

Site Reliability Engineering

Reliable operations of applications in an agile business world

T Systems

Let's power
higher performance

Content

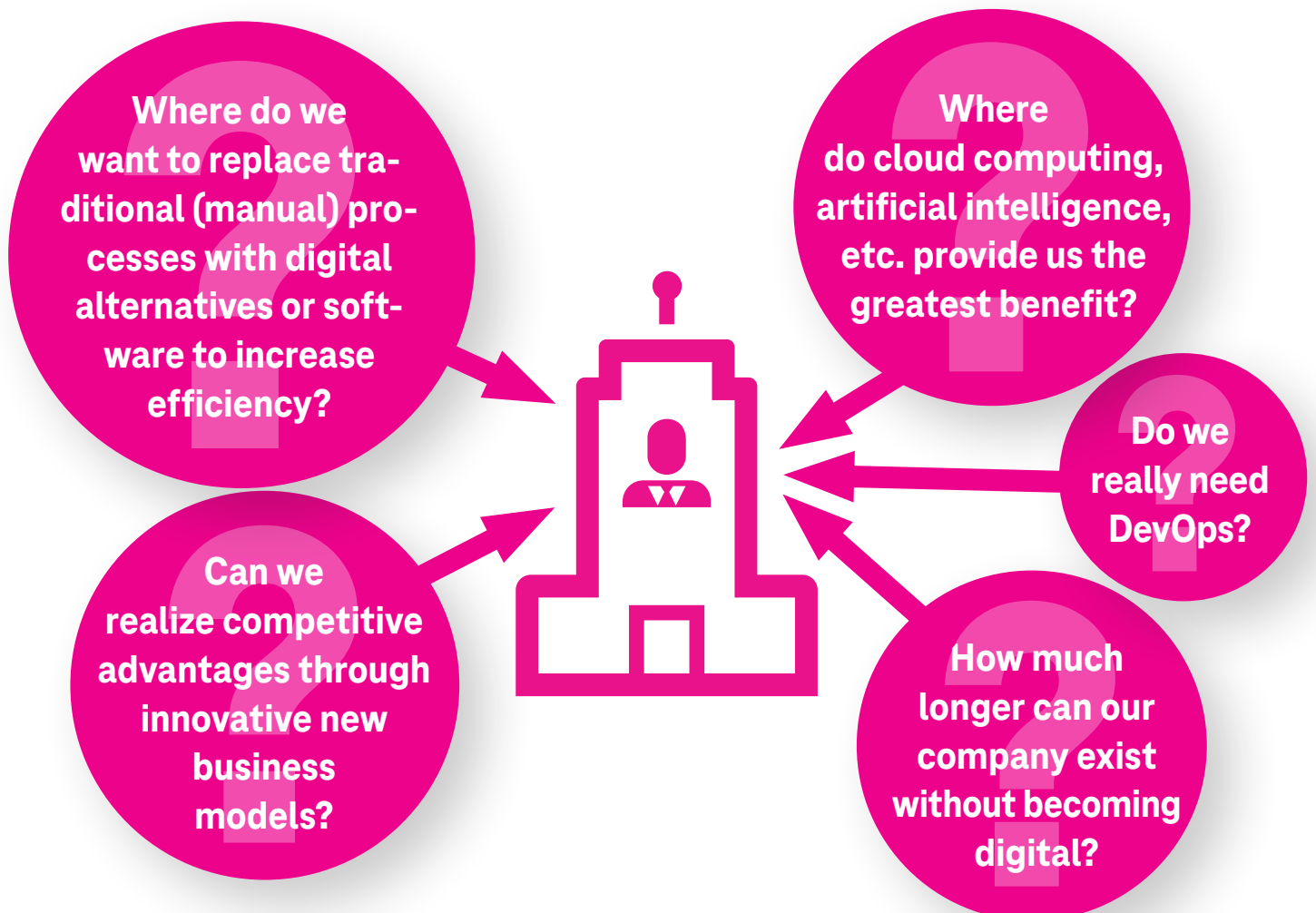
Business trigger, definition, meaning of DevOps for business	3
The status of DevOps	7
Site Reliability Engineering	10
SRE tools, Service Level Agreements	13
Service Level Objectives	14
Service Level Indicators	15
Benefits of SRE	16
Implementation of DevOps/SRE	17
Application example: business-critical applications at Deutsche Telekom	18
Conclusion	19
Contact/Publisher	20

