State of DevOps 2021

Presented by

nagarro

Content

Executive summary

The year of DevOps 2021

Our very first **State of DevOps Report** this year was based on our client landscape. You were all invited to give us an insight of your current status-quo and your current DevOps journey.

We asked many domain experts and peers to share their experiences with us and as a result we gained some very interesting insights.

The focus of this year's evaluation was mainly in the areas of:

- **Culture**
- **Organizational and team structures**
- **Architecture**
- **Automation**
- **Testing practices**
- **Cloud**

In the first section, we will give you an overview of how we performed measurements and how we reached our findings.

If you are interested in the tool landscape, we recommend that you take a look at the appendix where we describe the different working environments.

To all readers who have joined us but had not participated in the survey, the use of "you" allowed us to communicate the results better. However, maybe you will still find yourself within this analysis.

We hope you will enjoy reading our insights.

Key findings

1 The industry needs a DevOps kick start.

With 49%, almost half of the respondents have not yet started their DevOps evolution or are still experiencing difficulties getting going.

2 Influence your culture to be successful.

While agile methodologies are part of the cultural toolbox, other parts of the DevOps mindset are reaching widespread adoption.

3 Architectural and automation outcomes are important.

It is essential that teams achieve architectural outcomes to ensure that the quality required for a complete and reliable automation is met.

4 Cloud is a differentiator and drives high performance.

High-performing teams have a high cloud adoption rate, but we still see a lot of untapped potential to improve the way of working with cloud services.

Introduction to DevOps

DevOps came about as an effort to break down silos between development and operations. This expanded, as the approach developed, to include information security and business leadership.

Now, DevOps is an organizational culture that combines lean, agile principles with technical tools and practices.

"In empirical and scientific research, in recent years, in the DevOps community especially from DORA/ Google or Puppet, it has been found that companies that can respond as flexible as possible to the market by emphasizing data-driven business decisions, foster a high level of trust and autonomy, and offer the possibility to run several experiments per year, are the more successful ones."

Jürgen Pointinger,
Quality DevOps Practice Lead at Nagarro

Introduction to Nagarro

Nagarro is a global digital engineering leader with a full-service offering. We conducted this survey to better understand what factors influence software delivery and organizational performance. This survey allows us to better serve you as a client and help drive your business forward.

Our report will allow you to establish internal benchmarks, compare yourself to established good practices, and locate areas for improvement within your organization.

- **The questionnaire contained 63 multiple-choice and rating questions.**
- **This survey took approximately 30 minutes to complete.**
- **All data was collected anonymously. Answers were not linked to your email. All information kept confidential.**

Thank you for participating.

Your Nagarro team

Acknowledgments

Our research is informed by our work and conversations with the DevOps, Agile, and broader technical community; many thanks to all of our peers and colleagues who so openly share their experiences, stories, success, challenges, and failures.

The authors would like to thank several people for their input and guidance on the report this year. All acknowledgments are listed alphabetically by type of contribution.

We would like to thank **Samanta Mika** and **Thomas Goldberger** for helping us develop the questions for this survey, who set the foundation for the success of the report.

A big thanks to **Elisabeth Schweighofer** and **Susanne Soumelidis** for the numerous reviews of the survey questions and preparation of the snowball sampling. No matter which areas it concerns, you always have good ideas how we can make something even better.

We would also like to thank **Martin Schweinberger, Stefan Schwartz** and the entire **Global Delivery Management** team who helped us bring the survey to our clients. They have been tirelessly talking to us so that we can provide a report this year.

Our work is not possible without the careful review of subject matter experts; for their time on supplementary topic review and input on measurement items, we are extremely grateful. Many thanks to **Alexander Birsak** and **Manisha Mittal**.

Thanks to **Peter Hammer** for the linguistic review, to **Gil Rodrigues** for the graphical editing, and to **Benjamin Chemelski** for designing this report.

Finally, we would like to thank all **participants** in this year's survey. Who took their time to answer all our questions. Without you, this report would not exist. Thank you for that.


Do we already know DevOps?
nagarro

One to rule them all

...maybe not

DevOps first became a hot topic in 2008. After that, for many, it was all about collaboration between development and operations.

Since then, DevOps has crossed the chasm. DevOps is widely recognized as a best-practice model for agile development, and technology experts at all levels can easily provide their own definitions.

But now that it has become mainstream, we're catching on that solutions or approaches related to DevOps are interchangeable. There is no silver bullet for all malpractices in development. We are operating in a complex and emergent environment.

To understand what DevOps is and how DevOps can be used to your organization's advantage, it is important to first understand what DevOps is not.

DevOps is not

- the simple combination of development and operations teams
- a product
- a specification
- a job title
- a tool
- a separate team
- a one-size-fits-all solution
- cloud
- automation or infrastructure as code

The history reminds us, that DevOps is ...

- for practitioners and by practitioners
- an experience-based movement
- decentralized and open to all

DevOps is here to stay

Market trends

2021 – Diffusion of DevOps innovation

DevOps is very versatile and covers the entire software delivery life cycle. Here is an overview of the current DevOps trends on the market and in the DevOps community.

Innovators

- FinOps
- Policy as Code
- Cross-cloud / Cloud-native hybrid approaches
- Cross-cloud uniform infrastructure automation
- Data Mesh
- No copy data sharing
- Sustainability accounting

Early adopters

- AI/MLOps
- Edge computing
- Service messages and programmatic edge
- Active-active global DB ops
- Serverless databases
- Fullstack tracing
- Continuous testing
- Chaos engineering practices
- ChatOps
- DataOps
- Developer experience “DevEx”
- Team topologies
- Measuring Performance (Accelerate)
- Transformational leadership (Accelerate)

Early majority

- Cloud FaaS/BaaS
- Observability
- Continuous delivery
- Feature flags & Blue/ Green deployments
- Shift-left on Security/ InfoSec
- GitOps
- CD for mobile
- Client error tracking
- [Enterprise] DevOps toolchain
- Site Reliability Engineering (SRE)
- Blameless Post-mortems

Late majority

- Continuous integration best practices
- Monitoring and logging
- Centralized log aggregation
- Infrastructure as code
- Containers
- Container orchestration
- Software-defined networks (SDN)
- Continuous integration tooling
- Self-service platforms
- DevOps toolchain
- General DevOps

How we measure performance?

nagarro

Measuring DevOps Performance

Our measurement and clusters are an interpretation of DORA's published industry-standard key metrics*.

Since we believe that security plays an accelerating factor in software delivery, we have extended the metrics and reinterpreted them for ourselves.

Just after finalizing our report this year, DORA's new Accelerate State of DevOps report 2021** was released. For this reason, those insights were not included in our work.

* Using the four keys to measure your DevOps performance
** DORA 2021 Accelerate State of DevOps report

Measuring DevOps Performance

No trade-offs

The industry-standard four key metrics for software development and delivery are driving organizational performance in technology transformations. Software delivery performance (SDP) is comprised of speed and stability, and both are possible without trade-offs.

The improvement of the four key metrics are led by, among others, technical practices, cloud adoption, organizational practices, such as approval processes, and culture. Each metric creates a virtuous circle and directs teams toward continuous improvement.

Speed

- Deployment frequency – How often an organization successfully releases to production
- Lead time for changes – The amount of time it takes a commit to get into production

Stability

- Change failure rate - The percentage of deployments causing a failure in production
- Time to restore service - How long it takes an organization to recover from a failure in production

These four key metrics are a simple and yet powerful tool to help leaders and teams focus on measuring and improving what matters. A good place to start is to instrument the build pipelines so you can capture the four key metrics and make the software delivery value stream visible.

High-performing teams create high-performing organizations.

DevOps performance clusters*

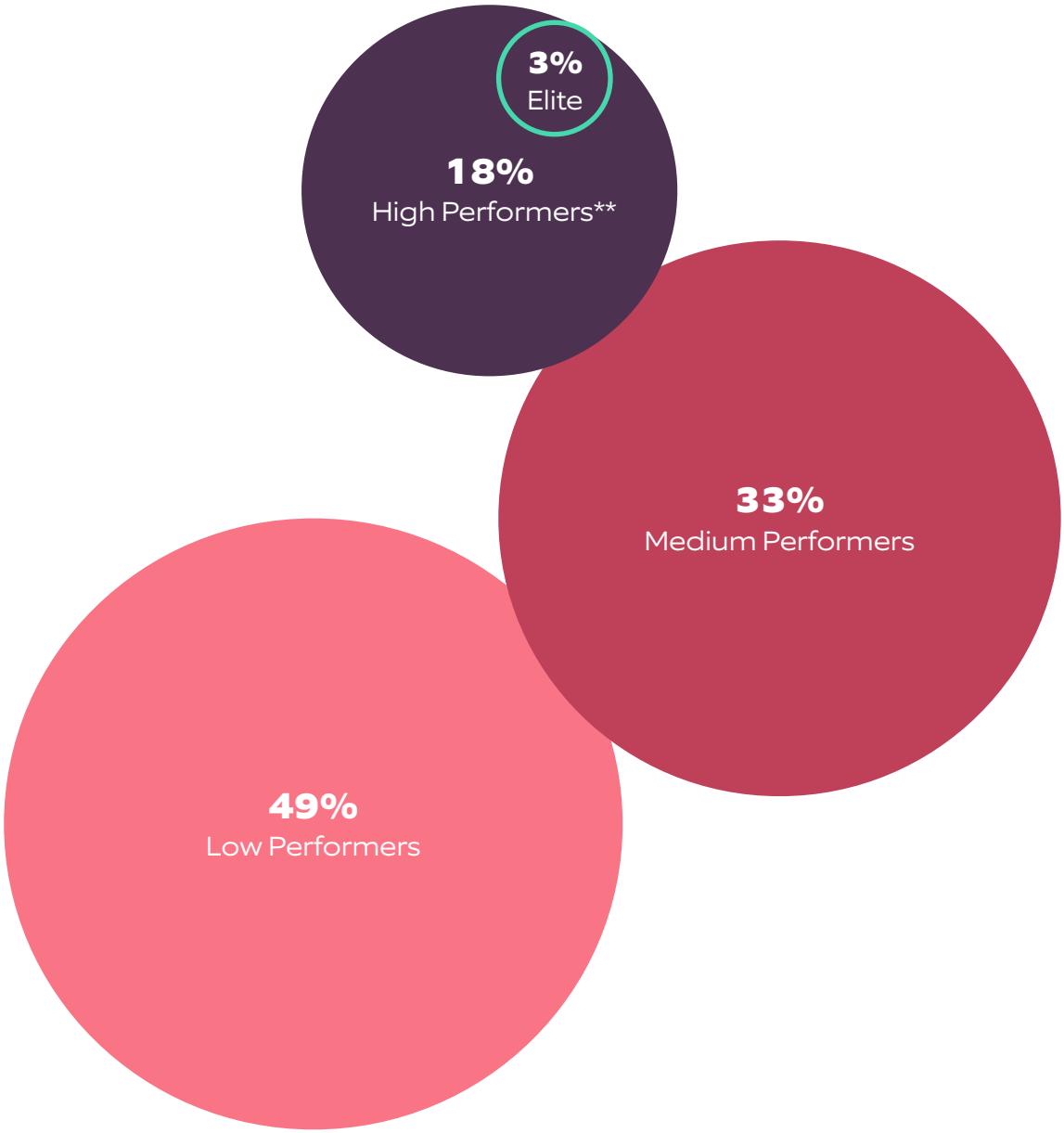
What performance profile is the most fitting for you?

Metric	Elite	High	Medium	Low
Deployment frequency	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 month and once every six months
Lead time for changes	Less than one day	Between one day and one week	Between one week and one month	Between one month and six months
Time to restore service	Less than one hour	Less than one day	Between one day and one week**	Between one week and one month
Change failure rate	0-15%	0-15%	0-15%	46-60%
Critical security vulnerability rate**	Less than one day	Between one day and one week	Between one week and one month	More than one month

* Performance indications and clusters derive from the State of DevOps Report 2019 published by DORA.

** Not included or different to the DORA performance clusters

Your DevOps performance profile*



* Performance indications and clusters derive from the State of DevOps Report 2019 published by DORA.

** High- and elite-perform teams are considered as one cluster. We see organizations that have already reached the elite level, however, due to the small quantity, a separate analysis is not feasible.

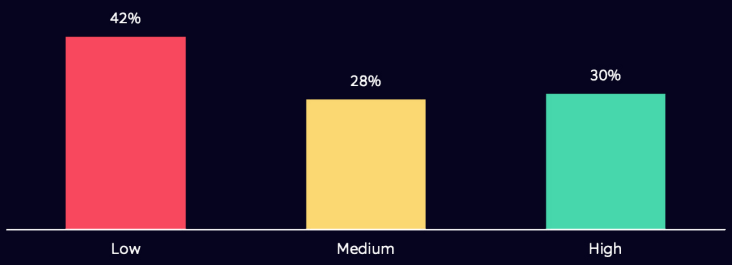
Speed

Deployment frequency

How often an organization successfully releases to production

The deployment frequency is how often a product can be released to production, or at least to a pre-production like environment. It indicates how often you can run experiments and shows how flexible you are to react to the market needs, and how responsive you are to your customers needs. Deployment frequency enforces your teams to improve their processes and automation.

A large proportion, specifically 42%, of the respondents have not yet started their DevOps evolution or are still experiencing difficulties at this stage. The second largest part with 28%, however, is already in the midst of evolution and has taken the first steps towards an improved deployment strategy.

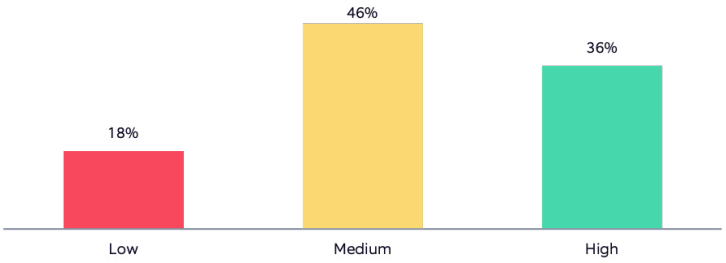


Lead time for change

The amount of time it takes a commit to get into production

To reduce lead time, you reduce wasteful activities which, in turn, lets you deploy more frequently.

More than twice as many respondents are currently facing lead times between one week and one month (medium). Even 18% are struggling with lead times of between one month and six months (low). There is still a lot of potential for improvement.



