SRE VS. DEVOPS: WHAT'S THE DIFFERENCE?

With the growing complexity of application development, organizations are increasingly adopting methodologies that enable reliable, scalable software.

DevOps and site reliability engineering (SRE) are two approaches that enhance the product release cycle through enhanced collaboration, automation, and monitoring. Both approaches utilize automation and collaboration to help teams build resilient and reliable software—but there are fundamental differences in what these approaches offer and how they operate.

So, this article delves into the purpose of DevOps and SRE. We'll look at both approaches, including benefits, differences, and key elements.

(This article is part of our DevOps Guide. Use the right-hand menu to navigate.)

DevOps basics

DevOps is an overarching concept and culture aimed at ensuring the rapid release of stable, secure software. DevOps exists at the intersection of Agile development and Enterprise Service Management (ESM) practices.

Early methodologies involved development and operations teams working in silos, which led to slower development and unstable deployment environments. To solve this, the DevOps methodology integrates all stakeholders in the application into one efficient workflow, which enables the quick delivery of high-quality software.

By allowing communication and collaboration between cross-functional teams, DevOps also enables reliable service delivery and improved customer satisfaction.

DevOps practices & methods

DevOps practices are based on continuous, incremental improvements bolstered by automation. While full-fledged automation is rarely possible, DevOps methodology focuses on the following elements:

- **Continuous delivery and integration (CI/CD).** Using CI/CD pipelines, you can seamlessly connect processes and practices while using automation for fast and frequent updates and code releases. You also have continuous monitoring and deployment for consistent code across software versions and deployment environments.
- **Infrastructure as code (IaC).** With the abstraction of IT infrastructure, you can manage software engineering and provisioning automatically so your team efficiently tracks changes, monitors infrastructure configurations, and can roll back changes if there are undesired or unintended effects.
- **Automated testing.** Code is automatically and continuously tested while it is being written or updated. By eliminating the bottlenecks associated with pre-release testing, the continuous mechanism speeds up deployment.
- **Framework integration.** DevOps works with frameworks that enable comprehensive automation, along with faster delivery, efficiency, and enhanced collaboration. DevOps is ideal with the Scrum framework that designates project roles and defines workflows, the Kanban framework for workflow management, and Agile for rapid, frequent, and iterative updates through flexible and shorter development cycles.

Site reliability engineering (SRE) basics

SRE provides a unique approach to application lifecycle and service management by incorporating various aspects of software development into IT operations.

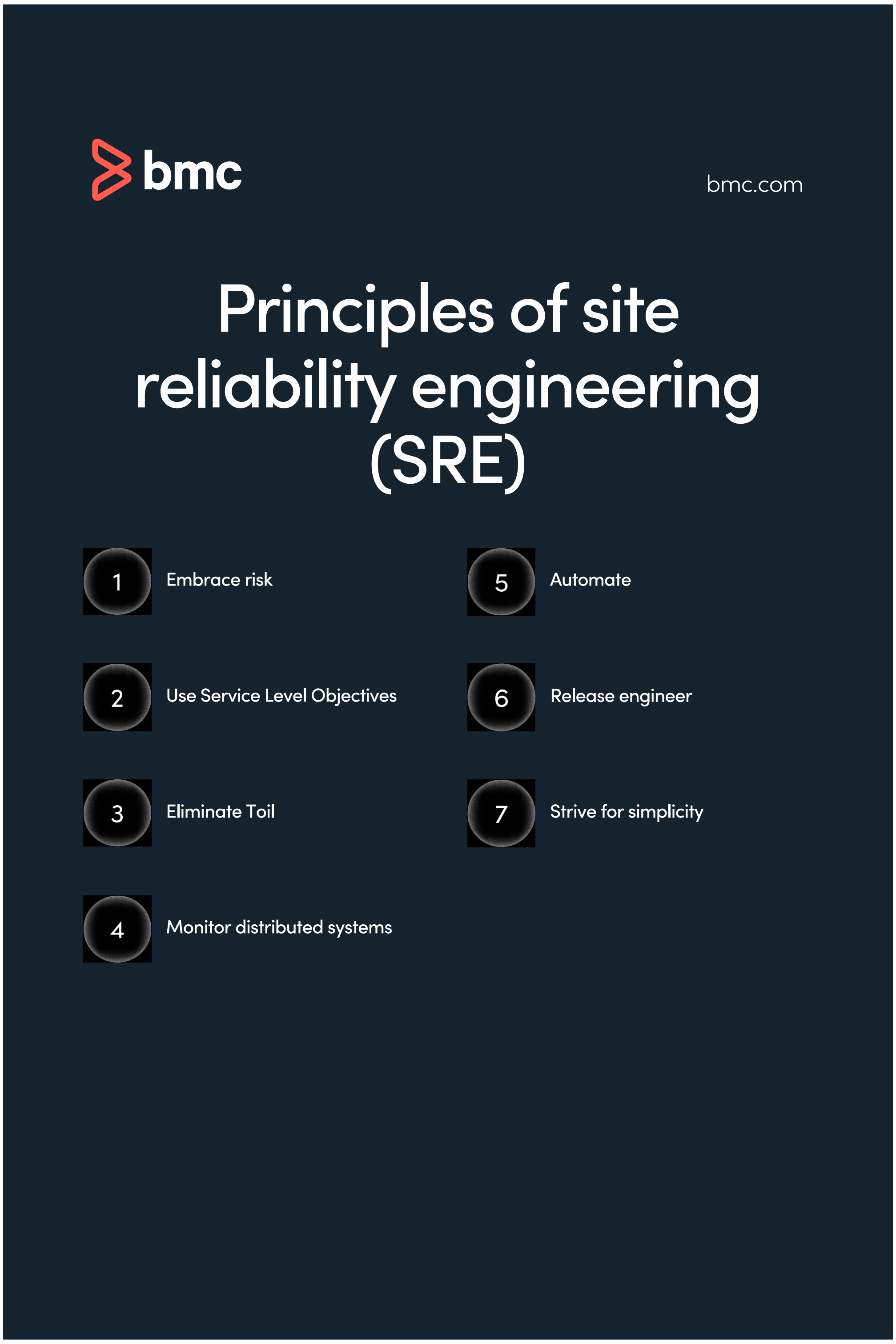
SRE was first developed in 2003 to create IT infrastructure architecture that meets the needs of enterprise-scale systems. With SRE, IT infrastructure is broken down into basic, abstract components that can be provisioned with software development best practices. This enables teams to use automation to solve most problems associated with managing applications in production.