”

**Every company today
is a digital company**

- Forrester, 2021 -

Even if the statement probably does not actually apply to all companies, it can be said: Hardly any company of a certain size can do without a digital strategy today. This includes a multitude of components and questions:





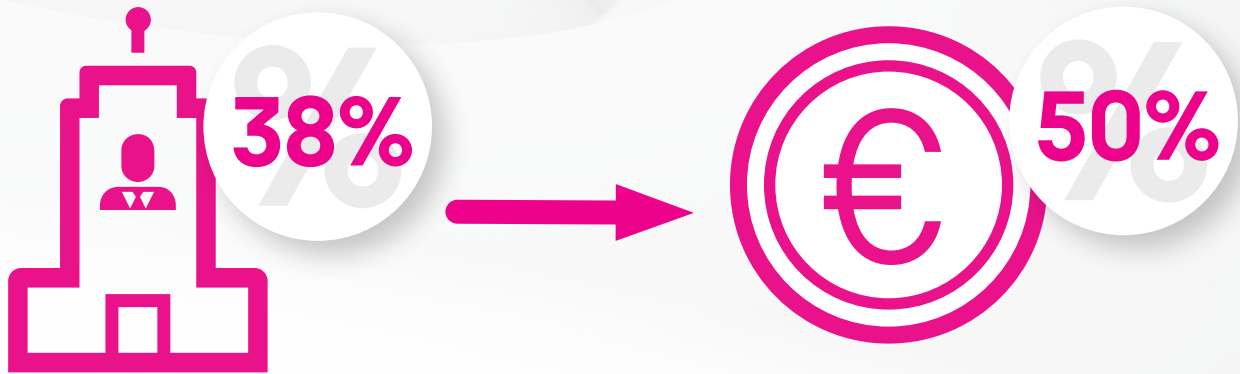
It is clear that size and industry expertise are no longer the sole success factors. In recent years, **speed** and **agility** have become the new business maxims. Today's companies are characterized by their ability to react quickly to market developments and to offer new services in shorter cycles. Fast companies are more successful and slow ones lag behind. Digitalization and digital skills play a decisive role.



In this environment, the discussion is still mainly about the cloud. In fact, **the cloud is only the** (albeit essential) **starting point for digitalization**. The majority of companies has already opted for the cloud. They are pursuing cloud-first strategies. And this is not only the way to go for enterprises: For example, cloud-first is also at the heart of the European Commission's cloud strategy (published in 2019). It is classified as a trailblazer for the overarching digital strategy of the European Commission.



Many industrial companies are aiming for a dramatic shift to digital software and services by 2023.



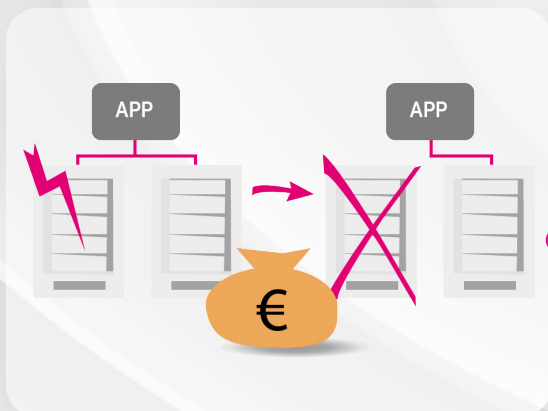
38% of the industrial companies will aspire to generate 50% of revenues until 2023 from digital technologies and services.