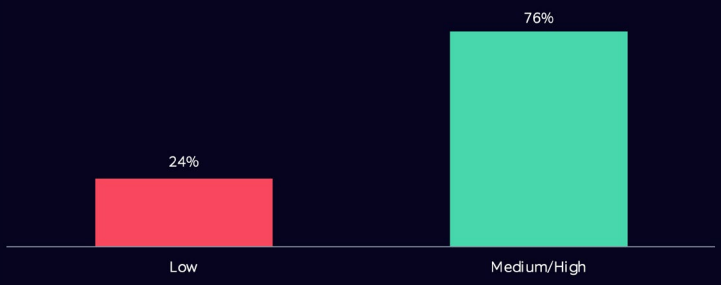
Stability

Change fail rate

The percentage of deployments causing a failure in production

The change fail rate shows how often a software deployment leads to a failure in production. From experience, this is a metric that reveals the often complicated collaboration between Dev and Ops visible. Development wants to deliver new features quickly and frequently, while Operations is concerned with keeping the systems as stable as possible - a conflict of interest arises.

Around a quarter of respondents struggle with a change fail rate higher than 15% (low). Contrary to the DORA report, we have classified all responses with a CFR greater than 15% as low-performing teams. However, this has no effect on the clustering.

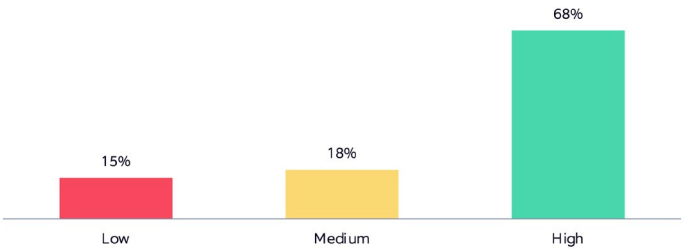


Time to restore service

How long it takes an organization to recover from a failure in production

The speed at which you recover from a failure is improved through better practices, automation and monitoring, reducing the frequency of failures.

We are thrilled to see that you are investing in service recovery right away and that it is an essential part of your software delivery process. We are talking about more than 65% of organizations in the high-performing clusters being able to restore their services within a day.

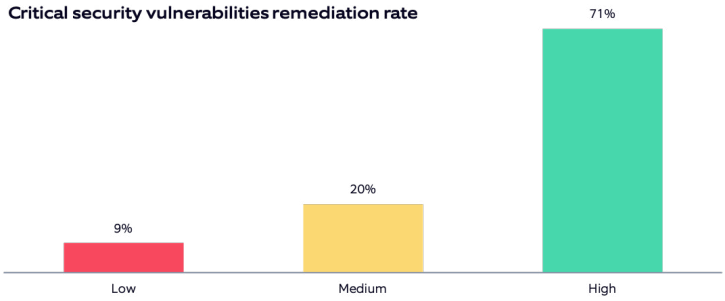


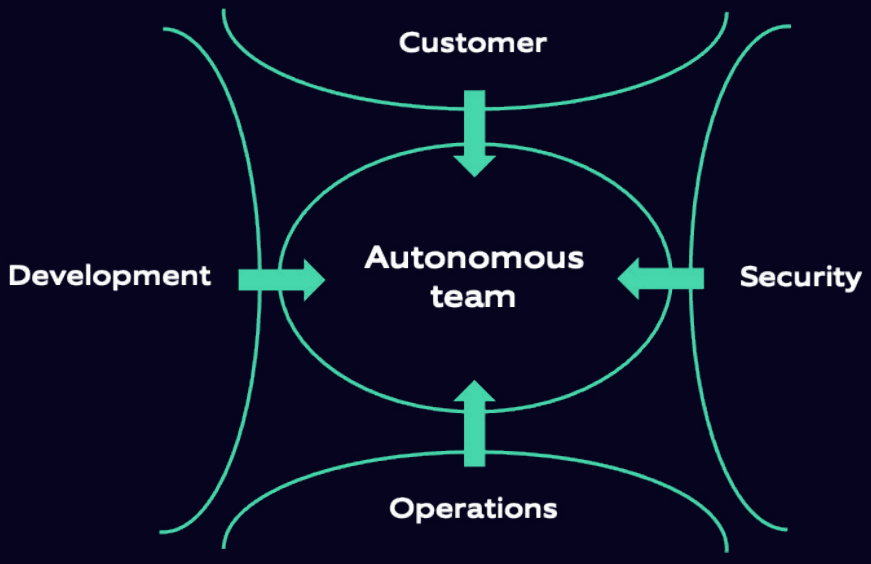
Security

A well-known silo

In our discussions with you, we hear about DevSecOps frequently. It puts the security aspects and particularly Security Information at the heart of actions in software deployment and takes security-related matters closer to the delivery team.

For this reason, we wanted to know how much you are currently struggling with vulnerabilities in your system and how quickly you can react. We identified security as one of the key aspects for deployments and see it as a critical factor for future implementations.





The industry needs a DevOps kickstart.

1

The survey reveals that more than two thirds of respondents are still at the beginning or in the middle of a DevOps journey*.

Almost half are still struggling to get started or have not started at all.

This led us to take a closer look at startup difficulties and make recommendations on how to begin a DevOps journey.

* We considered responses that are classified as "low-performing" and "medium-performing" as being at the beginning or in the middle of a DevOps journey – although that does not necessarily mean that they are currently in an ongoing process of transformation

Delivering software quality, reliably, and safely is at the heart of technology transformation and organizational performance.

Starter Kit for enabling Enterprise DevOps

In an emergent and complex environment, we must have the ability to respond quickly to market demands. If we consider the figures describing the profiles of delivery performance, we can see that at least the technology-centric aspects indicate many organizations have some catching up to do.

For organizations looking for guidance on how to improve, we point to the only true path forward:

Start with the basics, then adopt a continuous improvement mindset by identifying your unique constraints (or set of constraints).

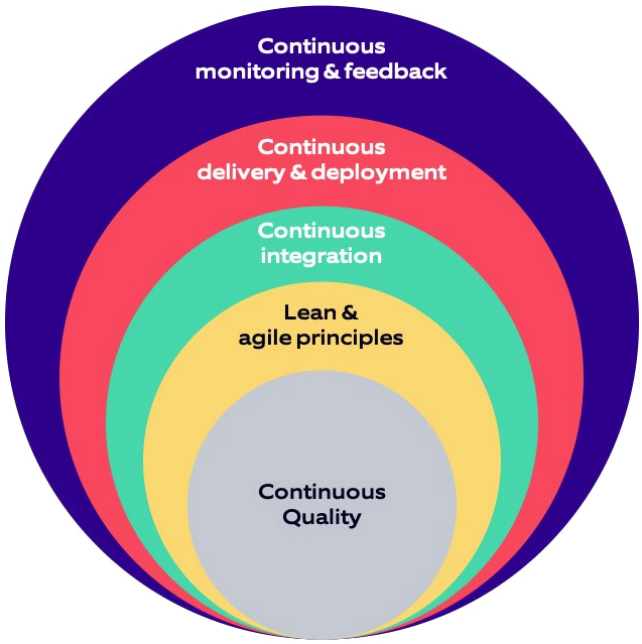
1. Never settle, continuously improve your ways of working.
2. Leaders and teams agree upon and communicate measurable outcomes; Teams discover how to achieve them.
3. Achieve large-scale change, once those limitations are no longer holding you back, iteratively and incrementally, by repeating the process.

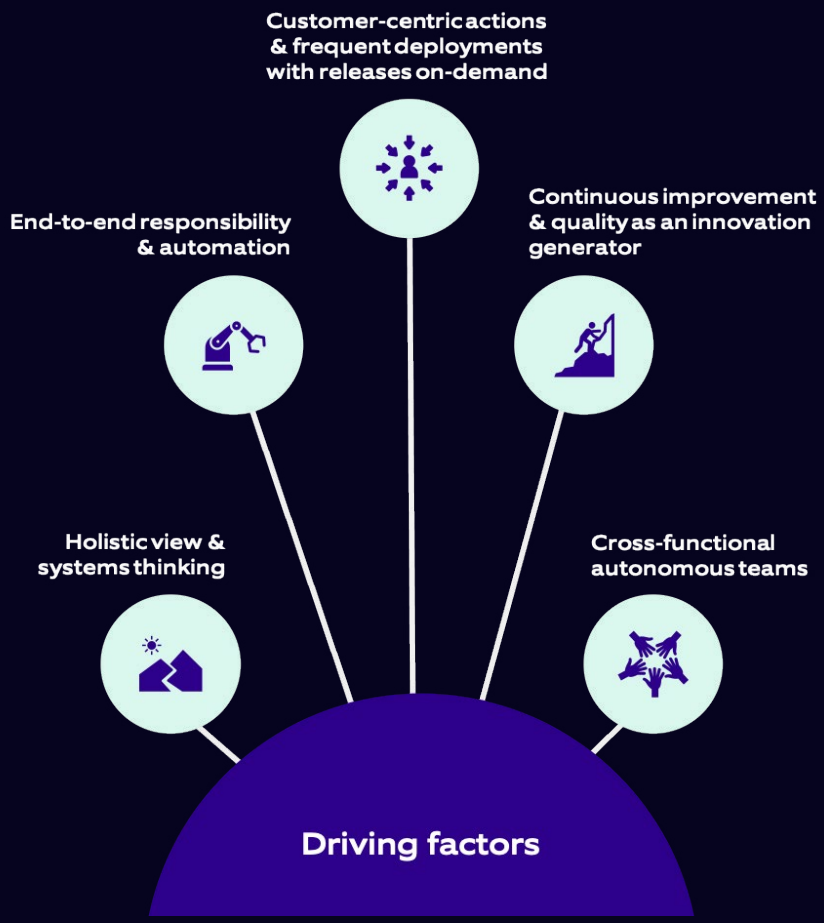
What drives organizational performance?

Enterprise DevOps is not easy to achieve. There is no such thing as one-size-fits-all for organizational DevOps. There is no silver bullet. There is no one way, no DevOps-in-a-box that optimizes outcomes in all contexts.

There is a need to focus on the benefit hypothesis and fast feedback to maintain flexibility and pivot to achieve the desired outcomes optimally.

The following principles are in the focus of our change:





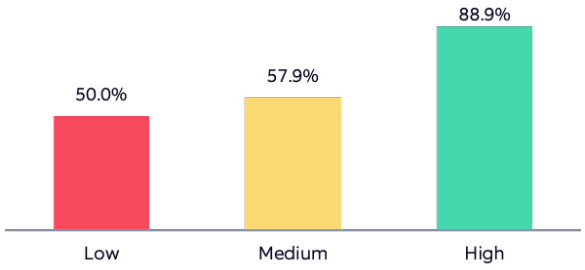
Foster developer experience (DevEx)

Improve your performances

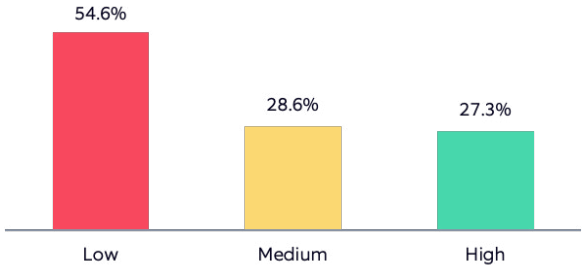
People have a limited velocity to unlearn and re-learn. The pace of change cannot be forced, it can only be fostered. Knowing this, we can take steps to encourage the development of technical excellence fostering a good developer experience to focus on the creation of continuous added value.*

In 2019 DORA** found that key capabilities significantly impact software delivery, operational performance, and organizational performance.

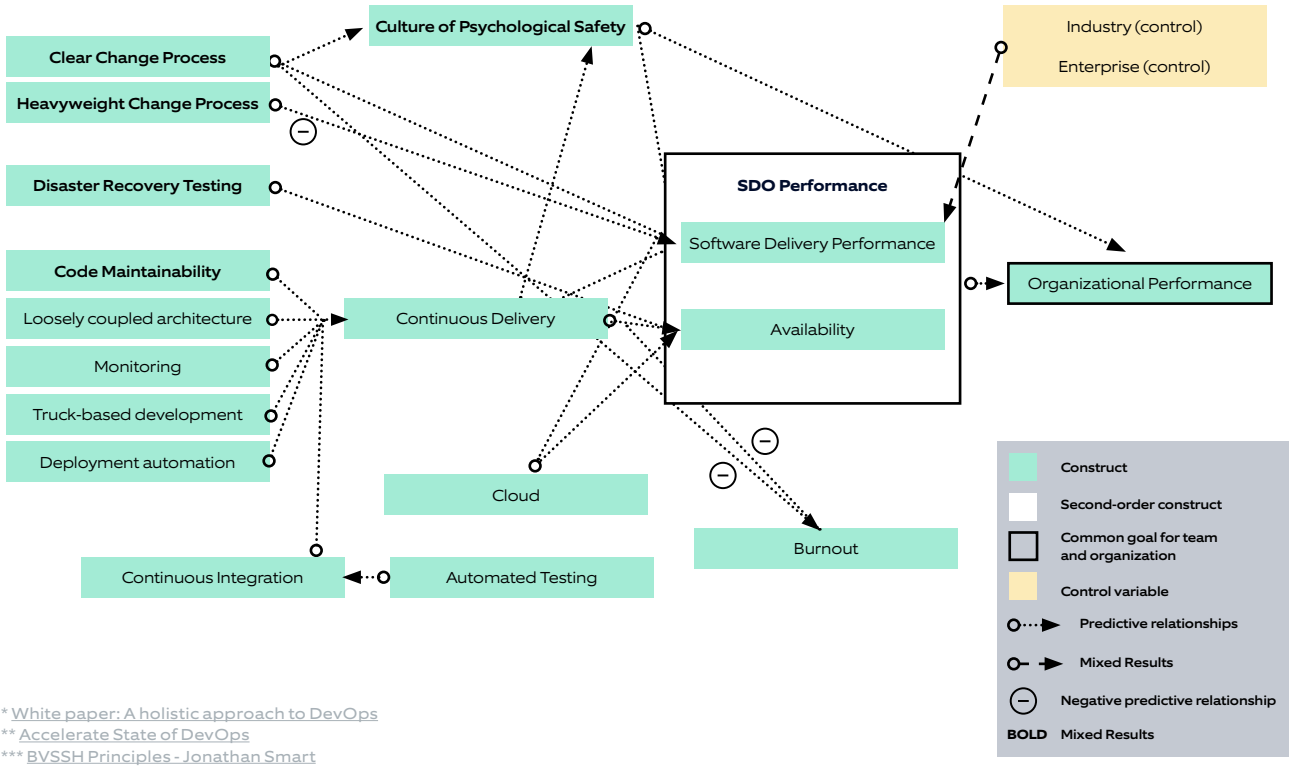
They provide predictive guidelines to improve your technology delivery and deliver value. Do not make the mistake of trying to do everything all at once. "Think Big, Start Small and Learn Fast."***



Fully automated deployments to production (higher than 46%)



Multiple handoffs between teams are required for deployment of products and services.



