

[DevOps](#) > DevOps metrics

DevOps metrics

TABLE OF CONTENTS ▾

If you adopted some of the technical, cultural, and process capabilities of DevOps, you may wonder whether your hard work is paying off. Adopting practices, learning tools, and building deployment pipelines takes time and effort. Now you need a way to see if this is helping you achieve your goals. You can do this with metrics.

To measure your work system, it needs to have the properties of observability. In [Observability Engineering](#) Charity Majors, Liz Fong-Jones, and George Miranda define observability as:

...a measure of how well internal states of a system can be inferred from knowledge of its external outputs.

If a system is observable, you can gauge its health from its outputs. You're already putting this into practice if you're monitoring your software. You need to do the same for your process so you can observe and generate ideas for improvements.

The most widely-recognized starting point for measuring DevOps are the DORA metrics, often called *the four keys*. Then, when you're ready to take your measurement journey to the next level, you can use the SPACE Framework to design your own measures.

Designing good metrics can be difficult. Your DevOps system consists of individuals in teams creating value, which Jim Benson calls [The Collaboration Equation](#). If you focus too much on a single metric, people might seek to achieve it to the detriment of other important measures. That's where DORA metrics and the SPACE framework use combination metrics to balance a system's response to measurement.

The goal of your DevOps metrics is to measure and improve the performance of the whole system. This is crucial. Bad things happen when you try to turn the measurements around

to assess and improve the performance of individuals. In DevOps, you measure success by the impact on organizational goals.

Let's look at the DORA metrics before expanding into the SPACE framework.

What are DORA metrics?

The [DevOps Research and Assessment](#) (DORA) metrics are performance metrics linked to DevOps success. The metrics result from extensive DevOps community surveys and identifying the factors that matter most. The DORA team conducts yearly investigations into the state of DevOps and shares their findings in a report. The DORA metrics are:

- Lead time for changes
- Mean time to recovery (MTTR)
- Deployment frequency
- Change failure rate
- Reliability

The DORA metrics should balance each other out. For example, lead time balances against change failure rate. Without change failure rate as a metric, you could achieve a very low lead time for change without accounting for quality. Research finds these metrics predict both software delivery performance and organizational success.

Lead time for changes

When delivering software, we store source code in a repository like Git. Developers make changes to the code by committing code to the repository. The commit doesn't immediately change the live product. Before reaching the production environment, changes must progress through a series of stages.

Lead time for changes is the amount of time it takes a commit to get into production.

To measure lead time for changes, you would track the time between commit and its deployment. By getting changes to production faster, you get valuable feedback sooner. DORA metrics give the standard for lead time for changes, where a shorter lead time is better for DevOps success.

For the primary application or service you work on, what is your lead time for changes (for example, how long does it take to go from code committed to code successfully running

in production)?

- **Low** - Between one month and six months
- **Medium** - Between one week and one month
- **High** - Between one day and one week

Mean time to recovery

All software is subject to failure, whether as a result of a change or due to external factors. Rather than attempting to reduce the number of failures, DORA recommends teams focus on the speed of recovery.

Mean time to recovery is how long it takes an organization to recover from a failure in production

The faster an organization can recover from failures in production, the better the experience for end users.

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 or service impairment)?

- **Low** - Between one week and one month
- **Medium** - Between one day and one week
- **High** - Less than one day

When measuring recovery times, you can reduce blind spots by looking at a distribution or scatter plot of resolution times. A rare incident with a long time to recover would be hard to spot against a mean or median average if there were many short incidents in the dataset.

Deployment frequency

A typical DevOps deployment pipeline splits code releases across different environments (Test, Development, and Production). Before progressing to Production, the Test and Development environments verify experiments and new versions. Once a release is ready to go live, it moves into the Production environment and the hands of end users.

Deployment frequency is how often an organization successfully releases to Production.

More frequent releases into Production can show:

- A higher number of updates for end-users
- More opportunities for feedback
- A streamlined deployment process

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

- **Low** - Between once per month and once every 6 months
- **Medium** - Between once per week and once per month
- **High** - On-demand (many deploys per day)

Change failure rate

When deployments make it to Production, they can sometimes cause errors. Minimizing the percentage of deployments that cause failures in Production can contribute to DevOps success. These failures often show that something is missing from the deployment pipeline. To detect these problems earlier, you can add automated testing to identify failures before they make it to Production.