McKinsey, Four myths about building a software business, 04/2021

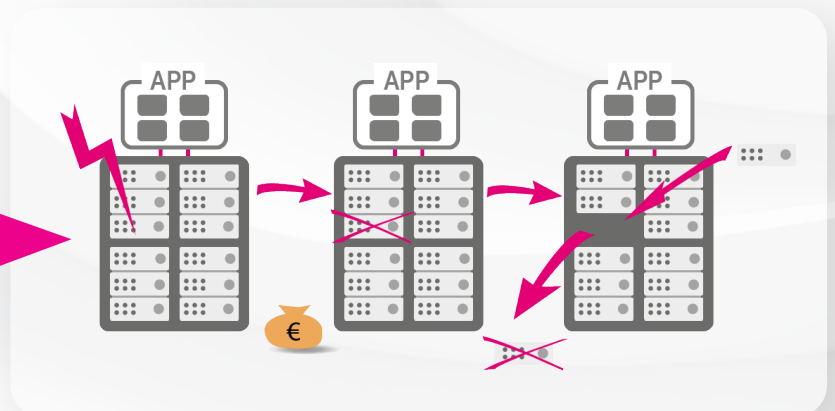
Mode 1

Availability at **infrastructure level**



Mode 2

Availability at **application level**



But cloud infrastructures and platforms are only "enablers". Companies that migrate applications to the cloud will initially not get much out of it – apart from the reduction of infrastructure costs and the use of services from the cloud provider ecosystem.

Differentiation in the cloud age does not arise on the infrastructure level, but on the application level – with the direct support of business processes. The cloud creates the basis for agility and speed, but the application level is at the heart of digital value creation. Consequently, companies cannot avoid developing modern software skills. In other words:

Enterprises who are in for cloud need to be in for DevOps as well

– and then evolve into a developer-centric organization.

The status of DevOps

Whether over-the-air updates for cars, immersive shopping experiences or AI-based quality control in production – more and more product managers are demanding and designing digital products based on software with which they (want to) achieve added value for customers. Regardless of the industry, these products are in a continuous development process. They are built into physical products or delivered directly as a virtual service via the Internet or mobile communications.

The competence of software delivery and operation (SDO) is thus becoming a critical success factor for competitive differentiation and market success. Therefore, DevOps or the DevOps capabilities of a company play an essential role. And DevOps goes far beyond a technical view: With DevOps, IT staff adapts the attitude of business managers as a new mindset. They are no longer just responsible for bits and bytes, but work towards customer acceptance of “their” product. This is one aspect of the cultural shift that comes with the cloud. DevOps combines business goals with a company's ability to develop and operate software.

DevOps principles

Until now, development and operations have been separate worlds. Different tools, methods and goals determined of the two domains. Companies that think cloud-native are using DevOps to close the gaps between development and operations. DevOps relies on five central maxims:

Breaking down
organizational
silos

Extensive
automation

Acceptance
of mistakes

Collection
of data

Small
development
steps

Rapid provision of added value for customers is a business manager's primary target. It results in customer satisfaction – the digital age's ultimate currency. Customer satisfaction, a classic business goal, is thus becoming a measure of IT success through DevOps. However, the relation "higher customer satisfaction through DevOps" will only work if DevOps is realized with a high degree of professionalism.

But which factors determine the quality of a company's software delivery and operation (SDO) capabilities? The Accelerate State of DevOps Report identifies **four factors**:

How quickly are code changes implemented?

LEAD TIME OF CODE CHANGES

How quickly can a service be restored if the new code is faulty?

TIME TO RESTORE SERVICE

How often is a changed code rolled out?

DEPLOYMENT FREQUENCY

How often do such failed code changes occur?

CHANGE FAILURE RATE

Although DevOps actually – as the name suggests – characterizes a comprehensive approach that sees the development and operation of software as an integrated task, it can be observed that – even among the frontrunners – up to

15%



DORA, DevOps Research & Assessment, Sept. 2021

of all rollouts are unsuccessful. The rate increases to up to

30%

Elite companies roll out new code several times a day.

The time for changes is **less than an hour** – as well as the time to restore a service.

Only

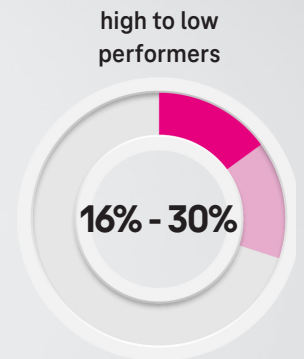
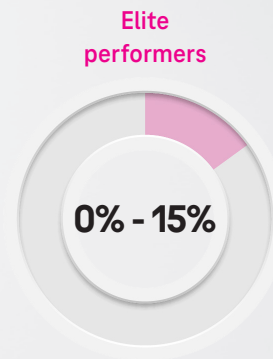
each 7th

rollout (or less) *fails*.

for companies in the following categories.

Change failure rate

"For the primary application or service you work on, which 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)?"



Thus, DevOps must also face the central SDO dilemma: Either code is delivered faster with lower quality and higher risks, or the risks are reduced and the quality increased, which means that the delivery times are longer (time to market slowed down).