* White paper: A holistic approach to DevOps
** Accelerate State of DevOps
*** BVSSH Principles - Jonathan Smart

Your Starter Kit for Enterprise DevOps

Key capabilities with the highest impact on performance for low-performing teams

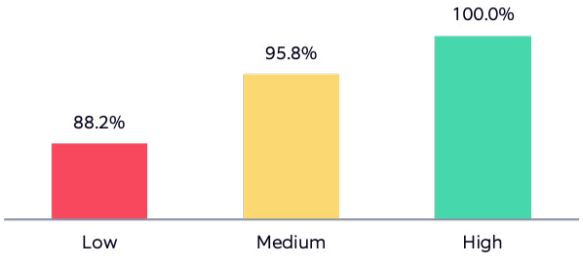
To establish the foundation, start with the basics:

- basic automation (e.g., version control and automated testing), monitoring
- clear approval procedures for changes
- a healthy corporate culture

That said, three capabilities have the highest impact on improving your software delivery performance and productivity when you're just about to start your DevOps journey.

Code maintainability

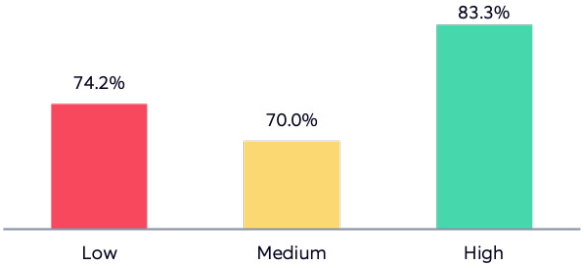
Systems and tools make it easy for developers to change code maintained by others, find examples in the codebase, reuse other people's code, and add, upgrade, and migrate to new versions of dependencies without breaking their code.



Changes in the code base can easily be tracked and audited

Continuous testing

Testing throughout the software delivery lifecycle rather than as a separate phase after "dev complete." With continuous testing, developers and testers work side by side. High performers practice test-driven development, get feedback from tests in less than ten minutes, and continuously review and improve their test suites (for example, to better find defects and keep complexity under control).

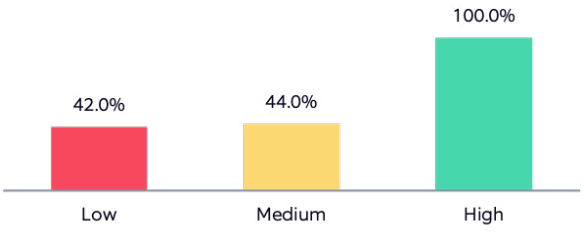


Automated test suites are executed at least once every day.

Cloud infrastructure

Many organizations are adopting cloud computing. But there's more to cloud than buying your infrastructure from a cloud provider. The US National Institute of Standards and Technologies (NIST) defines five essential characteristics of cloud computing.

You will also find additional information on this topic in our report.



Cloud-only usage

Scaling Enterprise DevOps

nagarro

Transform & scale

Improve your performances

When it comes to scaling to Enterprise DevOps, it's important to realize that it's not a journey one can take overnight. Specific practices emerge by applying principles to a specific context, through coaching and experimentation, leveraging decentralized knowledge.

The successful pattern is to identify the most important principles, that you wish to promote most in your organization, communicate them relentlessly, and recognize behaviors that align with them. They are positive behavioral guardrails.

The following approaches can have a positive impact on spreading agile and DevOps methods*:

- Ways of Working Center of Enablement
- Advocacy or Enablement teams
- Community of Practice
- Bottom-up or grassroots

What approaches can have a negative impact on your success:

- Training centers or dojos
- Center of Excellence
- A big-bang approach

The best strategies for scaling DevOps in organizations focus on structural solution that build community.

* State of DevOps Report 2019

Influence your culture to be successful.

2

Culture is one of the foundations. It is a key factor in successful and scalable technology projects. Successful teams need a culture of trust, autonomy, psychological safety, meaningful work, and clarity.

This kind of team environment allows members to take calculated and moderate risks, speak up, and be more creative.

Tools & culture: A lovestory

Improve your performances

DevOps is often referred to as a tool centric change in the way of working of teams, but it is also becoming public knowledge that a DevOps evolution influences culture.

An aspect which we experienced is easily overlooked, is that there are technical practices from the continuous delivery capability which have a big impact on cultural improvement. Some of those practices can be seen as a steppingstone for your DevOps evolution, leading to quick wins.

Easily adoptable practices for beginners to get going are for example:

- Version control
- Continuous integration
- Test automation
- Continuous testing
- Trunk-based development
- Empowering teams to choose tools

Generative culture

A quick recap

According to Westrum, a generative culture is essential for establishing a high-performing team. This finding as widely agreed upon in today's software development world. Creating a culture of trust, psychological safety, meaningfulness and clarity is imperative for creating a highly functional and highly performing team.

The six key aspects of a team culture according to Westrum are:

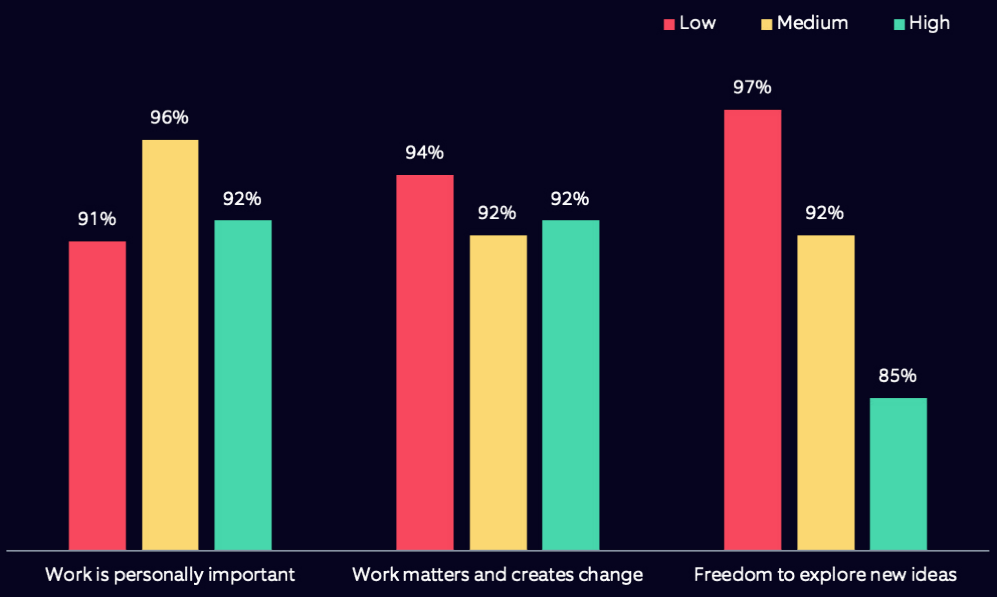
Pathological	Bureaucratic	Generative
Power oriented	Rule oriented	Performance oriented
Low cooperation	Modest cooperation	High cooperation
Messengers "shot"	Messengers neglected	Messengers trained
Responsibilities shirked	Narrow responsibilities	Risks are shared
Bridging discouraged	Bridging tolerated	Bridging encouraged
Failure leads to scapegoating	Failure leads to justice	Failure leads to inquiry
Novelty crushed	Novelty leads to problems	Novelty implemented

Westrums' principles for a performance-oriented culture in today's software development projects are widely agreed upon and are adopted by about 77% of participants.

Experience vs data

According to our results more than two thirds of participants classify their culture as being very close to a generative culture reflective of their key values. This was a surprising finding for us since our experience shows that this rating should be much lower. A possible explanation could be that only one data point per team was allowed and the survey was often filled out by a lead or senior team member.

Interestingly, the data shows that the freedom for exploration, taking pride in the work and creating change were hugely important across all performance levels.



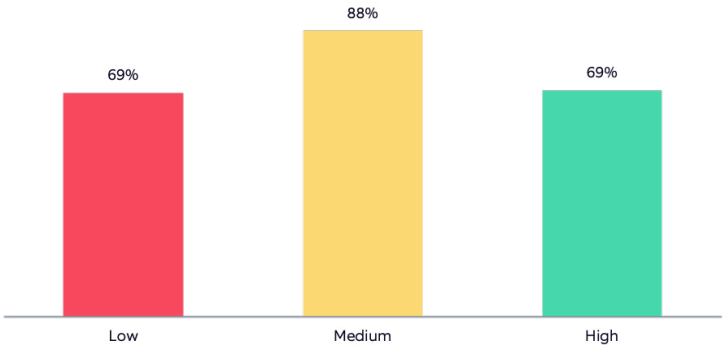
Team setup

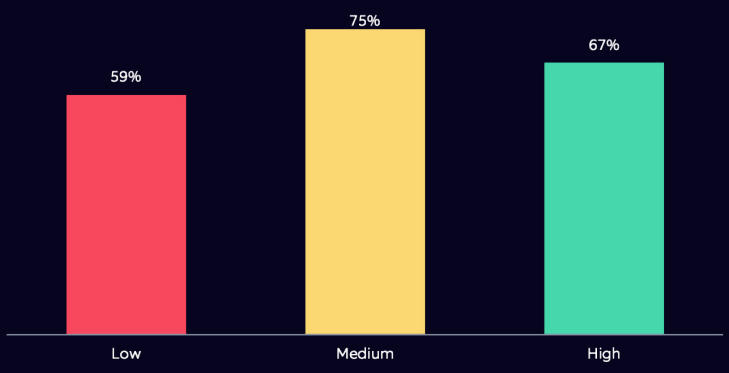
We are fully cross functional

The team is consisting of e.g., business analysts, developers, quality engineers,...

An interesting finding in our results is, that cross functionality is distributed in a bell shape across performers. 69% cross functionality is a good result for low-performing teams but is significantly out done by the mid-performing ones at 88%.

High-performing teams are only 69% cross functional. This drop could be caused by a broader versatility of high performers team members and thus reduced their need of highly specialized ones.





We are fully co-located

The team is working in the same office, ideally in the same room

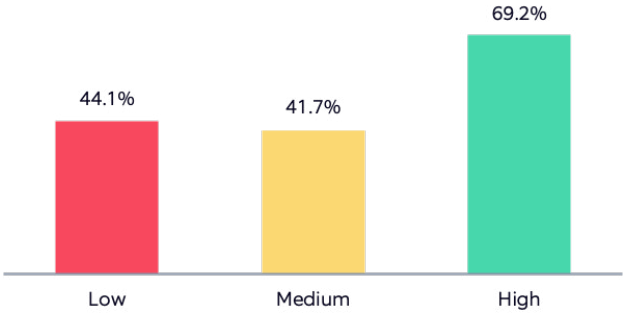
The results show that the percentage of teams working fully co-located is rising from low-performing teams 59% to mid-performing teams 75%, followed by a drop down to 67% for high-performing teams.

This drop from mid to high profiles could be caused by the experience in the high-performing teams and that in our experience, although being fully co-located is the optimal way of working, the more expertise and seniority a team has, distance is becoming increasingly more irrelevant.

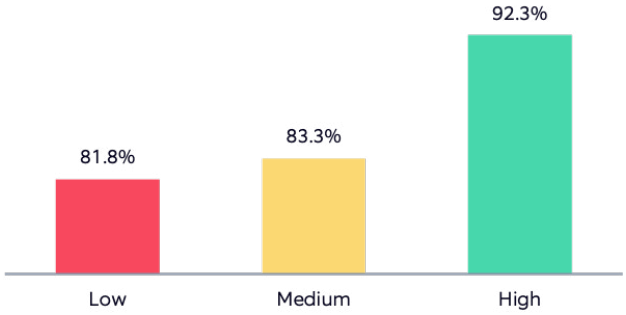
Collective ownership

Our data shows that the principle of collective code ownership is well adopted throughout the performance profiles.

With an agreement rate of above 80% across the board, there is also a distinct separator between the low- and mid-performing teams in comparison to the high-performing ones in the team members freedom to add changes to the code base.



Anyone can change our source code

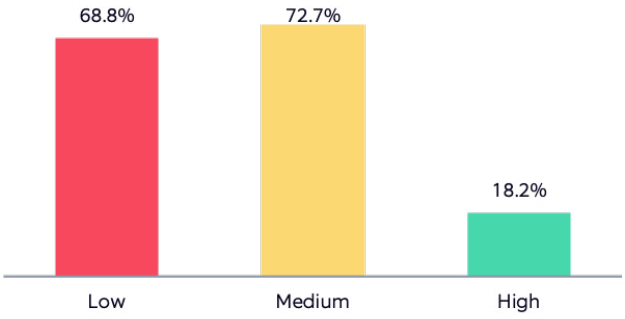


The source code is owned by the team

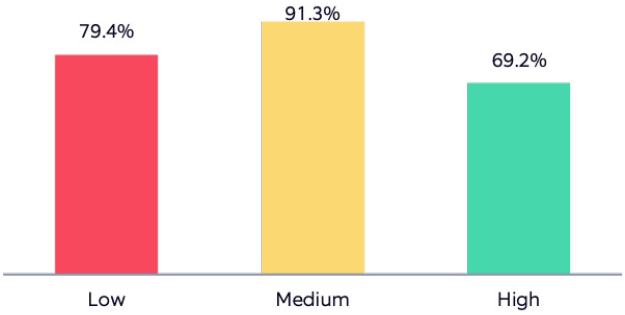
Collaboration among teams

Collaboration among teams is an important and interesting aspect in a DevOps evolution. It is imperative for a well functioning DevOps team to break up silos and work a lot with other teams to increase transparency and communication.

Going from very explicit efforts to accomplish things to being more implicit in their ability to collaborate while still being able to solve their blockers is very well represented in our data.



We regularly need to be unblocked by other teams in real time.

