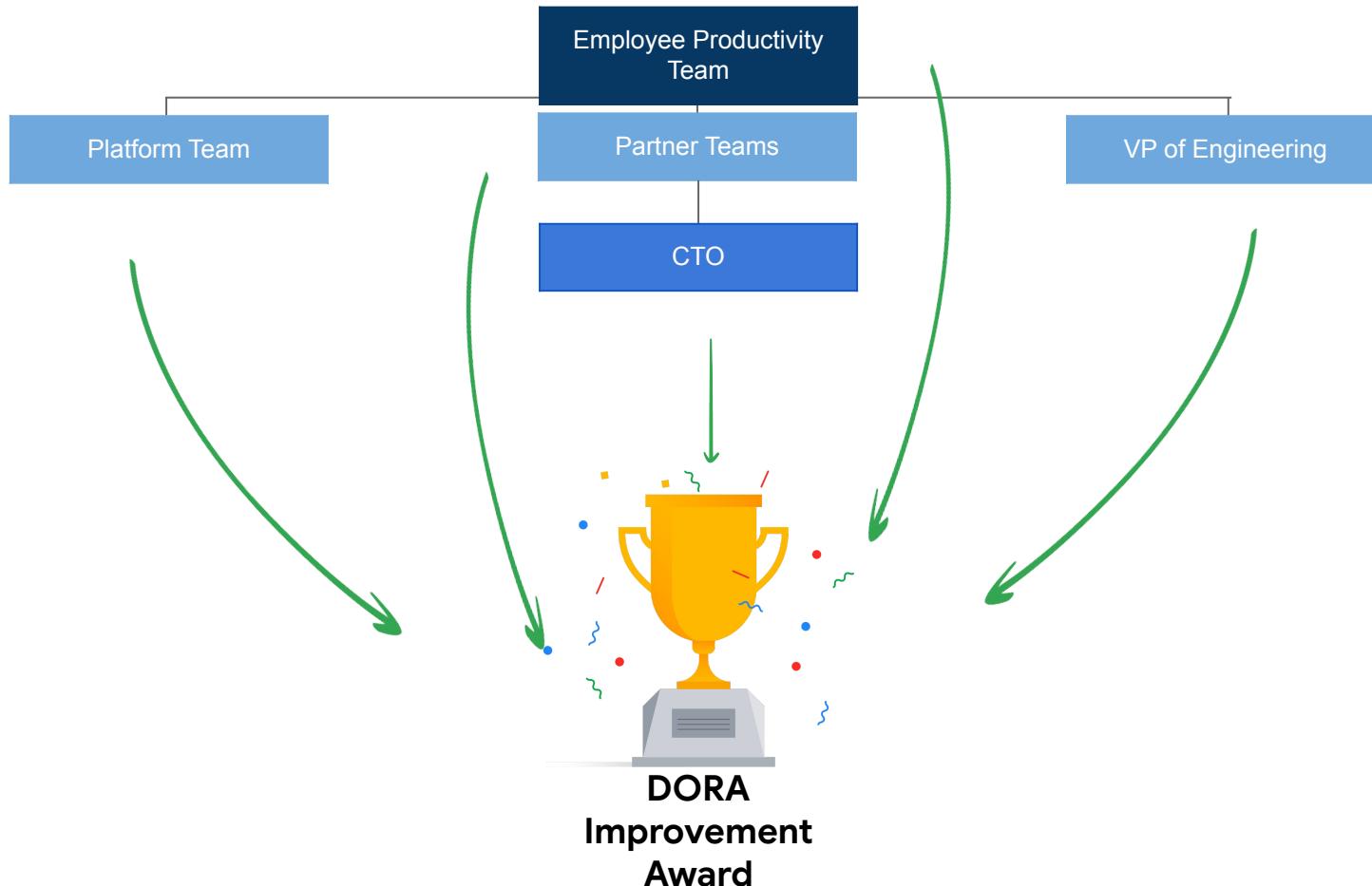


## Underrepresented groups

Those who identified as women or self-described their gender and people who are underrepresented have higher levels of burnout due to more toil and less-recognized work



Team leaders **shape** the culture according to their own proclivities, by **creating incentive structures** that **reward certain behaviors**. These incentive structures also affect how team members process and share information, cooperate and collaborate.



Presented by



Google Cloud

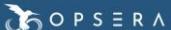
# Accelerate State of DevOps Report 2023



## Premiere Sponsors



digital.ai



BROADCOM  
SOFTWARE



Deloitte.

Qarik

v. 2023-10

## How can you help?



Download the report at

**bit.ly/dora-sodr**