How can this dilemma be solved?

This is, where Site Reliability Engineering comes into play ...

Site Reliability Engineering

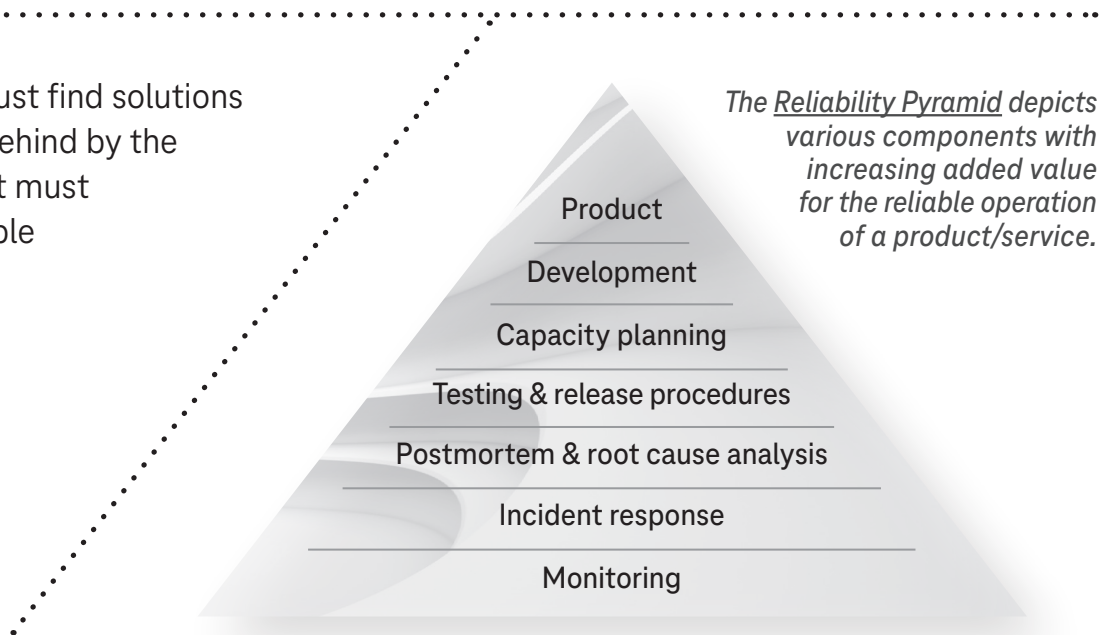
Lower service quality or reduced competitiveness – it is an unpleasant decision that companies have to make against this background.

The results of the above report suggest that in reality DevOps still has a strong focus on software **development**. With the definition of a fifth success factor "operational performance", the 2021 report for the first time took greater account of the aspect of **operating** cloud-native applications:



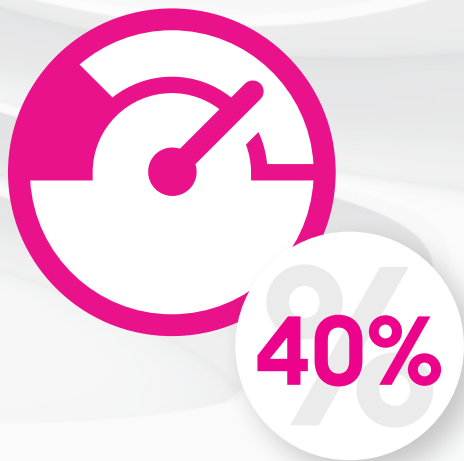
* against the background of constant changes

The operational side must find solutions in order not to be left behind by the pace of development. It must enable stable and reliable operation. **If this does not succeed, DevOps leads companies into a systemic problem.**



Operations teams arriving in the DevOps world have already recognized this challenge. Traditional operations teams that roll out an update or two each year are unprepared for the challenge. The cultural change triggered by the cloud thus is felt also behind the scenes, right in the middle of the IT department. If software developers become drivers of corporate development, **the operation of applications must also be rethought**. Operations teams also become drivers of business development with a responsibility for business success.