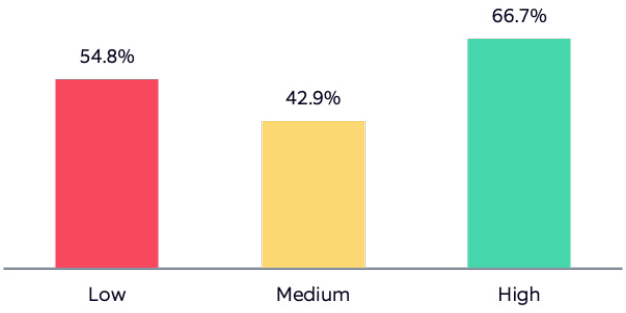


We collaborate with other teams during the design phase of features and solutions.

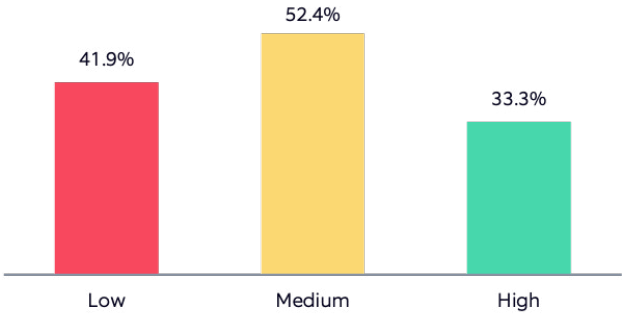
Team experimentation

Experimentation vs generative culture

As stated before, Westrums' principles for a generative culture are widely agreed upon and are adopted by about 77% of participants. With that in mind it is very interesting, that the team autonomy for experimentation and working without explicit permission is far behind our expectation revealing a possible bias about the teams' design scope and freedom.



We are able to do work without having to ask for permission.



We are able to make changes without having to ask for permission.

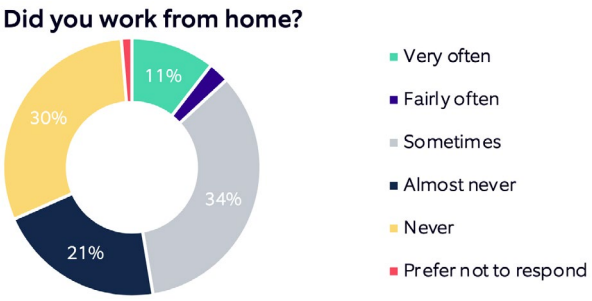

COVID-19

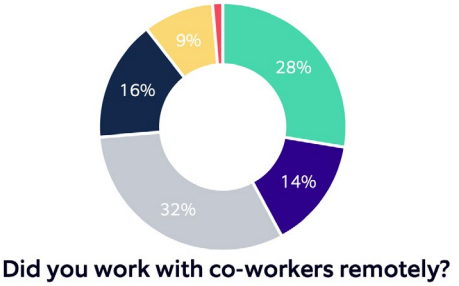
Remote work

Shaped by the pandemic

COVID-19, has had a defining impact over the last 2 years. Before the pandemic, only 10% said they often worked from home, and 37% said they sometimes worked from home. Collaboration has also changed noticeably, before the pandemic, 57% of respondents said they rarely or never collaborated with others remotely.

These values changed massively during the pandemic. We are curious to see how these figures will change in the coming years.





Did you work with co-workers remotely?

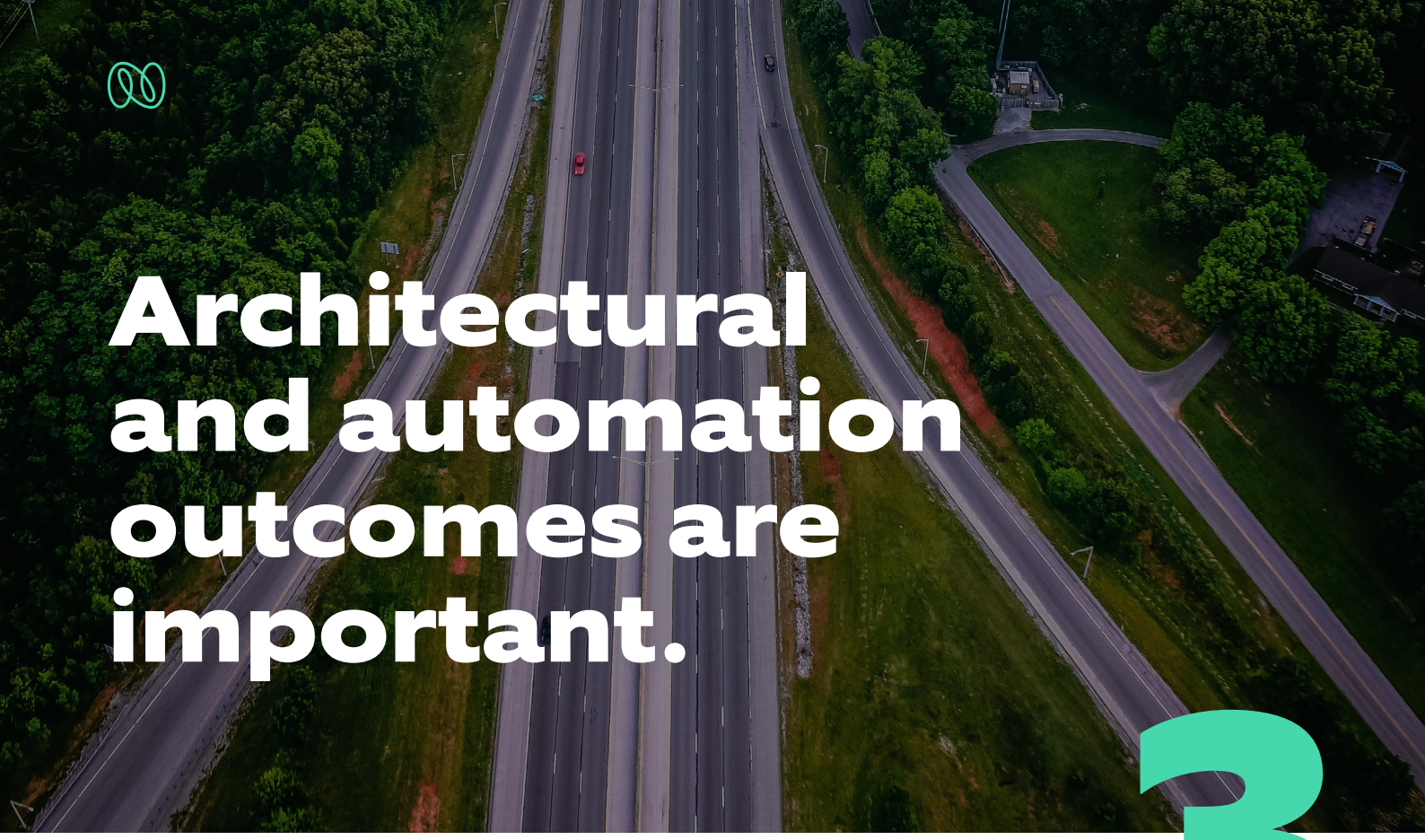
A change in the way of working

The pandemic has hit us all very hard. It has turned our lives upside down, but it has also possibly brought us closer to our loved ones again.

When evaluating the questions on the status before, during (and after) the pandemic, trends can be seen.

99% of respondents answered YES to the question “Have you been working from home during the COVID-19 pandemic?”.

Please keep taking care of yourself and stay healthy.



Architectural and automation outcomes are important.

3

The organization is compelled to produce designs that reflect or imitate the organization's real, onsite communication structure. This has significant strategic implications for any organization that designs and builds software systems, whether internally or through third parties.

The team structures and the communication between teams provide the foundation for successful product development.

Teams must achieve architectural outcomes to ensure that the quality required for complete and reliable automation is met. Establishing a high level of automation provides measurable benefit for project when it comes to DevOps performance.

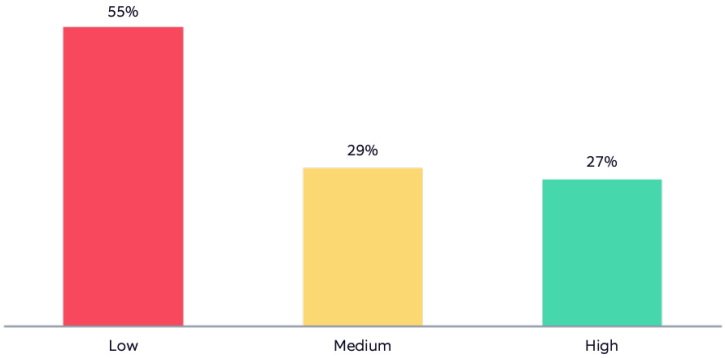
Hand(s) off

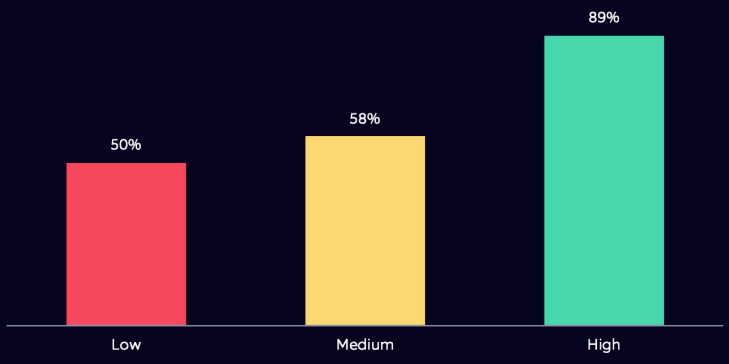
Multiple hand-offs

The required, manual activities to handover on batch of work to another team

Low-performing teams are more than 2x as likely to have to hand off their work multiple times to other teams to get it into production, compared to their medium- or high-performing peers.

Based on past project experience and supported by results presented later (team structure), this is most likely not caused by technical limitations, but rather the non-optimal structure and unnecessary dependencies between teams and products.





Fully automated software deployment. No manual hand-offs necessary

Such handovers take time and prevent product deployment from being fully automated, reliable, repeatable, and sustainable.

When we look at how many of our participants' deployment pipelines are automated, there is sharp increase in fully automated deployments for high-performing teams.

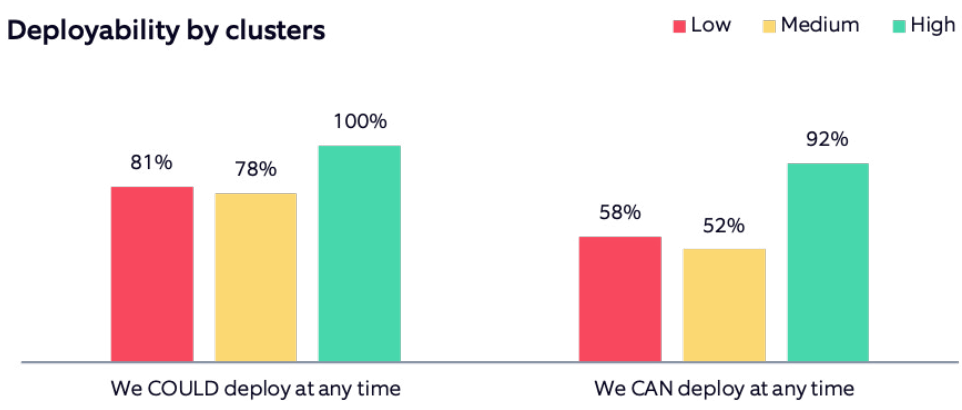
If you can deploy, why don't you?

Deploying frequently is a staple in agile work environments. You want your customers to have quick access to your new features – not only to profit from a fast feedback loop.

While a very high percentage of respondents across all performance profiles have a potentially deployable solution throughout its lifecycle, just over half of low- and medium-performing teams can deploy to production at any given time. High-performing teams, in contrast, are almost 2x as likely to be able to deploy on-demand.

We see two potential scenarios:

- External factors could inhibit continuous delivery (e.g., looking at restrictions in different app stores), which you can't influence directly.
- There might be internal factors, like missing automation pipelines and long-winded approval processes, which can be tackled by improving your tooling and culture.



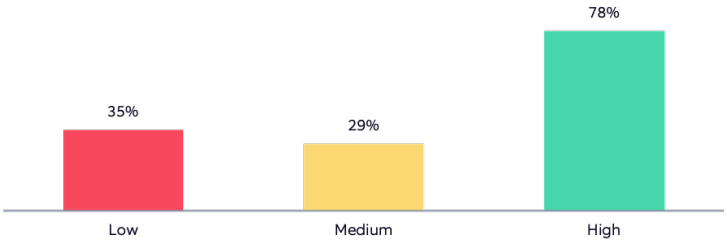
Liberate teams

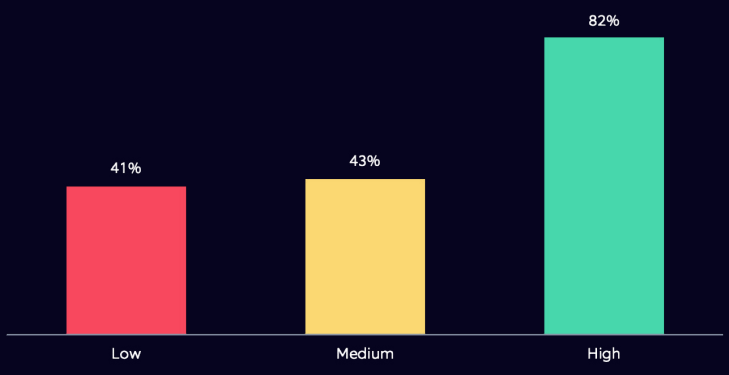
Independent Changes

Implementing large-scale changes without depending on other teams

Supporting the previously mentioned findings that only half of the low- and medium-performing teams can actually deploy at any point in time, is the fact that there is a high inter-team dependency for large-scale changes. This is especially noticeable in low- and medium-performing teams, where only 35% and 29% can make implement large-scale changes to their system without depending on other teams to make changes in their systems.

In general, it is advisable to liberate teams as much as possible and design team- and system boundaries to support independence and isolation as much as possible. Improper team structures that require alignment between teams prevents this and can lead to inefficiencies that are difficult or even impossible to remedy by tools or technical practices.