SRE uses three Service Level Commitments to measure how well a system performs:

- Service level agreements (SLAs) define the required reliability, performance, and latency of the system as desired by end users.
- Service level objectives (SLOs) target values and goals set by SRE teams that should be met to satisfy SLAs.
- Service level indicators (SLIs) measure specific metrics and aspects that show how much a system conforms to the SLOs. Typical SLIs include request latency, system throughput, lead

time, development frequency, mean time to restore (MTTR), and availability error rate.

Key principles of SRE include:



Principles of site reliability engineering (SRE)

1 Embrace risk
2 Use Service Level Objectives
3 Eliminate Toil
4 Monitor distributed systems
5 Automate
6 Release engineer
7 Strive for simplicity

What does a Site Reliability Engineer do?

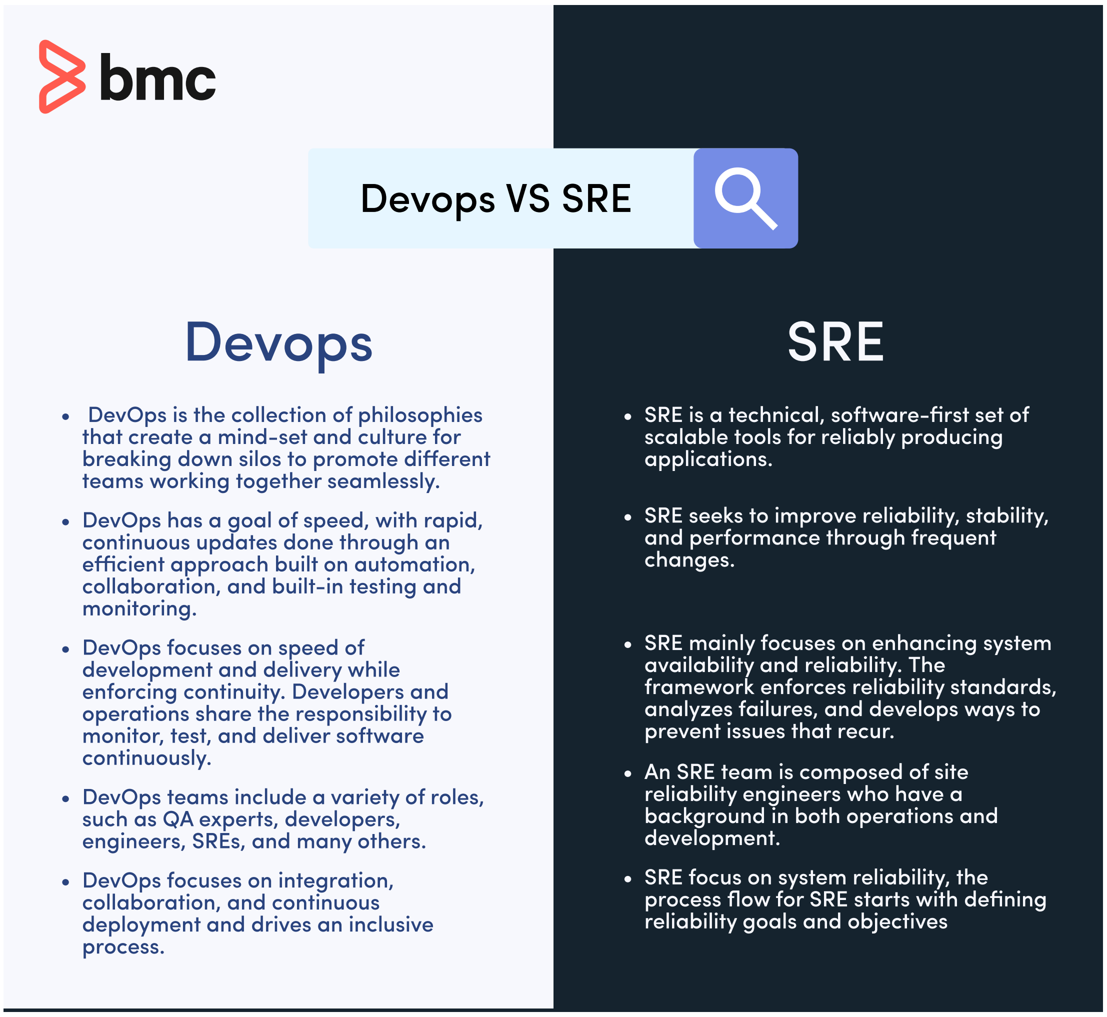
SRE essentially creates a new role: the site reliability engineer. An SRE is tasked with ensuring seamless collaboration between IT operations and development teams through the enhancement and automation of routine processes. Some core responsibilities of an SRE include:

- Developing, configuring, and deploying software to be used by operations teams
- Handling support escalation issues

- Conducting and reporting on incident reviews
- Developing system documentation
- Change management
- Determining and validating new features and updates

Differences between SRE and DevOps

Both methodologies enforce minimal separation between development and operations teams. But DevOps focuses more on a cultural and philosophical shift, while SRE is more pragmatic and practical.



Devops VS SRE

Devops

- DevOps is the collection of philosophies that create a mind-set and culture for breaking down silos to promote different teams working together seamlessly.
- DevOps has a goal of speed, with rapid, continuous updates done through an efficient approach built on automation, collaboration, and built-in testing and monitoring.
- DevOps focuses on speed of development and delivery while enforcing continuity. Developers and operations share the responsibility to monitor, test, and deliver software continuously.
