SRE vs. DevOps vs. Platform Engineering: Differences Explained

This article explains SRE vs DevOps vs Platform Engineering, including similarities and differences, and more.

7 min. read · View original

SRE, DevOps and Platform Engineering are important concepts in today's world of software development. There are dedicated teams to manage these areas, each with a unique primary focus, set of responsibilities, tools and metrics used to gauge their performance requirements.

This article explains SRE, DevOps, and Platform Engineering, including similarities and differences, and, most importantly, how these teams help streamline modern software development, delivery, and maintenance processes.

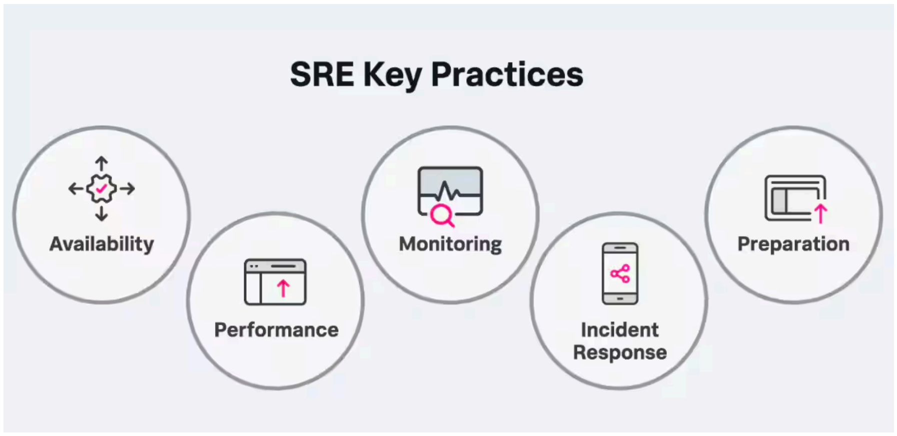
SRE overview

Site reliability engineering (SRE) is a practice that focuses on improving and maintaining the reliability of a software system. It utilizes software tools and automated tasks like application monitoring and reliability tasks to accomplish that.

In 2003, Google implemented SRE as a solution to the challenges associated with managing their large-scale and complex software systems. SRE greatly benefits teams, improving their collaboration, productivity and customer experience. The responsibilities of SRE teams include:

These teams work closely with development teams throughout the lifecycle and provide solutions for underlying system-related issues, such as bugs in software pipelines and automated jobs. They also help automate routine tasks to improve the productivity of developers.

(Learn about being an SRE & go-to SRE metrics.)