Independent Deployments

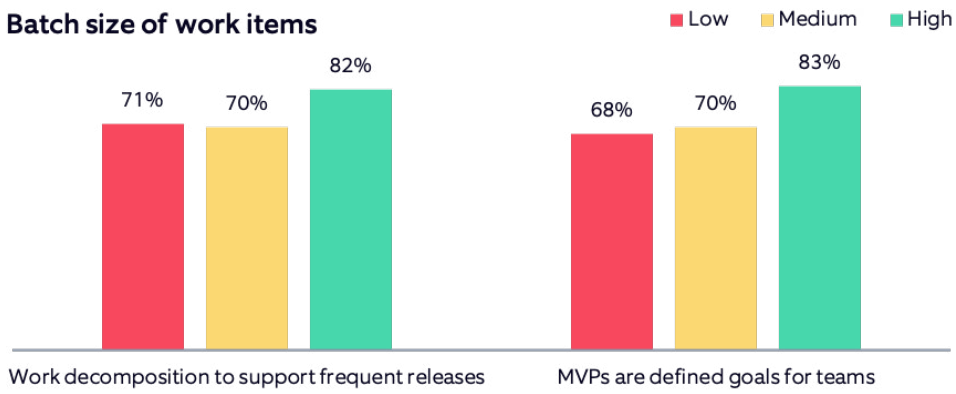
Deployment and release of changes independently of other services

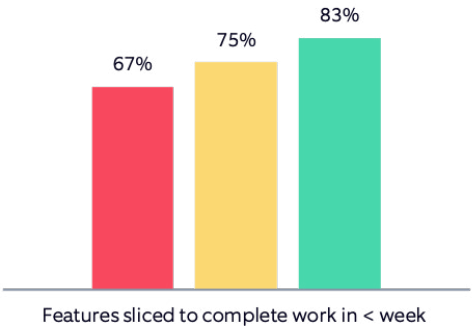
Similarly, to the prevalent dependency between teams for making changes, the results also shows that only less than 50% of low- and medium performing teams can independently deploy and release their product or service. Modern styles of architecture (Microservice oriented architecture) and clear interfaces between teams and products should not require those dependencies which often lead to complex deployment pipelines, time consuming release alignment procedures and overall, a reduction in fast flow of value.

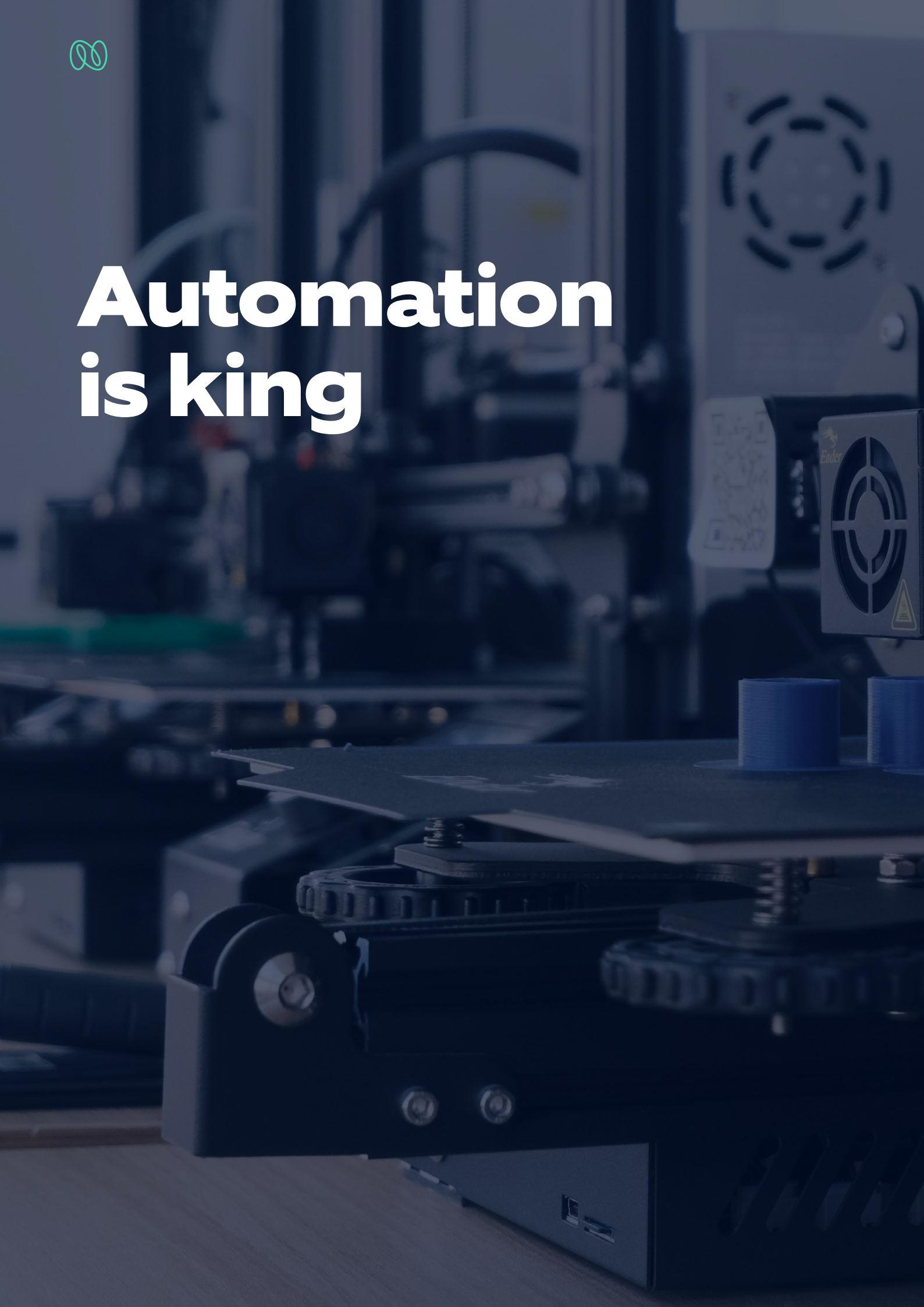
Batch size

Our data show that there is a slight trend supporting the claim that focusing on small batch sizes leads to higher delivery performance.

Slicing of features, which might be hard considering Conway's Law, is more demanding for low-performing teams, because this requires good system and team boundaries and high level of automation throughout the whole software delivery lifecycle.





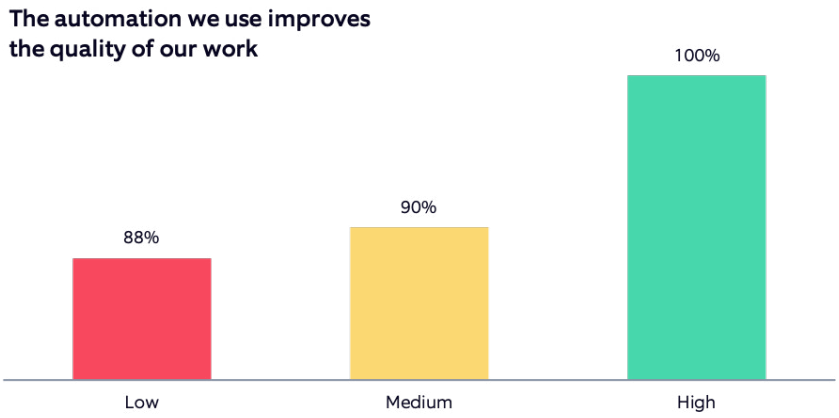


Automation is king

Automation improves quality

Almost all respondents are sure: Automation improves quality. Not everyone has this automation in place, but high performing teams have apparently mastered it.

But simply acknowledging the usefulness of automation in improving the quality of work does not necessarily lead to universal adoption in every area. For example, automated regular performance tests are only established by around 55% of respondents , with no noteworthy variation between performance clusters. Security tests or common vulnerabilities and exposures (CVE) checks are even less commonly automated and thus lead to potentially insecure software being released to the public.



Automation by performance profile

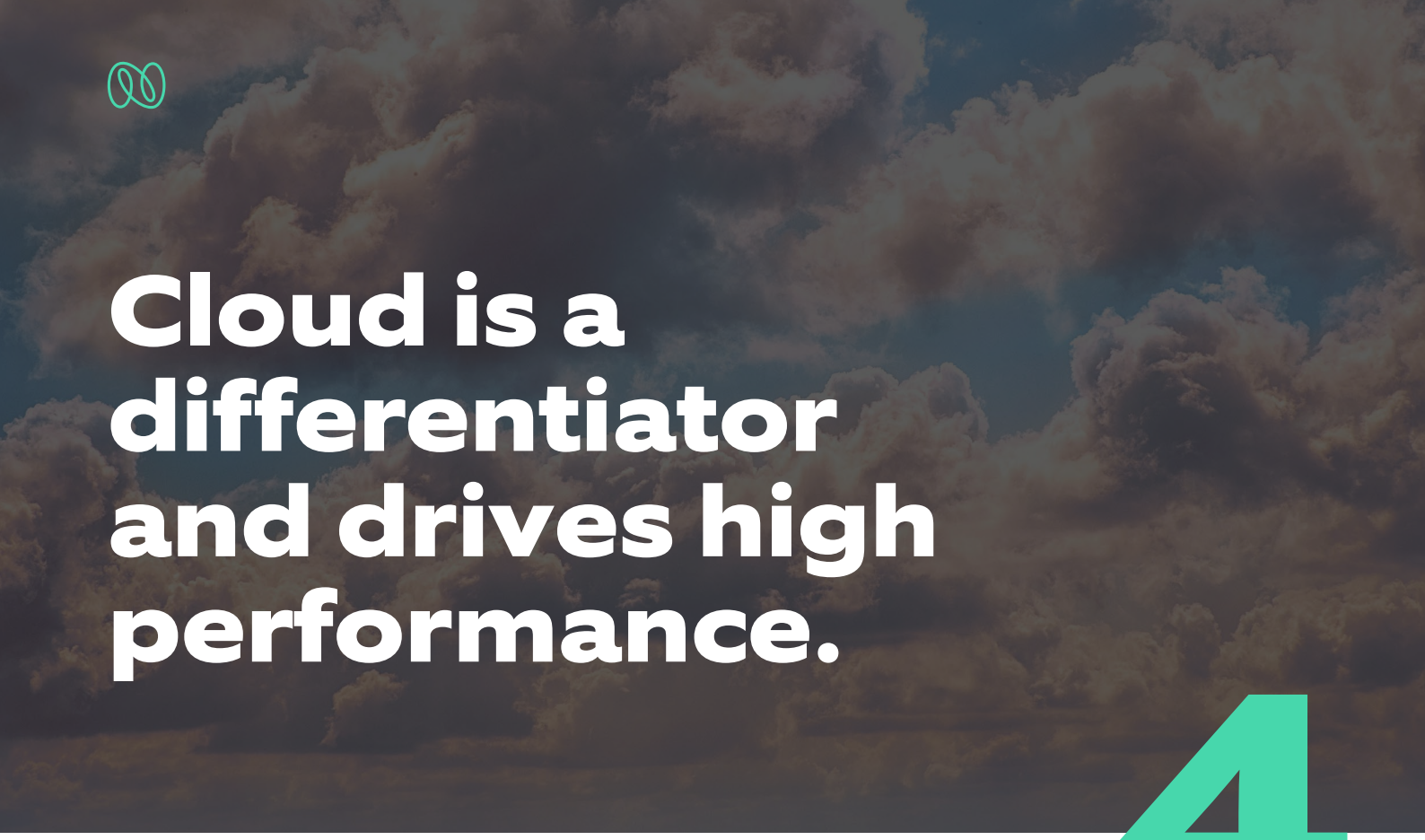
The unanimous agreement with automation, in general, is only partly reflected when analyzing concrete practices in detail.

As mentioned before, there seems to be a general lack of automated security and performance testing across performance clusters. In a world where serious security issues arise more frequently than ever and your company might just be one more data leak away from losing your customers' trust, putting security measures early on should be more a point of focus (currently overall, just above 50%).

Even relatively simple to introduce and commoditize (e.g., via a common platform) practices like automated common vulnerabilities and exposures (CVE) checks are only used by 55% of the respondents.

The decrease in the adoption of test result notification is likely related to a more tightly integrated usage of test automation by high-performing teams. This is where automated test results are primarily used as an automated quality gate for various stages of continuous integration or continuous delivery pipeline. In that case, the necessary information is if a quality gate passed or failed. In combination with in general smaller batch sizes, this limits the importance of active, push-based test result notifications.

	Low	Medium	High
Automated builds	94%	70%	82%
Automated tests	81%	57%	82%
Automated test suites	74%	70%	83%
Automated performance tests	55%	55%	55%
Automated security tests	70%	33%	50%
Automated provisioning and deployment to test environments	50%	58%	89%
Automated evaluation of test results	88%	70%	67%
Automated test result notifications	71%	63%	55%



Cloud is a differentiator and drives high performance.

4

Looking at Amazon, Microsoft and Google, some of the biggest tech giants on the planet have not only pioneered cloud technologies but established their own cloud products for others to consume. It should come as no surprise, that lots of companies take them up on these offerings and move to the cloud.

Our data show, that mainly teams from the higher-performing clusters are making significant use of cloud functionalities.

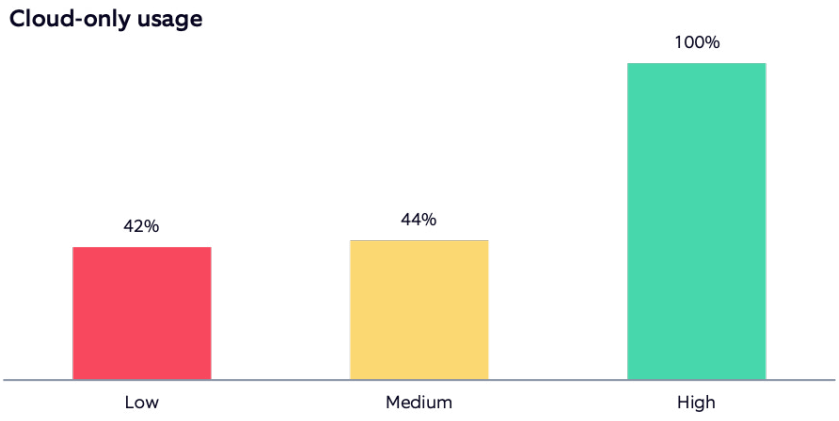
Yet, there is still room to improve.

Moving into the cloud

Through the last few years, cloud adoption was already on the rise. This already impressive growth was exponentially accelerated by the global pandemic. As Microsoft CEO Satya Nadella puts it: “We’ve seen two years’ worth of digital transformation in two months.”*

Our data shows: You are following this trend as well.