- DevOps teams include a variety of roles, such as QA experts, developers, engineers, SREs, and many others.
- DevOps focuses on integration, collaboration, and continuous deployment and drives an inclusive process.

SRE

- SRE is a technical, software-first set of scalable tools for reliably producing applications.
- SRE seeks to improve reliability, stability, and performance through frequent changes.
- SRE mainly focuses on enhancing system availability and reliability. The framework enforces reliability standards, analyzes failures, and develops ways to prevent issues that recur.
- An SRE team is composed of site reliability engineers who have a background in both operations and development.
- SRE focus on system reliability, the process flow for SRE starts with defining reliability goals and objectives

Essence

SRE is a technical, software-first set of scalable tools for reliably producing applications. It is focused on creating a set of practices and metrics that improve collaboration, system reliability, and service delivery. SRE measures reliability, using SLOs to sustain performance.

Rather than practical tools, DevOps is the collection of philosophies that create a mind-set and culture for breaking down silos to promote different teams working together seamlessly. It seeks specifically to connect development and operations to collaborate and automate, leading to higher quality, faster releases and reduced failures.

Goal

Both SRE and DevOps aim to bridge the gap between development and operations, but their goals are different.

SRE seeks to improve reliability, stability, and performance through frequent changes. It prescribes approaches that balance reliability against innovation, use automation, and offer tools for proactive problem-solving.

DevOps has a goal of speed, with rapid, continuous updates done through an efficient approach built on automation, collaboration, and built-in testing and monitoring.

Focus

SRE mainly focuses on enhancing system availability and reliability. The framework enforces reliability standards, analyzes failures, and develops ways to prevent issues that recur.

DevOps focuses on speed of development and delivery while enforcing continuity. Developers and operations share the responsibility to monitor, test, and deliver software continuously.

