Developing a Google SRE Culture

Learner Workbook

About this workbook

Welcome to the beginning of your SRE journey! This workbook contains key points and reflection exercises for each module. The reflection exercises can help you in future conversations with your leadership teams during your journey to SRE adoption.

We recommend that you review each exercise after completing the module video lessons.

Module One

Welcome to Developing a Google SRE Culture

1. Key Points
2. Reflection Activity

1. Key Points

- Customers' experiences with your service tell you how reliable it is.
- In many IT organizations, development and operations teams have conflicting priorities.
- Site Reliability Engineering (SRE) is the practice of balancing the velocity of development features with the risk to reliability.
- SRE can benefit IT teams, regardless of whether they are using cloud or on-premises technology, for both large projects and daily work.

2. Reflection Activity

Have you ever had a concern about your service's reliability? If so, what caused this concern? Were there internal or external factors? How did you address it?

Write down your thoughts below, and keep your experience in mind as you learn about Google's SRE practices.

Module Two

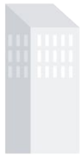
DevOps, SRE, and Why They Exist

1. Key Points
2. Reflection Activity

1. Key Points

- DevOps emerged to help close gaps and break down silos between development and operations teams.
- DevOps is a philosophy, not a development methodology or technology.
- SRE is a practical way to implement DevOps philosophy.
- Developers focus on feature velocity and innovation; operators focus on reliability and consistency.
- SRE consists of both technical and cultural practices.
- SRE practices align to DevOps pillars:

Share ownership



1. Reduce organizational silos

Blamelessness

2. Accept failure as normal

Reduce cost of failure

3. Implement gradual change

Toil automation

4. Leverage tooling and automation

Measure toil and reliability

5. Measure everything

2. Reflection Activity

In this module, you heard the story of an online retailer whose developers suffered from burnout due to the demands of increased feature deployment while addressing reliability issues on the side.

Have you ever noticed this type of behavior with your development teams? If so, what do you think caused it?

Module Three

SLOs with Consequences

1. Glossary
2. Key Points
3. Reflection Activity
4. Postmortem Template

1. Glossary

- **Blameless postmortem:** Detailed documentation of an incident or outage, its root cause, its impact, actions taken to resolve it, and follow-up actions to prevent its recurrence.
- **Reliability:** The number of “good” interactions divided by the number of total interactions. This leaves you with a numerical fraction of real users who experience a service that is available and working.
- **Error budget:** The amount of unreliability you are willing to tolerate.
- **Service level indicator (SLI):** A quantifiable measure of the reliability of your service from your users' perspective.
- **Service level objective (SLO):** Sets the target for an SLI over a period of time.

2. Key Points

- The mission of SRE is to protect, provide for, and progress software and systems with consistent focus on availability, latency, performance, and capacity.
- Understanding SRE practices and norms will help you build a common language to use when speaking with your IT teams and support your organization’s adoption of SRE both in the short and long term.
- Experienced SREs are comfortable with failure.
- Failures are documented in postmortems, which focus on systems and processes versus people.
- 100% reliability is the wrong target because it slows the release of new features, which is what drives your business.

- SLOs and error budgets create shared responsibility and ownership between developers and SREs.
- Fostering psychologically safe environments is necessary for learning and innovation in organizations.
- Organizations developing an SRE culture should focus on creating a unified vision, determining what collaboration looks like, and sharing knowledge among teams.

3. Reflection Activity

1. Think about your IT teams. List some scenarios where working in a psychologically safe environment would benefit them.

2. Do you think blamelessness is achievable in your organization? How can you support and encourage blamelessness and psychological safety within your teams?

Write down as many ideas as you can. Share these with your leadership team when you start your SRE implementation conversations.

4. Postmortem Template

Below is a basic postmortem template. Share this with your IT teams as you start to implement the SRE role and postmortem practice.

Part 1. What happened?

Title:

Date:

Authors:

Status: In Writing/In Review/Reviewed/Published

Summary: --- *What was the incident? Its duration? Its cause?* ---

Impact: --- *Latency? Data loss? Availability?... Include revenue impact if known* ---

Root causes:

Trigger: --- *Action that initiated the incident* ---

Resolution: --- *Actions taken to mitigate or prevent the incident's impact in the short term. Actions taken (fixes deployed) to address the root causes* ---

Detection: ---*How was the incident detected?* ---

Lessons Learned

Some guiding questions:

- *Was the incident detected quickly, or did it take a long time for a human to notice?*
- *Did teams coordinate well among each other, or were there communication problems?*
- *Were the escalation paths clear, or did engineers not know where to go for help?*

What went well?

....

....

....

....

What didn't go so well?

....

....

....

....

Where did we get lucky?

[There is often some aspect of an incident that ensures that it wasn't as bad as it could have been. Often, this aspect wasn't by design. Call this out explicitly so you can build new safeguards and not rely on luck next time.]

....

....

....

....

....

Part 2. What can we do differently next time?