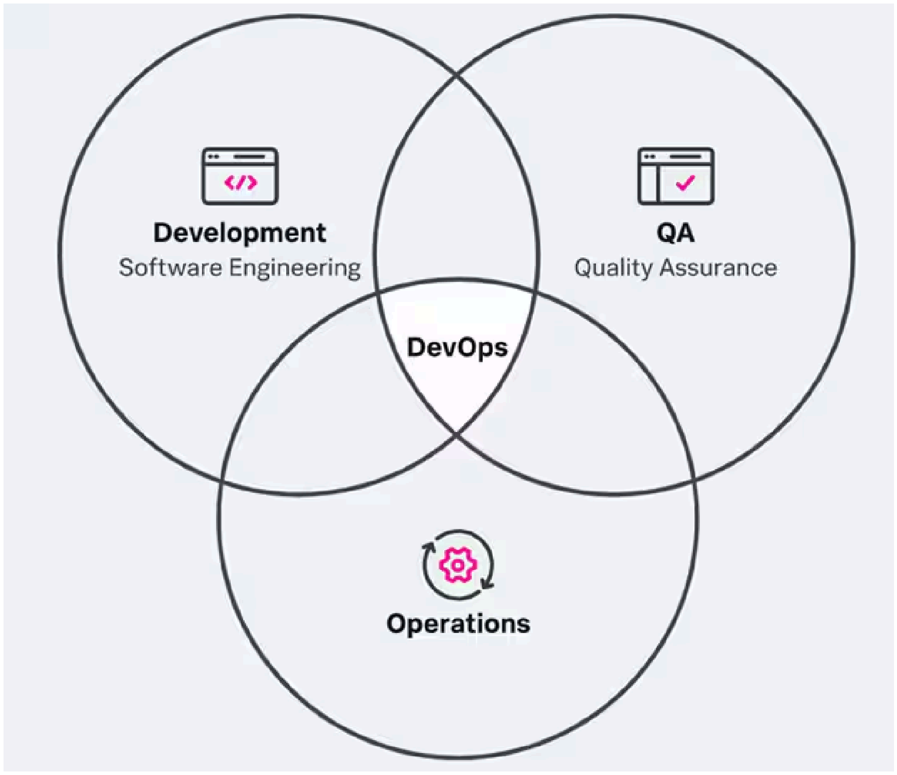
DevOps overview

For many folks, DevOps represents a major cultural shift in organizations. Traditionally, development and operations teams functioned separately. This siloed culture often led to issues like low-quality software and delays in software delivery. DevOps culture breaks down this siloed nature of development and operations teams by combining the tasks of two teams.

DevOps teams work in collaboration to automate and streamline the software development process. Also, DevOps greatly benefits teams by improving collaboration, communication, software delivery speed and quality. Responsibilities of DevOps engineers include:

- Prioritizing communication and collaboration.
- Automating processes.
- System monitoring.
- Optimizing system performance.
- Troubleshooting issues.

(Read about the state of DevOps today & which DevOps certifications to earn.)



Platform Engineering overview

Platform engineering is a rising discipline in today's cloud-native era. It aims to build toolchains and workflows covering the operational needs of the entire software development lifecycle, enabling self-service infrastructure capabilities.

Platform engineers & PE teams might focus on developing things like build tools, version control systems, and automated testing frameworks. They also build some workflows, such as CI/CD, alerting, and deployment workflows. These processes help software developers build and deliver software more efficiently. Platform engineers are responsible for:

Ultimately, platform engineering's aim is to solve issues that arise from poorly adopted SRE and DevOps practices.

DevOps vs. SRE vs. Platform Engineering

OK, so we've got a brief understanding of each of these concepts. SRE and DevOps aren't inherently contrasting ideas, though Platform Engineering does arise in response to common SRE and DevOps challenges or poor implementation.

Now let's look at these disciplines in a little more detail.

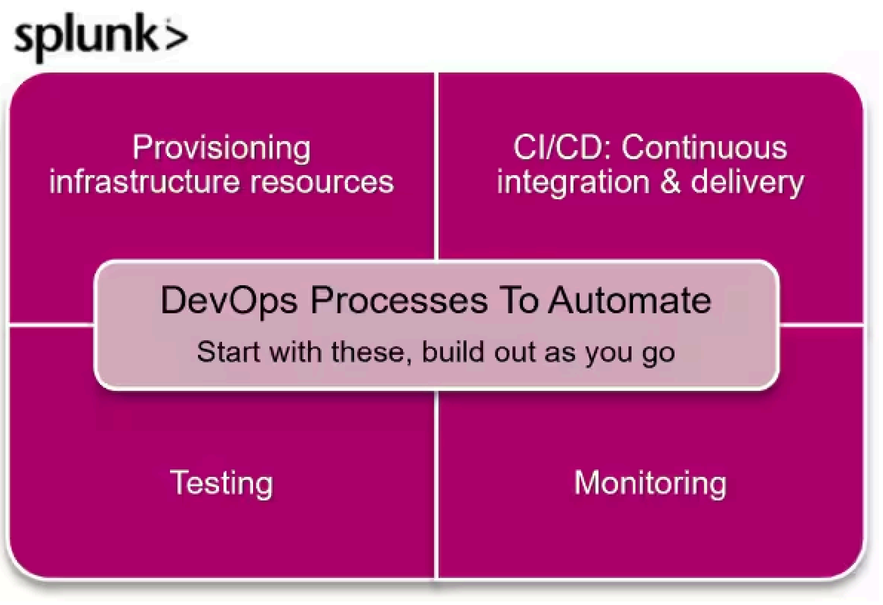
SRE vs. DevOps: Comparing, contrasting

What are the similarities between DevOps and SRE? SRE and DevOps teams have a lot in common:

- **Focus on automation**. SRE and DevOps both encourage automation, communication, and collaboration. Both teams bear responsibilities like monitoring, optimizing system performance, and troubleshooting issues.
- **Communication & collaboration**. Both DevOps and SRE encourage communication and collaboration between development and operations teams as a core principle. It helps them deliver high-quality and reliable software.
- **Tools.** SRE and DevOps teams are responsible for production monitoring and troubleshooting. They frequently use log analysis and monitoring tools like Splunk, Grafana, and NewRelic to

identify issues and improve the performance of software systems.

- **Metrics.** Both SRE and DevOps teams monitor metrics that reflect the behavior of applications and systems. Even if there are separate metrics, some metrics are useful for both teams. For example, response times, error rates, and other failure metrics help detect and resolve issues before they impact users.
- **Continuous improvement.** Both SRE and DevOps focus on the requirement of continuous improvement to enhance reliability, efficiency, and performance. This involves feedback loops, regular retrospectives, and iterative enhancements, ensuring that operational practices and delivery pipelines are evolving.


splunk>
Provisioning infrastructure resources
CI/CD: Continuous integration & delivery
DevOps Processes To Automate
Start with these, build out as you go
Testing
Monitoring

But how SRE and DevOps differ is quite illuminating:

Primary focus

Of course, there are some primary differences between them, too. The first major difference is the breadth of the focus. DevOps focuses on the entire software development process, while SRE narrowly focuses on the reliability and scalability of a system. Of course, in SRE's case, that narrow focus can have a significant berth in practice, as system reliability can touch a lot of disparate areas.

Cultural changes

DevOps breaks down silos between the development and operations teams, facilitating a collaborative and non-siloed culture. The main focus of SRE is to establish a culture of reliability and accountability.

Incident response

DevOps teams focus on preventing incidents from occurring in the first place through tasks such as automated software development, testing and proactive monitoring. In contrast, SREs focus on investigating the root cause of incidents and implementing measures to prevent them from happening again.

(Learn about incident response & incident management.)

Metrics

DevOps teams focus on DORA metrics such as deployment frequency, lead time for changes, mean time to resolution (MTTR), and change failure rate. In contrast SRE teams focus on metrics such as latency, traffic, uptime, error rates, and service level agreements (SLAs).

(Check out the ultimate guide to DevOps metrics.)

DevOps vs. Platform Engineering: Similarities & differences

- **Automation.** DevOps and platform engineering both rely on automation. Platform engineering leverages automation in CI/CD, alerting and deployment workflows. DevOps also utilizes automation in testing, CI/CD, monitoring, alerting and incident response.
- **Communication & collaboration**. DevOps and platform engineering promote communication and collaboration between development and operations teams to deliver high-quality and reliable software.
- **Infrastructure management**. Both teams focus on infrastructure management. DevOps engineers manage the complete infrastructure, while platform engineers focus on the platform layer specifically, which includes the underlying software and tools that support application development and delivery.

Some people argue that platform engineering is an evolution of DevOps engineering. However, the two roles differ primarily in terms of their main focus and the tools used to carry out their day-to-day tasks.

Primary focus

DevOps teams prioritize delivering the technical features of an application as fast as possible with high quality through task automation, communication, and collaboration. Platform engineering teams, on the other hand, focus on identifying the operational needs of development teams and building platforms, toolchains, and workflows to facilitate them. In other words, platform engineering focuses on building and maintaining a platform for software development rather than the development itself.

Tools

DevOps tools help automate monitoring and alerting while streamlining software development, deployment, and management. Examples include continuous integration and delivery (CI/CD) tools like Jenkins and GitLab, and collaboration and communication tools like Slack and JIRA. On the other hand, platform engineering tools automate infrastructure resource provisioning, deployment, and management. Examples include Kubernetes, Terraform, Ansible, AWS Code pipeline, and gitStream.

