

Site Reliability Engineering (SRE)

Idea In Short

This guide will help you better understand Site Reliability Engineering (SRE), which is an advanced practice in software operations that can help increase software efficiency and reduce costs.

Site Reliability Engineering (SRE) is an aspect of software engineering that aims to ensure the **ongoing reliability of software systems**.

This kind of work can help reduce the cost of software operations. It can also increase the performance capabilities of the overall system, which can support future growth.

Site Reliability Engineers (SREs) are production-level engineers focused on software performance once it enters the real world.

Their north star is software reliability.

We want to keep our site up, always. — JC Van Winkle, Site Reliability Engineer at Google Zurich.

Unpacking SRE

The aim of Site Reliability Engineering (SRE) is to increase the reliability of software once it has been put into production, ensuring that it functions optimally and is accessible to users.

Reliability, simply put, is the absence of errors.

Unpacking this: reliability is the ability of a system to function correctly and consistently under various conditions.

Within the context of S.R.E., **reliability refers to the software system's ability to perform as expected**, without any downtime or disruptions.

S.R.E. aims to do the following:

- increase the uptime of software (because software does go down)
- boost the performance of software so that it runs at an optimal speed
- enhance the quality of the code that runs the software and
- fortify the security of software to protect it from intruders

For teams running applications on cloud infrastructure, Google Cloud consulting services can also support SRE goals by helping improve scalability, reliability, monitoring, and operational efficiency across production systems.

In the words of the S.R.E. discipline's originator, Ben Treynor Sloss, Vice President Of Engineering at Google:

Site Reliability Engineering is when you treat [software] operations as an engineering problem

This statement highlights the importance of **treating software operations as a complex and challenging engineering problem** that requires proactive, strategic, and specialized expertise.

Because of its role in software operations, **S.R.E. is directly aligned with the widely-adopted DevOps movement**, which seeks to eliminate barriers between software development and operations.

More than a narrow discipline

To accomplish its goals, Site Reliability Engineering entails fine-tuning the software and its underlying infrastructure. This can mean developing, tailoring, or designing bespoke tooling, as well as advocating for superior work practices.

The biggest misconception about Site Reliability Engineering is that it's only focused on

reliability from a narrow lens i.e. the software is accessible.

But there's much more to it.

Issues like performance, quality of code, and security affect the quality of user experience and the perceived reliability of the software.

Because of this, SRE teams need to address multiple capability areas like observability, performance management, DevSecOps, and more to address all the needs of the software system.

In essence, S.R.E. is not just a reactive approach to fixing problems. It is a comprehensive and proactive approach that seeks to prevent problems before they occur.

Why SRE?

The specialized nature of software developer or software engineering (SWE) roles means some of the work needed to assure reliability is missed.

Let me explain further.

Software engineering (SWE) roles require a high level of specialization and a deep understanding of the technical intricacies involved.

For example, a backend developer may only specialize in Javascript programming as it on its own forms a significant cognitive load.

But this kind of specialization leads to a problem further down the software development lifecycle (SDLC).

Due to the complex and dynamic nature of software development, **it can be easy to overlook measures that ensure the quality of the software in production** This is where Site Reliability Engineering (SRE) comes in.

SREs are responsible for ensuring that the software is reliable, scalable, and efficient. They work closely with developers, operations teams, and other stakeholders to achieve these

goals.

By doing so, SREs play a crucial role in ensuring that software products are delivered on time and meet the high standards demanded by users and customers alike.

Based on this new understanding, you may now feel more confident about bringing SRE into your organization.

But how can you effectively convince others about this approach?

Let's explore this further.

First, let's address one fact that cannot be overlooked.

It takes confidence and conviction to introduce significant changes that may affect the entire team or organization.

You will naturally face resistance or hesitation when introducing change.

But proposing a new function like Site Reliability Engineering (SRE) or restructuring teams toward it can bring significant benefits.

With careful planning, clear communication, and an emphasis on the potential benefits, it can be a successful endeavor.

I will share more specific communication advice with you in a moment.

But let's first consider 3 arguments that you can use to strengthen the logic behind your proposal.

SRE As The Connective Tissue

According to Sebastian Vietz, Director Of Reliability Engineering at Compass Digital:

SRE acts as the connective tissue that brings all aspects of software development together, ensuring that software is reliable, scalable and efficient

Let's unpack the above statement in a few key points:

Site Reliability Engineering (SRE) is a discipline that serves as a foundation for other related fields.