Site Reliability Engineering (SRE) is how operations teams are responding to the DevOps challenge. The operational challenges ahead are addressed by adopting engineering practices (from development). At the same time, operating teams break down the established silos via SRE, thus initiating an exchange and cooperation across different units and creating common goals. SRE uses the same maxims and tools that DevOps uses – this actually creates a common basis between operations and development. When operations teams live the SRE principles, this results in a significant increase in SDO performance. In 40 percent of all cases, excellent SRE leads to higher SDO performance and in 80 percent of all cases to better business contributions. **SRE thus makes a significant contribution to the success of DevOps practices.**



higher SDO performance

&



better business contributions

Site Reliability Engineering applies DevOps principles to operations. How SRE teams map DevOps principles:



Breaking down organizational silos

using identical tools for development and operations, shared responsibility for a product's success



Acceptance of errors

definition and control of an "error budget", early problem solving ("shift left")



Small development steps

validation of updates through (limited) canary (test) deployments, automated rollbacks



Extensive automation

identification of the most common manual activities/procedures, more than half of the working time is spent on the development of processes that reduce manual activities



Collection of data

use of Service Level Indicators and Objectives, telemetry and monitoring

SRE-Tools

SRE uses three key tools to drive success:



Service Level Agreements



Service Level Objectives



Service Level Indicators

The idea behind these three tools is to create a link between the operations teams' work and the company's business goals.



[Google Cloud, SRE fundamentals: SLIs, SLAs and SLOs](#)

Service Level Agreements

Service level agreements are a classic in IT service provision. They indicate a promise of performance over a certain period of time. A service provider providing an IT service and his customer agree on a performance target, usually stated as a percentage (e.g. 99.5 percent service availability during a month). Failure to meet the SLA usually has negative consequences for the company or entity responsible for the SLA – mostly penalties that are explicitly laid down in the SLA contract. The SLA is a powerful control tool for service delivery.

Service Level Objectives

Availability (of a service) is the central prerequisite for business success – the latter being the maxim of SRE. Availability in terms of SRE means that a system is able to fulfill its intended function at a certain point in time. This perspective can be used retrospectively (as reporting) and also as a forecast for the future. Service Level Objectives are **target values for performance fulfillment** within an SLA. The SLOs specify the agreements of the SLA. And they set a concrete expected value for the fulfillment of the business function for the teams providing the service, e.g. at what speed service requests in a shop system are processed. The Service Level Objectives serve as guidelines that must be met for SLA fulfillment (while also characterizing the quality of the operation depending on business metrics).

Without such an SLO, a company cannot make any statement about the reliability of a service. The requirements for costs and development speed of the service can then also be derived from the SLO. Excessive system availability generates unnecessary costs and effort. Too low availability will result in not achieving the business goals.



Service Level Indicators

Operating teams use Service Level Indicators to measure and thus check whether the SLO is being met or, overall, whether the SLA is being met. They take a look from the users' perspective and test the behavior of the system. The more often the behavior of the system meets the set expectations, the better. However, if the reliability of the system is below the agreed quality, the operations team must take countermeasures to achieve the SLA. The measurability of the quality of a service is – as laid down in the DevOps maxims – also of central importance for SRE. Organizations can also use cloud-internal tools to collect SLIs. At AWS, SLIs can be measured automatically using such a tool. It creates automated reports in real time.

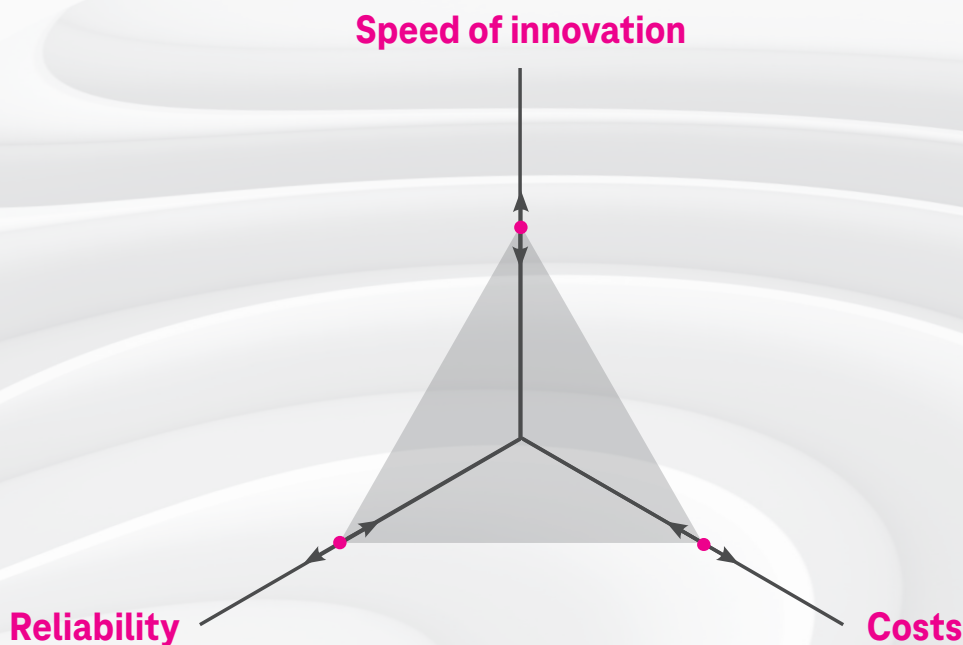
When redesigning a system, an agreement for SLI and SLO should also be provided as part of the system requirements. Companies that want to introduce SRE should define such SLI and SLO for their existing systems as a first (prioritized) step.