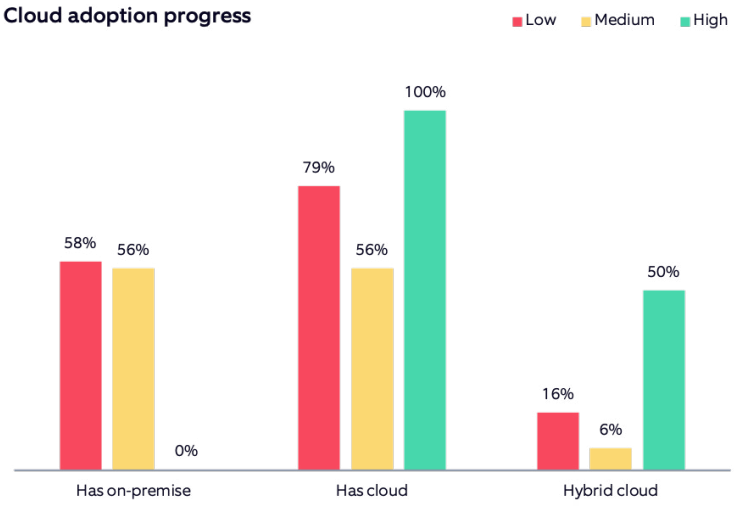
Especially the category of high-performing teams have made the cloud their home, with an adoption rate of more than 2x the rate of their peers.

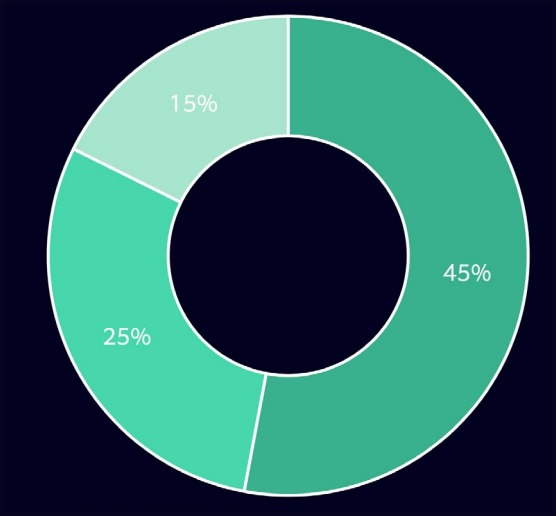


* Article: 2 years of digital transformation in 2 months

The hybrid/ multi-cloud approach

While more than half of all low and medium-performing teams have on-premise hardware that is not used in a private cloud environment, half of the high-performing teams embrace hybrid cloud solutions. They use a combination of public and private cloud infrastructure to further their needs.





Not only that, but about half of all respondents are utilizing multiple clouds to accomplish their goals. Their reasons for adopting several cloud providers are:

- Leveraging unique benefits of each provider (45%)
- Availability (25%)
- Disaster Recovery (15%)

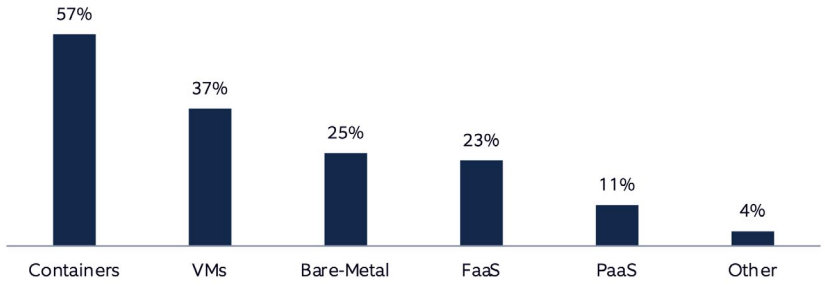
Containers: In the spotlight

When we asked where you deploy your services , one thing was clear: Virtualization is everywhere. And within that sector, containers and their need for orchestration are most widespread nowadays, as clearly more than half of teams are using it.

While there is still a considerable portion of bare-metal and pure VM deployments, a growing share already uses other cloud-native operation models like Platform-as-a-Service (PaaS) and Functions-as-a-Service (FaaS).

Standing out: FaaS usage sees almost 3x the rate of adoption (54%) in high-performing teams, compared to their peers in low-performing teams (20% respectively).

Where are you deploying your apps or services?



“Cattle, not pets.”

A mantra well served by containerization and cloud technologies.

Characteristics of cloud adoption

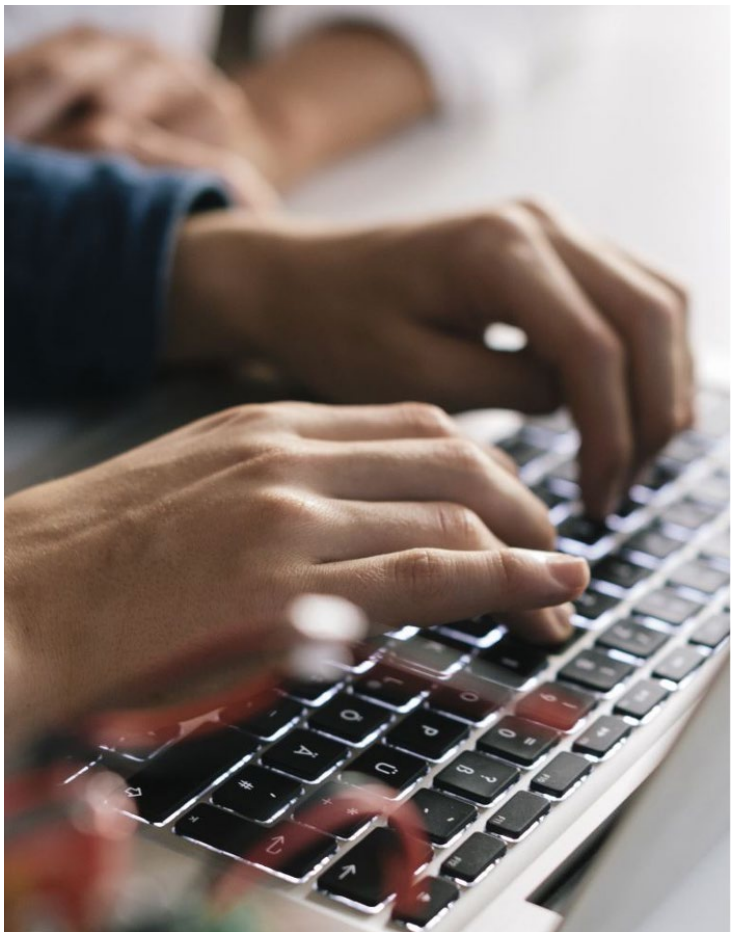
How cloud is helping DevOps

National Institute of Standards and Technology (NIST)

It's been ten years already since the U.S. National Institute of Standards and Technology (NIST) defined the five essential characteristics of cloud computing in 2011*, and those characteristics still stand today. They provide a good understanding of what benefits the cloud can offer, and we can infer, why those benefits can help implement a proper DevOps strategy on a technical level:

- **On-demand self-service** – It's easy to provision your resources in the cloud, and more importantly, you're able to automate this process (maybe even as code?).
- **Broad network access** – Providers promote a standardized way of accessing cloud resources, most commonly as web services, so that they can be used by all kinds of client devices.
- **Resource pooling** – Instead of needing to care about where exactly your resources will live, you can take more general decisions (country, city, ...) and your provider takes care of the rest.
- **Rapid elasticity** – Cloud providers allow you to easily and sometimes automatically scale your resources, allowing you to be flexible and provide as demand dictates.
- **Measured service** – If you can't measure it, you can't improve it, as the saying goes. Cloud infrastructure often comes with built-in monitoring and reporting tools.

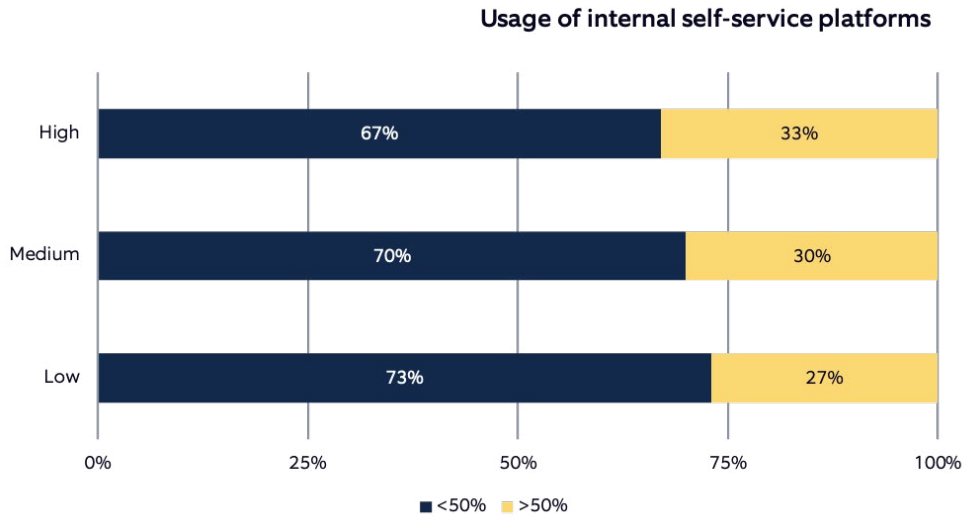
* The NIST Definition of Cloud Computing



On-demand self-service

Giving your developers an easy, on-demand, self-service way of resolving their needs enables them to work faster by automating away complex processes and reducing communication needs.

While 59% of teams say that they provide self-service APIs for other teams to consume, when it comes to the cloud and infrastructure self-service platforms, only an average of 30% of cases are used by more than half of the developers, with a slight uptick on higher-performing clusters.

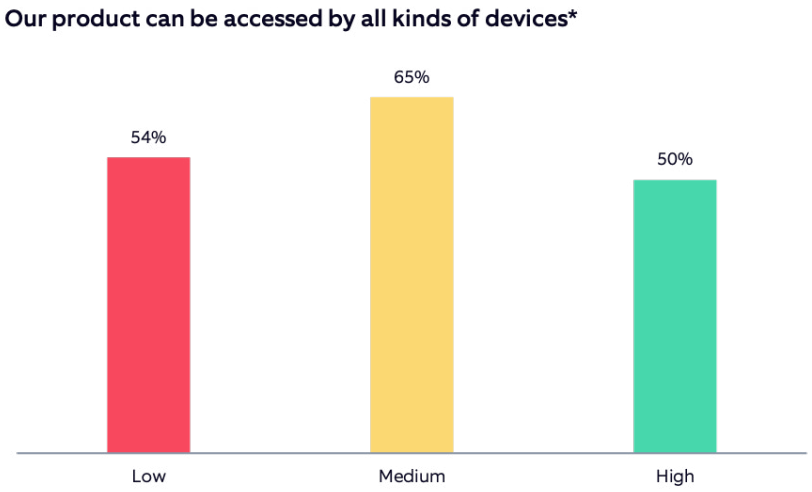


Broad network access

User device independence

The cloud offers broad network access, allowing all kinds of devices, from workstations to smartphones, to access products and services over standardized mechanisms, like your typical web service.

This can be especially valuable to reach your customers, now that almost everyone has a mobile device. Not all respondents make use of these possibilities thought, as just a bit more than half (54%) provide their services for a broad range of end-user devices.



* The full question text was: "The service or product that I primarily work on is designed to be accessed from a broad range of devices (e.g., smartphones, tablets, laptops) over the network without the need for proprietary plug-ins or protocols."

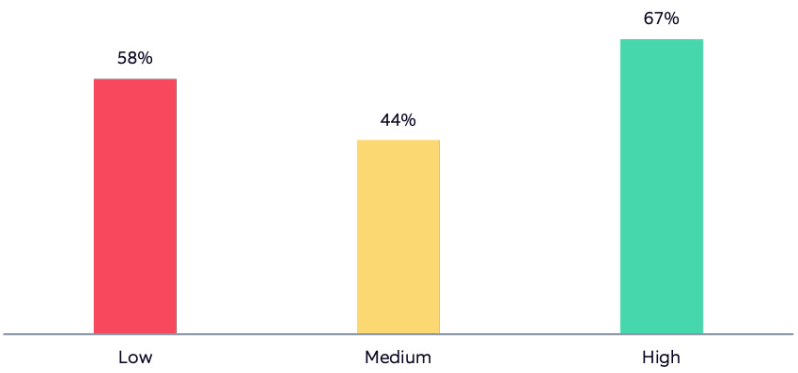
Resource pooling

When it comes to the cloud, collective ownership isn't as big yet, as it is in other parts of the development process. Just more than half of the teams (58%) share their cloud environments with other teams, instead of using them exclusively for their purposes.

Looking at the profiles, we can see that the high-performing teams are slightly more prone to sharing their environments, enabling them to use knowledge sharing and resource pooling.

Sharing is caring.

The cloud my product or service runs on serves multiple teams and applications (%)



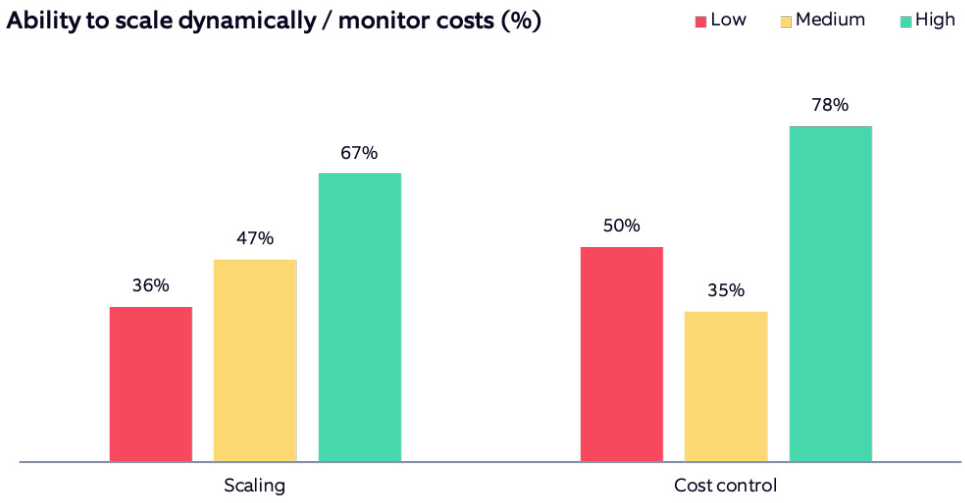
Rapid elasticity and measure services

Cloud adoption comes with many benefits, but also new challenges.