Change failure rate is the percentage of deployments causing a failure in Production

A lower percentage indicates a higher score, and anything 15% or lower is a good result.

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

- **Low** - 46%-60%
- **Medium** - 16%-30%
- **High** - 0%-15%

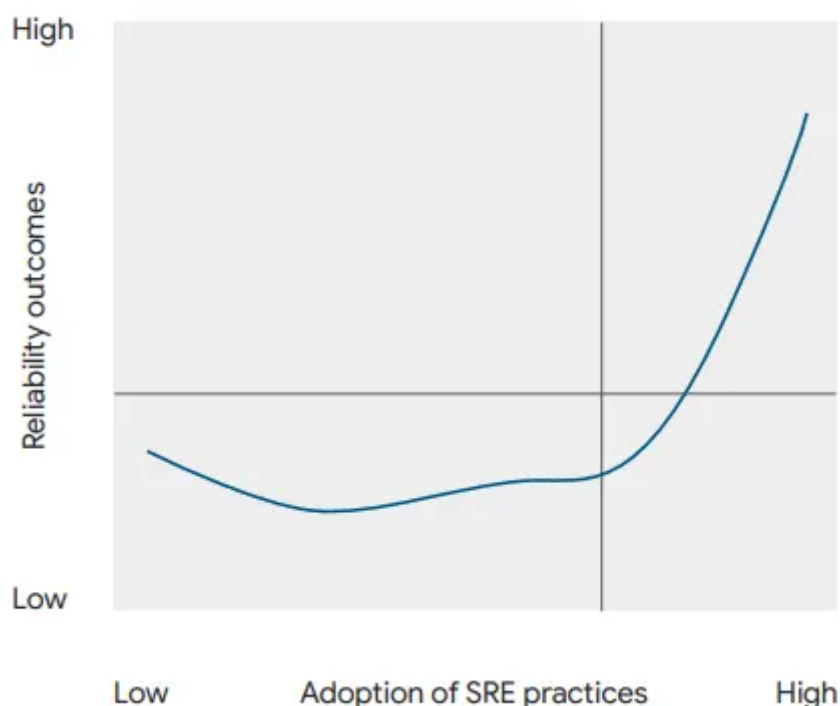
Reliability

Reliability is a metric that DORA added in after the first four metrics. Reliability measures how well a company meets the four metrics. According to the DORA 2022 report:

High performers who meet reliability targets are 1.4x more likely to use continuous integration

The DORA 2022 report suggests Site Reliability Engineering (SRE) adoption plays an important role in organizational success. SRE is an approach to operations that uses observed learning, cross-functional collaboration, automation and measurement. SRE is a way of adding reliability across a system.

Adopting SRE across a company is a journey, and the report indicates a 'J-curve' pattern to seeing results. In the initial adoption phase, there will be few results while teams learn new processes and implement systems. At a certain threshold, SRE yields tangible improvements to reliability as the curve trends upwards. See The J-curve below.



Using them together

Based on the surveys, the DORA report classifies each company on a scale of how well they meet each metric. The performance in each metric classifies companies into one of three categories: Low, Medium, and High. Based on the category, you can identify how well your company performs relative to other DevOps community performers.

Before 2022, there was an extra category named 'Elite.' This was, however, removed for the 2022 State of DevOps report as it found only three clusters in the data.

SDO metric	Low	Medium	High
Lead time	1-6 months	Weekly to monthly	1 day - 1 week
Deployment frequency	Monthly to biannually	Weekly to monthly	Multiple times per day
Change failure rate	46-60%	16-30%	0-15%
Mean time to resolve	1 week - 1 month	1 day - 1 week	< 1 day

11% of respondents were in the high category, 69% in the medium category, and 19% in the low category. There is a large gap between high and low performers, with high performers having 417x more deployments than low performers.

Case Studies and results

The DORA 2022 report cited four capabilities that predict organizational performance:

- Using version control
- Practicing Continuous Delivery
- Practicing Continuous Integration
- Have svstems based on a looselv coupled architecture



THE DEVOPS ENGINEER'S HANDBOOK ▾

scores.