Benefits of SRE

With the establishment of the SLX methodology, operations teams and their customers gain transparency about the expected outcomes and deliverables. They get rid of unquantified discussions ("The user experience is bad"). Both sides – customer and service provider – are required to define business expectations in a measurable way and to check them regularly. This creates a framework for the successful operation of cloud-native applications.

SRE optimally complements DevOps initiatives in companies, so that there is actually a bidirectional exchange at eye level between development and operations. For example, experienced SRE teams reflect their operational expertise back into the development teams and help identify bottlenecks in the architecture. The application reliability and the development speed are neatly balanced against each other so that – measured against the goals of the business unit – the optimal result for the company is created. The business goals become the benchmark for development and operation. They specify where, if necessary, priorities should be set.

With SRE, the company has an answer to the increasing speed of new developments and can quantify the consequences of business goals. Due to the high level of automation in operation, CI/CD pipelines are optimally supported.





Implementation of DevOps/SRE

In principle, there are two basic options for companies that want to introduce DevOps and SRE. In the "make" case, companies establish cross-functional teams that take over the overall DevOps responsibility for a service/product. This is the textbook approach. In reality, however, it will be necessary to culturally and methodically bring together the teams from development and operations that have been operating independently and in their silos. Specific operating teams are then no longer necessary. However, the company must then ensure DevOps upskilling. It can also draw on external expertise. This can be done either through training courses or on-the-job training, for example by temporarily integrating external experts into the operating teams and passing on their knowledge there.