(Learn more about monitoring, telemetry & observability for systems.)

SRE vs. Platform Engineering

The similarities between SRE and Platform Engineering include:

- **Automation.** For example, both disciplines use Infrastructure as Code (IaC) tools to

automate the provisioning and management of infrastructure resources.
- **Focus on availability, reliability, scalability.** SRE and Platform Engineering ensure availability, reliability, and scalability through common tasks like monitoring and alerting. It helps detect and resolve issues before they impact the development teams and end users. Additionally, both roles are responsible for designing and implementing scalable architectures.
- **Communication & collaboration.** Both SRE and Platform Engineering require collaboration between many teams, such as developers, testers, and product or project management teams.

Now onto the differences:

Primary focus

SRE teams primarily focus on the reliability and scalability of a system through tasks like monitoring, troubleshooting, and incident response. On the other hand, Platform Engineering prioritizes building toolchains and workflows required for software development, enabling self-service infrastructure capabilities.

Tools

SRE teams rely heavily on monitoring and alerting tools such as NewRelic, Prometheus, and Grafana and incident response tools like PagerDuty. In contrast, platform engineers are responsible for managing infrastructure tools like container orchestration tools, infrastructure management tools like CrossPlane and infrastructure provisioning tools.

Metrics

SRE teams focus on monitoring metrics related to latency, traffic, uptime, and error rates to ensure the reliability and availability of systems. In contrast, Platform Engineering teams measure:

- **Infrastructure productivity** like lead time, deployment frequency, etc.)
- **Stability** like MTTR, change failure rate, etc.
- **Efficiency metrics** like resource allocation to ensure efficient management of infrastructure.

How do SRE, DevOps & Platform Engineering work together?

Although each discipline has its own set of responsibilities, certainly their work overlaps. Today, all three roles are interconnected to ensure smooth software development, delivery, and production systems are running without issues.

All three roles promote close collaboration and communication between developers, operations teams, and stakeholders to ensure everyone is aligned on their business requirements, goals, and issues by accommodating each other's needs.

- **SREs work closely with DevOps engineers** to ensure that most of their development, testing, and deployment tasks are automated. They ensure that CI/CD pipelines are optimized for speed and efficiency.

- **DevOps engineers also collaborate with Platform Engineers** to ensure they have the required infrastructure, workflows, and tools to support the development and ensure smooth deployment. Both roles collaborate to create flexible and scalable platforms that meet the requirements of development teams.
- **Platform Engineers, DevOps, and SREs** also collaborate while responding to incidents and post-incident analysis. Incident response efforts are led by SREs, whereas platform engineers and DevOps help in finding out the root cause, providing fixes, and implementing measures to prevent similar issues.
- **SRE teams partner with Platform Engineers** to establish reliable, scalable, and secure infrastructure. Additionally, they implement the necessary monitoring and alerting tools to identify and respond to issues faster.
- **SREs, Platform Engineers, and DevOps engineers** work together to implement infrastructure as code (IaC) best practices. The goal is the management and provisioning of infrastructure resources, thereby improving scalability, reducing errors, and maintaining consistency.
- **DevOps engineers, SREs, and Platform Engineers** get involved in knowledge-sharing initiatives and cross-training, fostering a culture of collaboration and learning. This enhances cohesiveness among the team members. They can collectively deal with complex challenges and innovatively drive development and operations.

Conclusion

The current fast-paced software development environments demand collaboration among SRE, DevOps, and Platform Engineering to meet various requirements for smoother development, deployment, and improved production systems. While SRE teams mainly focus on improving the reliability of a software system, DevOps teams streamline software development and deployment through close collaboration with operations teams.

On the other hand, platform engineering teams facilitate infrastructure by providing toolchains and workflows required for the development teams. Automation and monitoring are common tasks for all three roles, using similar tools.

So we can say: that these roles are distinct, and their responsibilities may overlap.