SPACE framework

The SPACE framework is a way to understand productivity from the team level to the whole organization. The SPACE framework does this through five areas:

- **Satisfaction** - How fulfilled, happy, and healthy one is
- **Performance** - An outcome of a process

- **Activity** - The count of actions or outputs
- **Communication and collaboration** - How people talk and work together
- **Efficiency and flow** - Doing work with minimal delays or disruptions

The report suggests:

To measure developer productivity, teams and leaders (and even individuals) should capture several metrics across multiple dimensions of the framework - at least three are recommended

Below is a table from the report on some example metrics for each system. Each metric could proxy for more factors, so consider the ones you use carefully. You can use the SPACE framework at an individual, team, and system level, and each DORA metric could fit into one of the SPACE framework categories. The benefit of using the SPACE framework is that you tackle productivity from many angles rather than looking at a single metric.

Use any metric taken at the individual level as an aggregate for analysis of a larger context. That's because the SPACE framework should not be a developer report card. Rather than focus on the time taken for a developer to review a pull request, the information on review times should be an answer to a broader question.

How can you improve the distribution of code review age to ensure all code reviews are completed sooner, as this improves the flow of work?



NEXT ARTICLE

DevOps reading list

The table below is an example SPACE framework matrix, with metrics for each category. The metrics you choose can be different, and it's important to evaluate the broader questions you want to answer about your system.

Level	Satisfaction and wellbeing	Performance	Activity	Communication and collaboration	Eff
Individual	Developer satisfaction Retention Satisfaction with code reviews assigned Perception	Code review velocity	Number of code reviews completed Coding time Number of commits Lines of code	Code review score (quality or thoughtfulness) PR merge times Quality of meetings Knowledge sharing,	Cc tir Pr pe La int

Level	Satisfaction and wellbeing	Performance	Activity	Communication and collaboration	Eff
	of code reviews			Discoverability (quality of documentation)	
Team	Developer satisfaction retention	Code review velocity story points shipped	Number of story points completed	PR merge times Quality of meetings Knowledge sharing, discoverability (quality of documentation)	Cc tir ha
System	Satisfaction with engineering system (e.g.,CI/CD pipeline)	Code review velocity code review (acceptance rate) Customer satisfaction reliability (uptime)	Frequency of deployments	Knowledge sharing, discoverability (quality of documentation)	Cc tir Ve flc th

Conclusion

Metrics are useful in system observability by allowing you to assess the internal health of your system through external indicators. The DORA metrics are a set of performance metrics closely linked to DevOps success. The metrics are:

- Lead time for changes
- Mean time for recovery
- Deployment frequency
- Change failure rate
- Reliability

You can use these metrics together to see how your company performs relative to other successful DevOps companies.

DORA metrics give you a picture of overall DevOps system health. The SPACE framework allows you to look at productivity from an individual and team level and tackle productivity from multiple angles. The areas are:

- Satisfaction and wellbeing
- Performance
- Activity
- Communication and collaboration
- Efficiency and flow

DORA metrics and the SPACE framework help create ideas as part of your continuous improvement practice and offer insight into your DevOps performance.

[Terence Wong](#)

Thursday, December 8, 2022

Tags:

- [Metrics](#)
- [Improvement](#)

Categories:

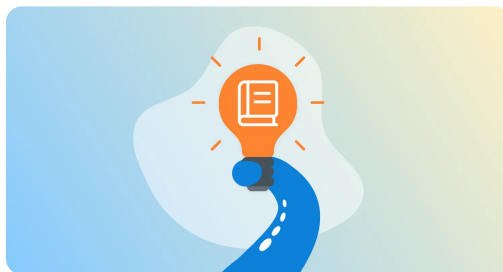
- [DevOps](#)

More resources



DevOps team structures

[Read more →](#)



DevOps reading list

[Read more →](#)



CATEGORIES

Continuous Delivery

DevOps

Platform Engineering

Site Reliability Engineering

TAGS

Automation

Books

CI-CD

Deployments

QUICK LINKS

[Octopus Slack Community](#)

Octopus Deploy

Copyright © 2022 Octopus Deploy

[Back to top](#) ^