While high-performing teams do a better job at utilizing cloud scaling capabilities, a third of them still have some homework to do. Two thirds of the low and more than half of the medium-performing teams must catch up even more.

Inventory management and cost control seem to be a similarly issue. Currently just half of the teams in the low-performing cluster know what resources they have and what they have to pay for it, while in the high-performing cluster almost 80% have a grip on these numbers.

Untapped potential.



Who took the survey?

Nagarro's research provides insight into software development and DevOps practices from working professionals in the Nagarro environment. The current global population of active professional software developers will increase from 23.9 million in 2019 to 28.7 million in 2024.*

This year, almost 180 people from various industries across the Nagarro family and friends added their voices to the 2021 report. We want to take this opportunity to thank all participants of this year's survey.

We hope to accompany you on your DevOps journey in the coming years.

* Evans Data Corporation - Worldwide Professional Developer Population

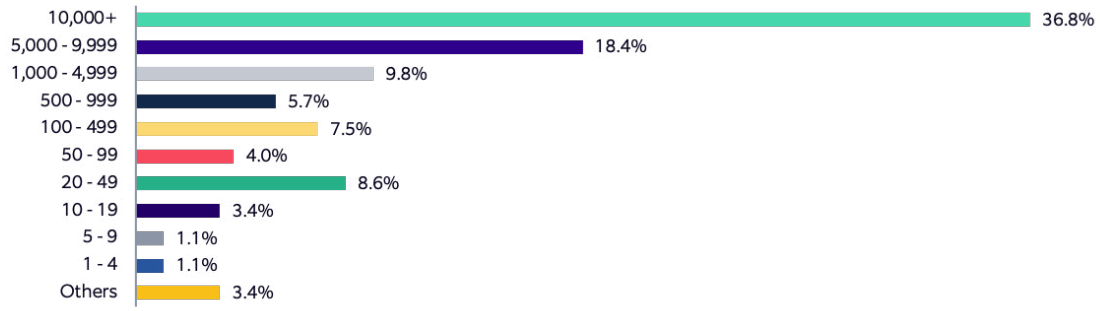
Organization

Organizations from – to

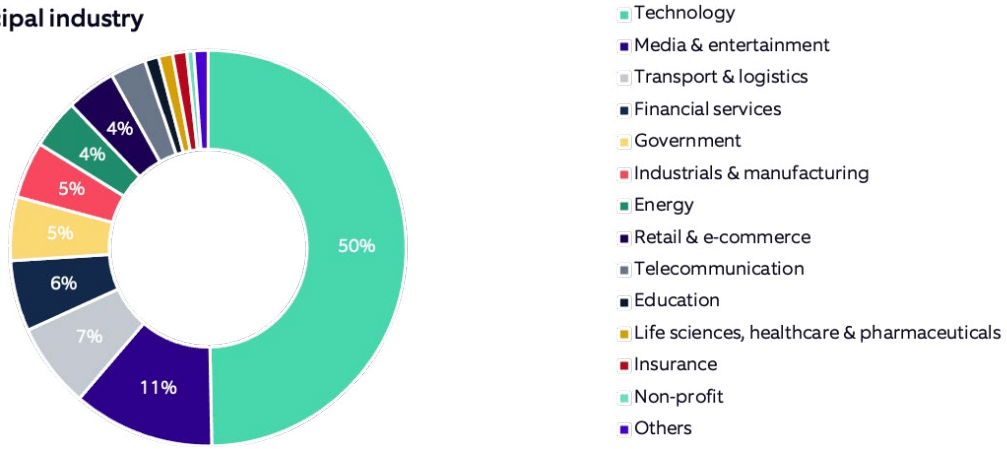
As we were able to convey, almost 70% of the respondents answered that they work in a large enterprise.

Around 26% see themselves in a small or medium-sized company (SME).

Organization size by employee headcount



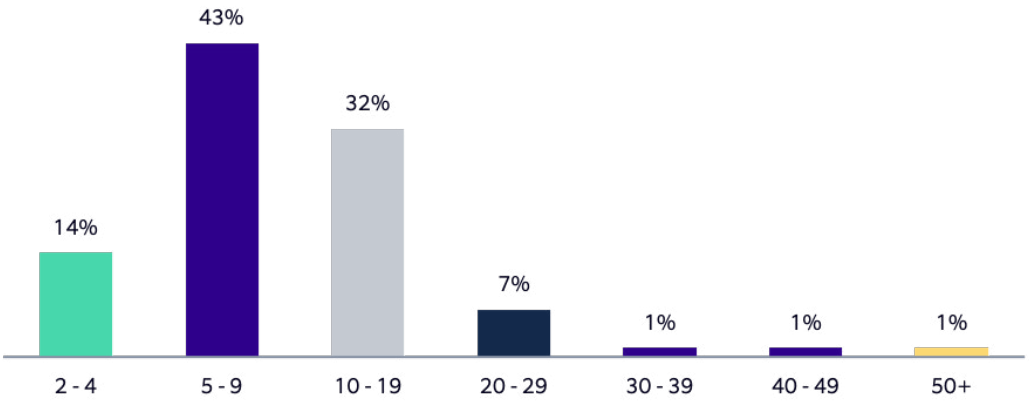
Principal industry



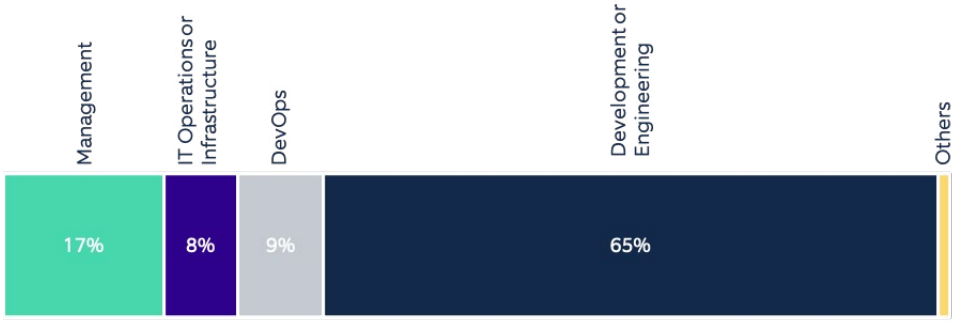
Teams and roles

Team size

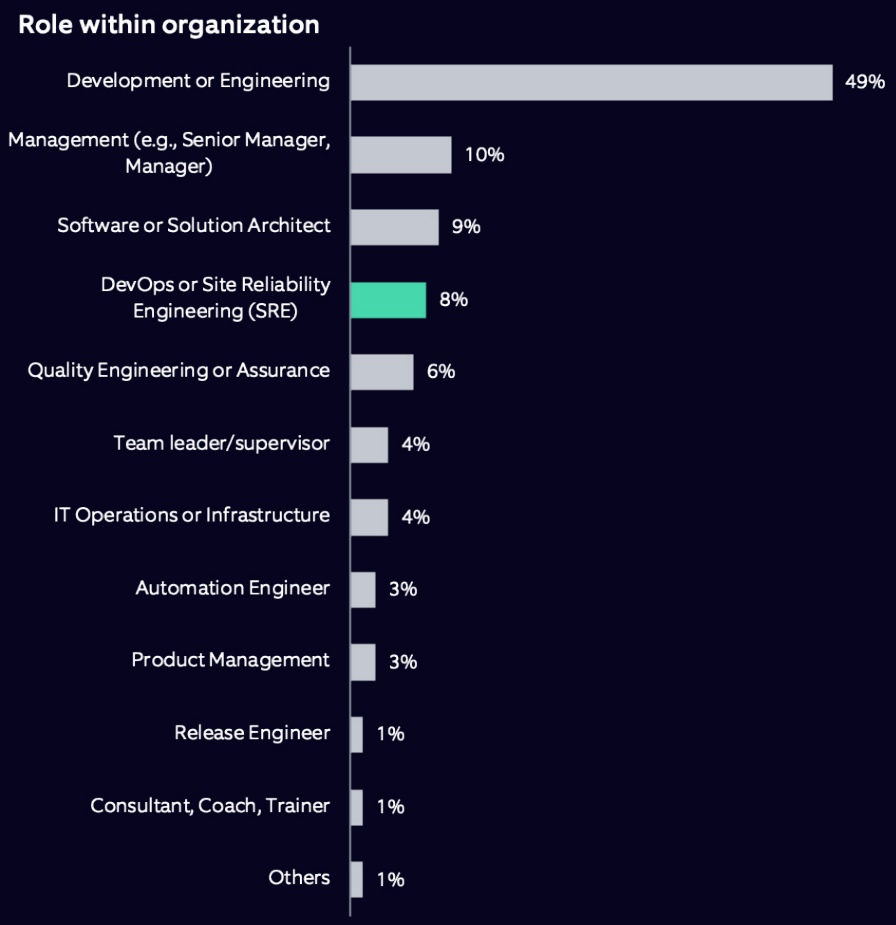
As is usual in agile teams, more than 40% of respondents assign themselves to such a team. Larger teams, in which communication becomes more difficult and possibly overhead, decreases.



Departments



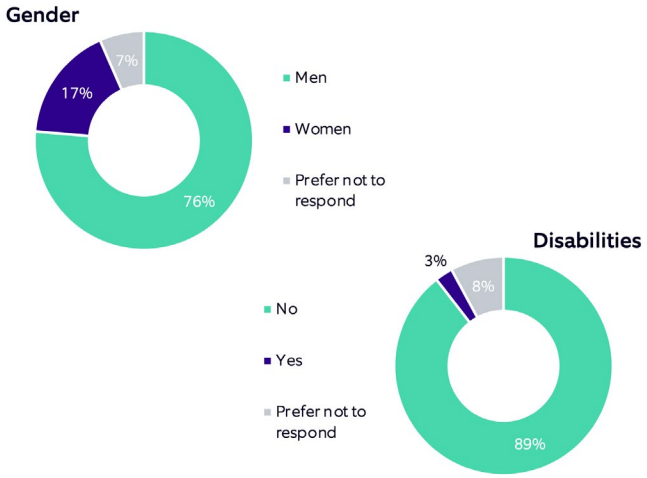
Teams and roles



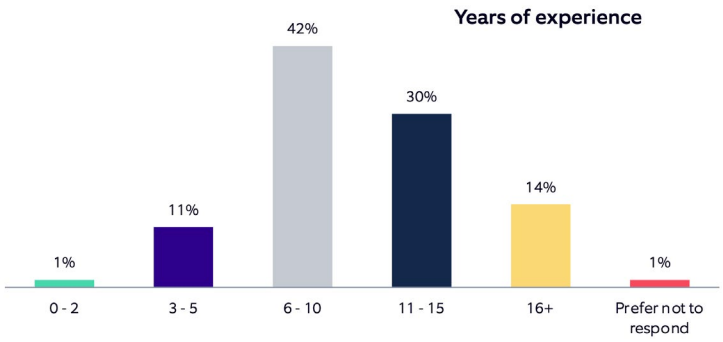
Gender, disabilities & years of experience

Underrepresented groups

The proportion of women, of those who participated in the first year of our survey, is just under 25% of the proportion of men, although women make up almost half of the total workforce.* Non-binary was included in our question, but none of the participants responded to the query with it.



The proportions seen when it comes to total respondents identify as having a disability is like elsewhere in industry.**



* This is mostly like proportions reported by the Stack Overflow Developer Survey 2019, which includes 90% men and 10% women. They do not include non-binary and "prefer not to respond."

** Stack Overflow Developer Survey 2019, reports 3-8% of total respondents identify as having a disability

Final thoughts

In our first ever Nagarro State of DevOps Report, we tried to gain new insights into our client landscape to help organizations like yours achieve their goals faster and with less pain. In doing so, we also looked to empirical studies, such as DORA and Puppet. I would like to take this opportunity to thank them as well, because what they do helps our entire industry immensely.

We hope this year's insights will help you scale your DevOps practices and ways of working more broadly across your organization.

We'd love to hear about your experiences and your comments on the survey and the report itself. Please feel free to contact us! You can email us directly at:

qualitydevops@nagarro.com

It is incredibly exciting to see how you, as our clients, are moving in the DevOps environment and which ways of working you are already using. It's even more interesting for me to see where you are currently in your journey, and it helps us a lot to be able to support you even better on your DevOps path in the future.

I'm already looking forward to talking with you about the exciting topics for you in the coming years.

Jürgen Pointinger,
Quality DevOps Practice Lead at Nagarro

Authors

Jürgen Pointinger

Jürgen Pointinger is a DevOps enthusiast and Practice Lead at Nagarro. He supports clients in reaching the greatest sustainable, fast, and secure benefits. Thereby he emphasizes an open, value-adding culture, trust, and continuous learning. He started his career as a Developer before working as a Software/Solution Architect, Team Lead, CTO, IT Consultant and DevOps Coach. He spends his free time with his family and learns from his children to look at the world with different eyes.

Stefan Gwihs

Stefan Gwihs is a Test Automation Architect and DevOps Coach at Nagarro. After completing his bachelor's degree in computer science and subsequently a master's degree in multimedia and software development, Stefan focused primarily on agile software development and test automation. He is certified by ISTQB® and IREB and has a very rich experience in the optimal handling of IT projects thanks to his diverse project activities.

Raffael Šala

Raffael Šala is a Test Automation Architect, DevOps Expert and DevOps Community Lead at Nagarro. During his latest projects, his focus shifted more and more from a pure Quality Assurance view to a wider DevOps perspective and how to incorporate QA into DevOps. He is working with a very diverse range of clients with a wide range of DevOps seniority, from clients starting their DevOps evolution, up to clients struggling advancing in DevOps performance levels.

Franz Buchsbaum

Franz Buchsbaum is a DevOps Expert at Nagarro, with a master's degree in data management and computer security, and more than 10 years of professional IT experience. An early adopter of container technology, Franz spent the last years working in CI/CD automation and enabling development teams, before switching focus to his second love: Database design and optimization, and ETL processes. In 2021 he returned to dive back into the DevOps scene.