SRE teams design and implement new tools and technologies to improve software system efficiency and effectiveness.

SREs work closely with developers, product managers, and other stakeholders to ensure that systems meet the needs of the organization and its customers.

For instance, SREs can assist AppSec engineers and teams in navigating system complexity to detect and mitigate security threats and vulnerabilities.

Balance Risks And Benefits Of Cloud Computing

Cloud computing has revolutionized business operations, particularly in regulated industries.

However, it presents unique challenges that must be addressed. One such challenge is implementing the same level of IT controls as before, which poses a significant risk when transitioning to cloud-native services.

To balance the need for flexibility with risk reduction, Site Reliability Engineers (SREs) can have a crucial impact on ensuring the smooth functioning of cloud-based systems.

They can do the following:

- educate engineers on best practices
- set up passive guardrails that prevent errors and reduce risks
- provide guidance on the most efficient and effective use of the cloud while staying within established policies.

Additionally, experienced SREs can confidently engage with different stakeholders in the risk management process.

In summary, organizations must balance flexibility and risk when using cloud software. SREs

can help achieve this balance and ensure that cloud-based systems are more resilient and better equipped to handle unexpected events.

Supports Operational Efficiency And Cost Control

Site Reliability Engineering (SRE) will:

- Bring in data and practices
- Increase operational efficiency
- Reduce software operations costs

SRE can display a wide range of data to identify potential bottlenecks and areas of improvement in service performance.

One key strategy is to proactively address incidents, reducing the severity of future incidents and potential financial damage. SRE can also improve DevSecOps practices to prevent costly security mishaps.

By taking a comprehensive and proactive approach to system management, SRE can ensure the long-term success and financial viability of the software.

Tips For Stakeholder Communications

Use Plain Language

Use language that highlights to stakeholders that SRE:

- Solves problems for users, not the system
- Benefits the business, not just the technology

Clearly Sell The Benefits

Here are 5 examples showing **how SREs clearly benefit the business**:

1. Save on cloud computing costs with potential savings up to \$100000+/month on cloud computing costs – make your CFO happy
2. Reduce the potential of active security threats by increasing passive security of

- software through DevSecOps – make your CISO happy
3. Prevent the loss of revenue caused by downtime by increasing the resilience, performance, and overall reliability of software
 4. Avoid outsized headcount growth while still scaling up the growth of software operations – 10X manual operations efforts through automation
 5. Assure fast product launches through continued developer velocity by improving DevOps of developer i.e. shift left

Temporarily Align Your Pitch With The Project Mindset

When working with stakeholders, it's important to consider their project mindset and how it might clash with the continuous improvement approach required for Site Reliability Engineering (SRE) functions.

To avoid confusion, consider introducing a maturity model for the SRE team as a clear path to success with specific milestones and an "end state" to demonstrate progress.

While maturity models may not be the ideal way to achieve long-term results, they offer a straightforward approach for stakeholders to understand the SRE team's goals and how they aim to achieve them.

As the project develops and the SRE team gains new capabilities, it may be possible to revisit the idea of continuous improvement for long-term success.

Position For Budgeting

When presenting a change proposal to executives, prioritize their budget and project mindset.

Pitch your change by communicating the benefits and addressing any potential concerns they may have.

This will gain their buy-in and support, ultimately increasing the likelihood of successful implementation.

Use Case Studies

Executive stakeholders often research other companies in their industry or advanced practitioners in other industries to guide their own strategies and practices.

They can use case studies of respected companies that have implemented SRE successfully to gain valuable insights into the challenges, opportunities, and strategies employed.

These case studies can inspire new ideas and provide valuable guidance for your own company.

To pitch the idea of implementing SRE:

- Create a well-crafted proposal that highlights the positive impact of the change on the organization
- Start with a document and follow it up with a slide deck for a formal presentation
- Emphasize the benefits of the proposed changes in a clear and concise manner
- Keep it simple when communicating with higher-ups, unless asked to elaborate
- Address any potential drawbacks or concerns proactively to counter or alleviate resistance to the proposed changes

Summary

Pitching a new approach to software operations can be a challenging task. It requires presenting a compelling proposal that emphasizes the benefits of Site Reliability Engineering (SRE), using clear language, clearly selling the benefits, and using case studies. I have provided you with three arguments in favor of SRE - it serves as the connective tissue for all other disciplines, balances the risk versus benefit of cloud computing, and supports operational efficiency and cost control. I hope this gives you enough insight to create an effective proposal for SRE in your organization.