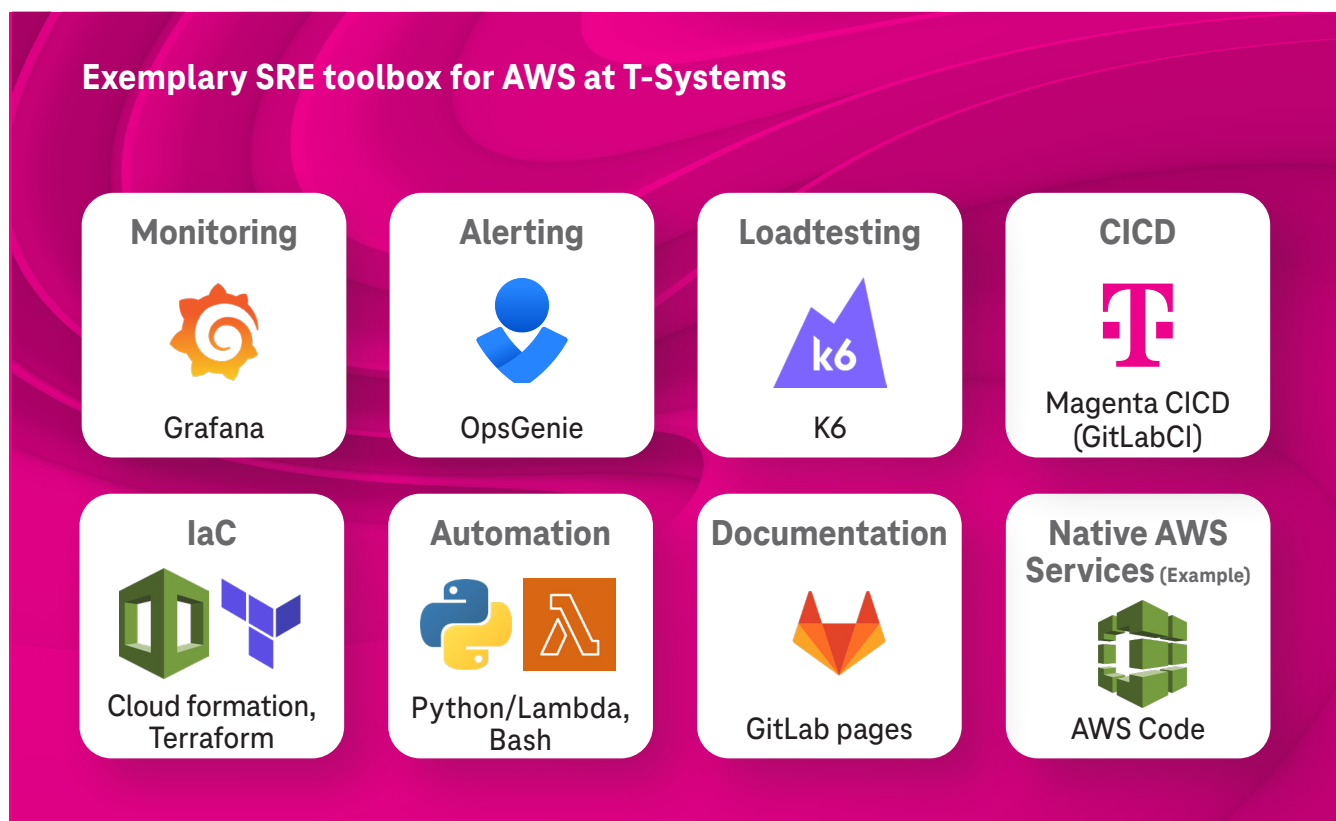
Alternatively, companies can also take a "buy" approach and completely outsource the Ops side of DevOps to an experienced partner. Here, too, close coordination between the organizationally separated units must be enabled. In addition, a uniform tool and method basis must be created, because the two units share responsibility for the reliability of the service. In this construct, the SLO becomes a decisive control instrument (see graphic p.14).

The reality in companies is between these two poles. Experienced service providers react with modular SRE support offers, in which they only take on partial tasks. These can be based on the Reliability Pyramid (p.10), for example. In any case, an active shift-left approach is essential – the operating team must be involved at an early stage. Integration into a Cloud Center of Excellence, as in the following example, is ideal.

Application example: business-critical applications at Deutsche Telekom

As part of its cloud-first strategy, Deutsche Telekom has moved some business-critical applications to the AWS Cloud. These are continuously developed in an agile model. At the same time, Deutsche Telekom wants the customer-facing services to be reliably available around the clock. The services contribute to two key business objectives: customer satisfaction (through self-service) and revenue (through online sales).

To ensure reliable operations, Deutsche Telekom IT's Cloud Center of Excellence has integrated an SRE team from T-Systems. The experts from T-Systems contribute Site Reliability Engineering tailored to the needs of the large company. They work together with the development units on the basis of common reliability and quality standards. Specifically, these are common CI/CD standards and SAFe as an agile methodology, but also the privacy and security assessment standards of Deutsche Telekom. T-Systems monitors the systems 24/7 and is responsible for incident and security incident response. To do this, the team uses a set of established methods (see box). The use of AWS-specific tools such as AWS Code is also possible thanks to the AWS know-how of the T-Systems SRE team.





Conclusion

Digitalization is paving the way to greater business agility. However, this consequently requires the use of agile development methods. But companies must not stop here either, because short development cycles challenge operations. How can the reliability of the services be ensured in addition to innovation in agile framework conditions? The answer lies in the adoption of Site Reliability Engineering that maps DevOps principles to operations. The introduction of SRE introduces a new culture of collaboration across classic unit boundaries and a high level of automation. Silo thinking is being replaced by common business goals. Only then will companies be able to reap the benefits of digitalization without a bitter aftertaste.



Do you want to introduce
SRE or get to know the
methodology better?

Contact us.

Contact

- Germany: 0800 33 09030
- International: 00800 33 090300
- Email: info@t-systems.com
- Internet: www.t-systems.com

Publisher

T-Systems International GmbH
Hahnstr. 43d
60528 Frankfurt am Main
Germany