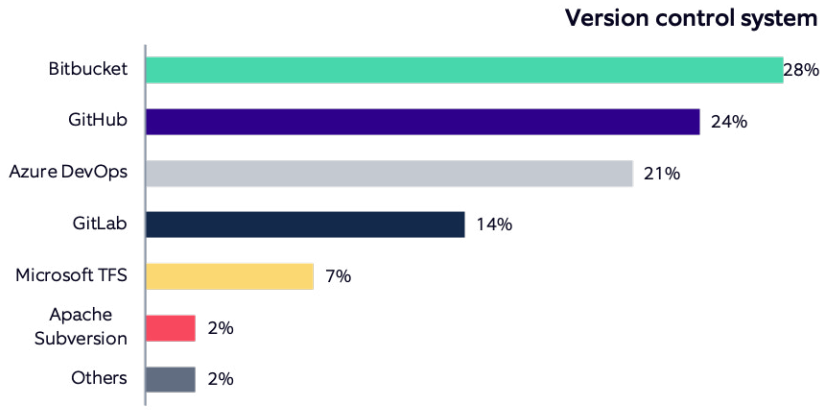
Appendix

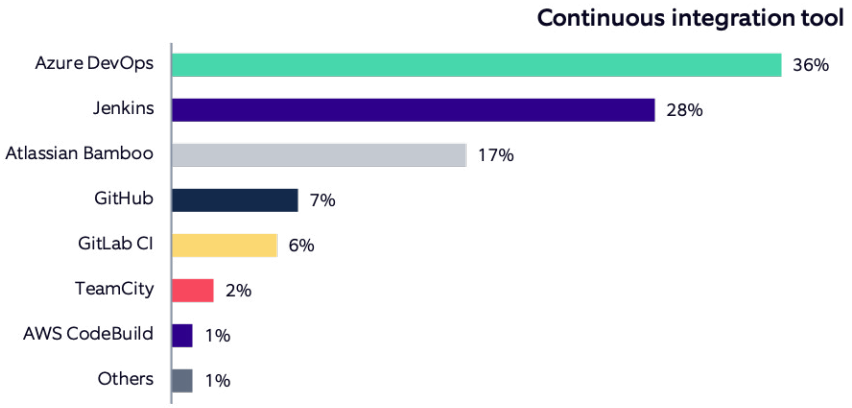
When we evaluated the survey responses, we also paid attention to the tools used in your work environment.

Even though a corresponding distribution is noticeable, we do not want to interpret the use of these, as they are needed for your specific use cases. Nevertheless, they provide us with a trend that will help us in our further orientation and thus also you.

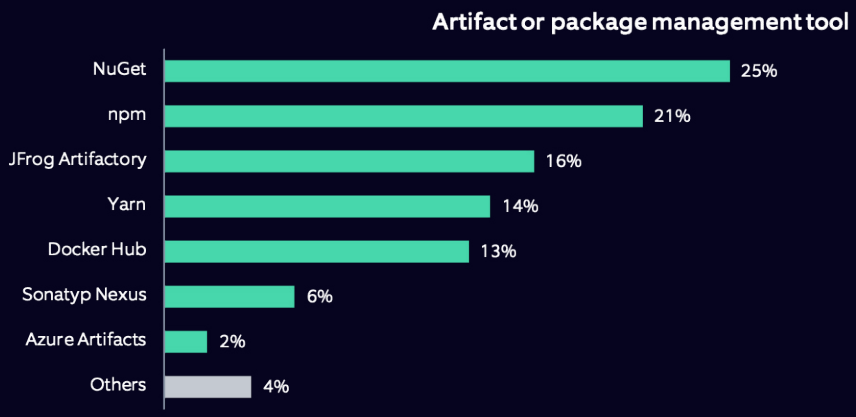
However, since we found them very interesting, you can consider them as an appendix.

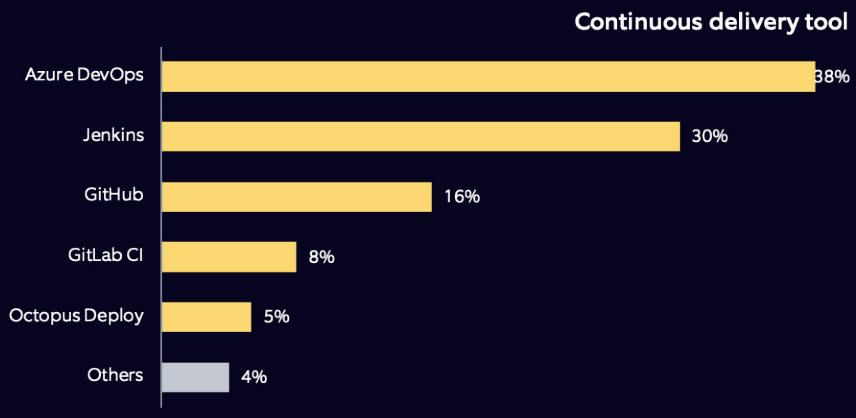
Appendix



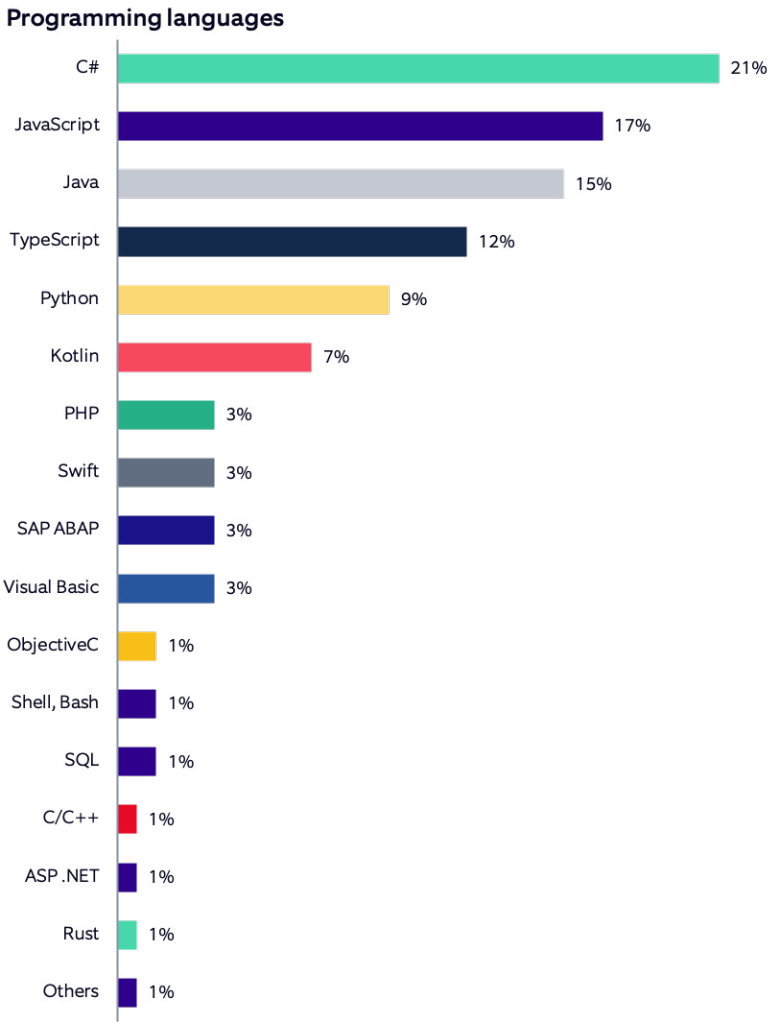


Appendix

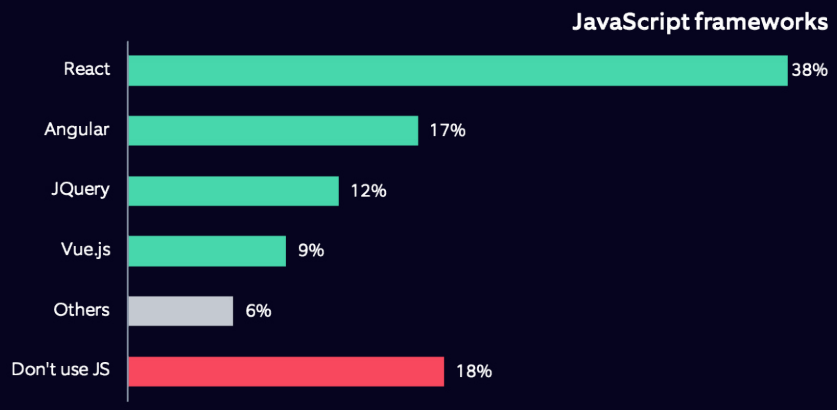


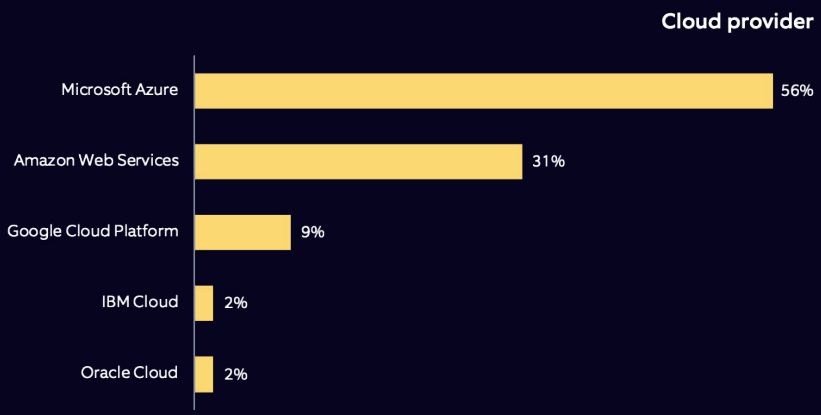


Appendix

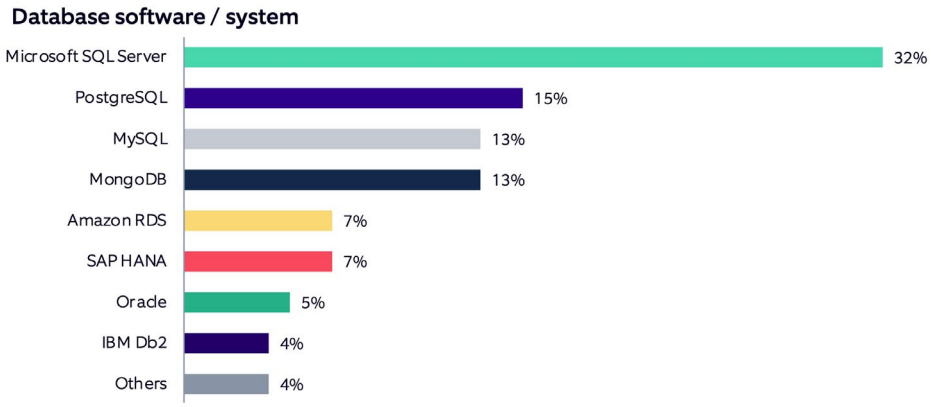


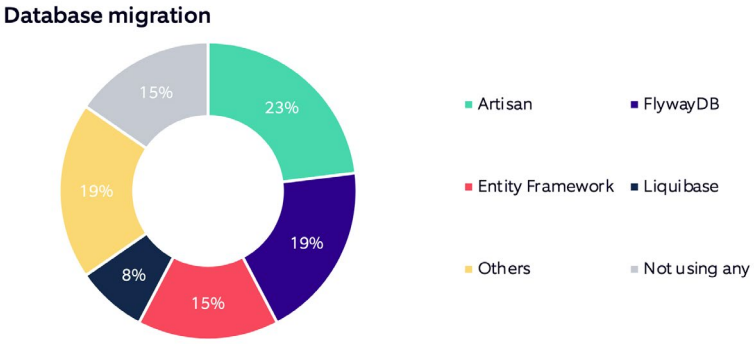
Appendix



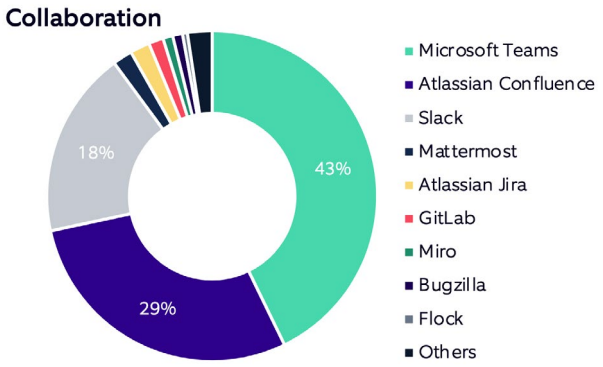


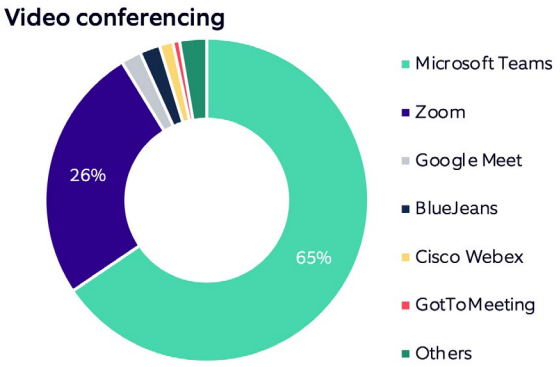
Database

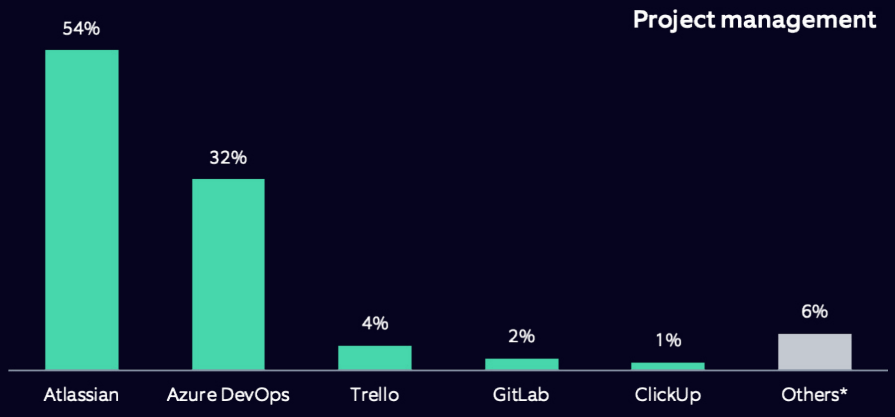




Collaboration, Video, Project management







* Asana, Clarizen, Forecast.app, MeisterTask, monday.com, Wrike, GitHub

Imagine what we can do together.

Let’s get in touch:
qualitydevops@nagarro.com

nagarro