CMS WiRE

Weighing The Differences Between DevOps and SRE: Which One is Right for You?

By Kaya Ismail | *Jan 11, 2019*

Follow 2,139 followers

📷 PHOTO: SHUTTERSTOCK

The average consumer demands speed, convenience and reliability from every digital experience. These demands have spurred brands to adopt leaner and more efficient software development and maintenance practices — namely DevOps and Site Reliability Engineering (SRE). Both could be described as the distant cousins of one another, as both methodologies share similar core principles and objectives.

Here we explore the differences between DevOps and SRE with the assistance of leading industry experts and practitioners.

What Is DevOps?

DevOps is an approach to software development that follows either lean or agile principles. The primary focus of DevOps is to enable continuous delivery with a frequent release rate and an automated approach to application development. It achieves this by fostering a collaborative working environment.

"DevOps brings the traditionally separate teams of development and operations under one roof to improve upon collaboration, communication, integration and software releases," said Davy Hua, head of DevOps at Shiftleft. "This is accomplished by the focus on end-to-end automation of builds and deployments as well as effectively managing the entire infrastructure as code."

Related Article: 7 Key Principles for a Successful DevOps Culture

What Is Site Reliability Engineering (SRE)?

SRE is a discipline that incorporates the various aspects of software development and applies it to issues and tasks in IT operations specifically. The main objective of SRE is to develop a highly reliable and ultra-scalable software application or system.

The concept of SRE first came about in 2003. Google's Ben Treynor, who came up with the idea, noticed that no architectural method had thus far managed to meet the requirements of developing large-scale applications or systems. SRE enables developers to create a framework to support the build of these large-scale applications.

Drew Farnsworth, partner and designer at Green Lane Design, shared this, "I generally like to think as SRE as a system wherein development controls operations. This is a system where the environment is broken down to the most basic components of the IT stack and rolled out with the best practices baked into the hardware."

Hua added that the prime focus of SRE is to "completely automate most, if not all, of the tasks to ensure reliability in the systems."

As for resolving operation failures, Hua mentioned these issues can be "remedied by strategically adding measurable metrics via the software side."

Related Article: Agile vs DevOps: What's the Difference?

What Are the Similarities and Differences Between DevOps and SRE?

The main difference between DevOps and SRE is that SRE is more operationally driven from the top-down, and it's governed by the developer or development team, instead of the operations team. Specifically, developers have more control over the software monitoring and maintenance processes, two jobs which are normally managed by the operations team. It's as Google's Ben Treynor once remarked, "SRE is what happens when a software engineer is tasked with what used to be called operations."

DevOps aims to bridge the gap between development and operations by culturally aligning their tasks, objectives and initiatives, SRE places the development team at the head of the entire initiative.

While DevOps and SRE sound like they're on opposite sides of the spectrum, both approaches share the same end goals.

- To make incremental changes fast and efficiently

- To reduce the number of organization silos

- To have a flexible, open-minded and adaptable working culture

- Use automation wherever possible

- To monitor performance and improve when necessary

"The general framework of DevOps and SRE are similar in that it shares all the key factors to a successful DevOps/SRE program. DevOps defines these factors as breaking down silos or walls between groups within the organization to encourage more efficient collaboration, blameless failures, automation, monitoring and observability. SRE encompasses all of the DevOps factors but delves deeper into each one to add [or] modify certain criteria to ensure a more detailed recipe to success," shared Hua.

What Are the Benefits and Drawbacks of DevOps?

One of the advantages of DevOps, as noted by Farnsworth, is that this "unification" between development and operations improves collaboration. "Developers have more ability to control deployments and [application running] within their environments and operations [have] a better window into development. [DevOps] can [also] aid in development's workflow and can improve resiliency in the event of failure."

Farnsworth also added that DevOps is "easier to implement and helps to bridge the divides between disciplines without creating an additional team."

However, one of the setbacks to DevOps, according to Hua, is that the DevOps framework does not contain "detailed steps on how to ensure success." Another drawback, also highlighted by Hua, is that "failures are generally accepted" in DevOps and this leads to complacency in addressing "issues quickly."

What Are the Benefits and Drawbacks of SRE?

SRE is well-suited for enterprises and organizations that want to manage large-scale applications. Companies like Netflix, Dropbox and Google, who founded the concept, have

all taken advantage of this approach. The "relentless" pursuit of automation in SRE helps brands eliminate manual work, giving developers more time to innovate and create. Plus, in comparison to DevOps, SRE provides a "good set of detailed steps in each part of the framework to reach a particular goal," explained Hua.

Furthermore, SRE is able to iron out flaws more quickly and efficiently. "If something doesn't work, a good SLI should show it and a change will be made," Farnsworth said.

But despite the advantages, SRE also comes with its own set of drawbacks. For one, Hua said, SLAs "may break down due to teams not honoring their contractual obligation." Since development is executing operation tasks, new SLAs will have to be drawn out to explicitly state roles and responsibilities.

DevOps vs. SRE: Best Use Cases

According to Hua, organizations are able to implement both DevOps and SRE since the two approaches do not conflict with each other. "DevOps generally focuses on the "what" whereas SRE focuses on the "how." Depending on the organization's goals, sometimes the 'what'' would suffice, but other times both the "what" and "how" are needed for a more complete roadmap," said Hua.

And finally, Farnsworth added, "SRE is great for a new rollout with a great deal of control over the entire stack. DevOps is more of a means of taking already embedded talent and organizing into logical connections."

💬 0 Comments

0 Comments   **CMSWire.com**   1 **Login** ▾

♡ Recommend   🐦 Tweet   f Share   Sort by Oldest ▾

Start the discussion…

LOG IN WITH   OR SIGN UP WITH DISQUS ❓

Name

Be the first to comment.

✉ Subscribe   Ⓓ Add Disqus to your siteAdd DisqusAdd   🔒 Disqus' Privacy PolicyPrivacy PolicyPrivacy

Featured Events

Jan 13   NRF New York City 2019

Jan 15   Customer Contact Week Nashville 2019

Jan 16   [CMSWire Webinar] How IT Leaders and Teams Can Win Big in 2019

Jan 21   OPEX Week: Business Transformation World Summit Orlando 2019

Jan 22   [CMSWire Webinar] Modern Intranets & Employee Experience: Trends, Challenges & Solutions

Jan 23   Carnegie Dartlet Conference Orlando 2019

Jan 30   Savage.Z Berlin 2019