Team structure

An SRE team is composed of site reliability engineers who have a background in both operations and development. They are, however, separate from development teams, working in a support function.

DevOps teams include a variety of roles, such as QA experts, developers, engineers, SREs, and many others. They work together through the entire software lifecycle, looking for ways to enhance teamwork and automate workflows.

Process flow

Because of its focus on system reliability, the process flow for SRE starts with defining reliability goals and objectives, then monitors and observes, reporting incidents as they happen and responding. SRE process flows include error budgeting, where if reliability slips, new features are delayed until stability can be reestablished. Automating tasks reduces manual work. The final steps in a SRE workflow are running postmortems and learning from failures, as well as planning for capacity needs to prevent future failures.

DevOps focuses on integration, collaboration, and continuous deployment and drives an inclusive process. It begins with planning features and the system design, and then moves quickly into developing test code and using CI/CD pipelines to automate builds and code integration. Testing for quality leads to release into production and deployment. Monitoring systems produce data and insights that feed continuous improvement.

Tools

In some cases, SRE and DevOps use the same tools. For version control, both use Git, GitHub, GitLab, and Bitbucket. For CI/CD, both use Jenkins, GitHub Actions, GitLab, and CI. DevOps also uses Azure DevOps Pipeline. For containerization, both use Kubernetes, with DevOps also using Docker.

DevOps uses Terraform, Ansible, Chef, and Puppet for configuration and IaC. Prometheus, Grafana, ELK, and Datadog are DevOps monitoring and observability tools. It uses PagerDuty and Opsgenie for incident management.

SRE also uses Terraform for configuration and Iac, adding Helm and Kustomize. It uses Kubernetes and Istio for containerization. SRE uses the same monitoring and observability tools as DevOps, adding also OpenTelemetry and Sentry. Incident management is done with PagerDuty, VictorOps, and Blameless. SLIs and SLOs are the key to error budgeting and reliability.

Similarities between SRE and DevOps

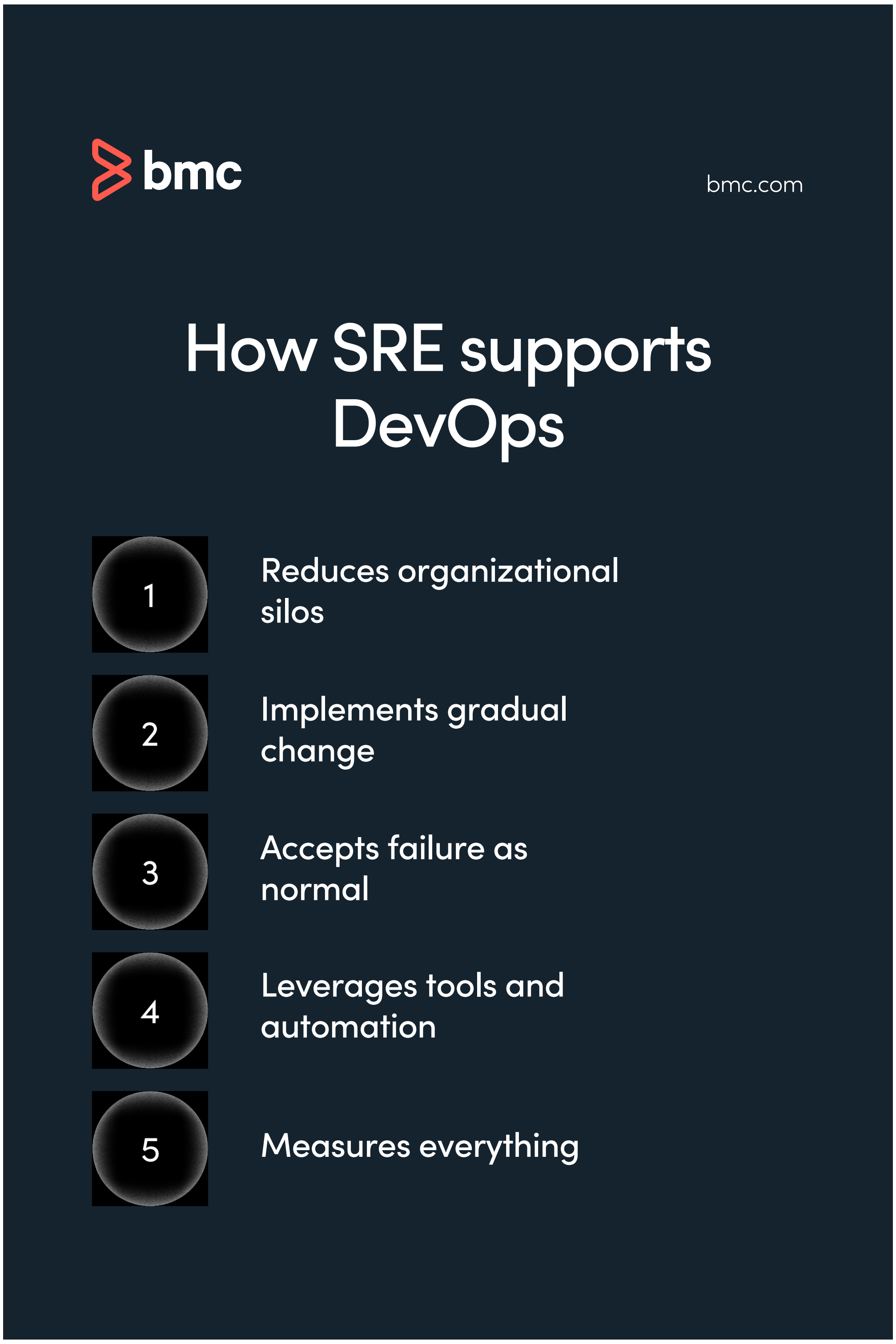
While the differences between SRE and DevOps are many, so are the similarities:

- Both seek to deliver reliable software more quickly and make use of automation.
- Automation and continuous improvement are supported.
- The frameworks both break the siloes of operations working separately from development.
- Both include monitoring as a key workflow step in improving software.
- Incident responses feed continuous improvement.
- IaC is a shared concept for managing and scaling environments.

How SRE supports DevOps principles & philosophies

SRE and DevOps are not competing methodologies. That’s because SRE provides a practical approach to solving most DevOps concerns.

In this section, let’s explore how teams use SRE to implement the principles and philosophies of DevOps:


bmc
bmc.com
How SRE supports DevOps
1
Reduces organizational silos
2
Implements gradual change
3
Accepts failure as normal
4
Leverages tools and automation
5
Measures everything

Reducing organizational silos

DevOps works to ensure that different departments/software teams are not isolated from each other, ensuring they all work towards a common goal.

SRE enables this by enforcing the ownership of projects between teams. With SRE, every team uses the same tools, techniques, and codebase to support:

- Uniformity
- Seamless collaboration

Implementing gradual change

DevOps embraces slow, gradual change to enable constant improvements. SRE supports this by allowing teams to perform small, frequent updates that reduce the impact of changes on application availability and stability.

Additionally, SRE teams use CI/CD tools to perform change management and continuous testing to ensure the successful deployment of code alterations.

Accepting failure as normal

Both SRE and DevOps concepts treat errors and failure as an inevitable occurrence. While DevOps aims to handle runtime errors and allow teams to learn from them, SRE enforces error management through Service Level Commitments (SLx) to ensure all failures are handled.

SRE also allows for a risk budget that allows teams to test the limits of failure for reevaluation and innovation.

Leveraging tools & automation

Both DevOps and SRE use automation to improve workflows and service delivery. SRE enables teams to use the same tools and services through flexible application programming interfaces (APIs). While DevOps promotes the adoption of automation tools, SRE ensures every team member can access the updated automation tools and technologies.

Measuring everything

Since both DevOps and SRE support automation, you'll need to continuously monitor the developed systems to ensure every process runs as planned.

DevOps gathers metrics through a feedback loop. On the other hand, SRE enforces measurement by providing SLIs, SLOs, and SLAs to perform measurements. Since Ops are software-defined, SRE monitors toil and reliability, ensuring consistent service delivery.

Summing up DevOps vs. SRE: Which is better?

In comparing SRE and DevOps, neither is better than the other. In fact, they can work together: 50% of companies using DevOps for speed and efficiency have also adopted SRE to improve reliability.

SRE offers tools and techniques that complement DevOps philosophies and practices. SRE applies software engineering principles to managing system reliability and scaling operations. DevOps brings a collaborative philosophy and structured approach for delivering software quickly and efficiently.

The goal of both methodologies is to enhance the IT ecosystem. DevOps improves the application lifecycle, and SRE improves the operations.

"Gain insight to the capabilities necessary to attract top SRE talent and make them successful in your organization with artificial intelligence for operations (AIOps) and artificial intelligence for service management (AISM) capabilities."

Related reading

- BMC DevOps Blog
- The Complete DevOps Certifications Guide
- The State of DevOps: A Report Roundup
- SRE vs ITOps: Are SRE & IT Operations The Same Thing?
- The State of SRE Today
- Implementing GitOps To Deliver Cloud NativeApplications