- Work together to document what you've learned from these issues and come up with Action Items.
- *Note: Do not focus solely on bug fixes. Also include procedural changes required to mitigate the impact of similar incidents.*

Owners	Action Items	Priority	Bug/Tickets

Module Four

Make Tomorrow Better than Today

1. Glossary
2. Key Points
3. Reflection Activity

1. Glossary

- **Continuous integration:** Building, integrating, and testing code within the development environment.
- **Continuous delivery:** Deploying to production frequently, or at the rate the business chooses.
- **Canarying:** Deploying a change in service to a group of users who don't know they are receiving the change, evaluating the impact to that group, and then deciding how to proceed.
- **Toil:** Work directly tied to a service that is manual, repetitive, automatable, tactical, or without enduring value, or that scales linearly as the service grows.

2. Key Points

- Change is best when small and frequent.
- Design thinking methodology has five phases: empathize, define, ideate, prototype, and test.
- Prototyping culture encourages teams to try more ideas, leading to an increase in faster failures and more successes.
- Excessive toil is toxic to the SRE role.
- By eliminating toil, SREs can focus the majority of their time on work that will either reduce future toil or add service features.
- Resistance to change is usually a fear of loss.
- Present change as an opportunity, not a threat.
- People react to change in many ways, and IT leaders need to understand how to communicate with and support each group.

3. Reflection Activity

1. Think about work your IT teams do that could be considered toil. How much of that toil is bad? How much is good? Write down your thoughts about the type of toil that you would consider automating, and the toil that you would consider keeping.

2. How might you present adoption of SRE culture and practices as an opportunity to your IT teams and other leadership? Brainstorm some ideas below.

Module Five

Regulate Workload

1. Glossary
2. Key Points
3. Reflection Activity

1. Glossary

- **Affinity bias:** Tendency to gravitate toward those who are similar to you, such as with race, gender, socioeconomic background, or education level.
- **Confirmation bias:** Tendency to find information, input, or data that supports your preconceived notions.
- **Selective attention bias:** Tendency to pay attention to things, ideas, and input from people whom you tend to gravitate toward.
- **Labeling bias:** Tendency to form opinions based on how people look, dress, or appear externally.

2. Key Points

- Measure reliability with good service level indicators (SLIs).
- A good SLI correlates with user experience with your service; that is, a good SLI tells you when users are happy or unhappy.
- Measure toil by identifying it, selecting an appropriate unit of measure, and tracking the measurements continuously.
- Monitoring allows you to gain visibility into a system, which is a core requirement for judging service health and diagnosing your service when things go wrong.
- Goal-setting, transparency, and data-driven decision making are key components of SRE measurement culture.
- To make truly data-driven decisions, you need to remove any unconscious biases.

3. Reflection Activity

1. Think about how your IT teams work. What are some things you know they are already measuring? What are some things you think they should measure that they don't already measure?

....

....

....

....

....

....

....

....

....

2. How do you currently set and measure goals in your organization? Is there anything you think you could improve about the process?

....

....

....

....

....

....

....

....

Module Six

Apply SRE in Your Organization

1. Key Points
2. Reflection Activity

1. Key Points

- **Kitchen Sink/”Everything SRE” team:** We recommend this approach for organizations that have few applications and user journeys and where the scope is small enough that only one team is necessary, but a dedicated SRE team is needed in order to implement its practices.
- **Infrastructure team:** This type of team focuses on maintaining shared services and components related to infrastructure, versus an SRE team dedicated to working on services related to products, like customer-facing code.
- **Tools team:** This type of SRE team tends to focus on building software to help their developer counterparts measure, maintain, and improve system reliability or other aspects of SRE work, such as capacity planning.
- **Product/Application team:** This type of SRE team works to improve the reliability of a critical application or business area. We recommend this implementation for organizations that already have a Kitchen Sink, Infrastructure, or Tools-focused SRE team and have a key user-facing application with high reliability needs.
- **Embedded team:** This team has SREs embedded with their developer counterparts, usually one per developer team in scope. The work relationship between the embedded SREs and developers tends to be project- or time-bounded and usually very hands-on, where they perform work like changing code and configuration of the services in scope.
- **Consulting team:** This implementation is very similar to the embedded implementation, except SRE are usually less hands-on. We recommend staffing one or two part-time consultants before you staff your first SRE team.

- Organizations with high SRE maturity have well-documented and user-centric SLOs, error budgets, blameless postmortem culture, and a low tolerance for toil.
- Engineers with operations experience and systems administrators with scripting experience are good first SREs to hire.
- Upskill current team members with necessary SRE skills such as operations and software engineering, monitoring systems, production automation, system architecture, troubleshooting, culture of trust, and incident management.
- Contact your Account Executive or Account Director to learn how the Google Cloud Professional Services team can support your organization’s adoption of SRE.

2. Reflection Activity

1. What do you think is your organization's maturity level for adopting SRE? Where does it fit into the SRE journey? Write down your ideas.

	1	2	3
Technical	Blameless postmortems SLOs Error budgets	CI/CD Canarying Toil automation	Reliability measurement Toil measurement Monitoring
Cultural	Psychological safety Share vision and knowledge Foster collaboration	Design thinking Prototyping Change management	Goal-setting Transparency Data-driven decision making

2. Think about your IT team composition. Are there already employees with the skillset for SRE? How might you quickly upskill and train these employees to move into the SRE role?

Resources

- Site Reliability Engineering

  Members of the SRE team explain how their engagement with the entire software lifecycle has enabled Google to build, deploy, monitor, and maintain some of the largest software systems in the world.

- The Site Reliability Workbook

  The Site Reliability Workbook is the hands-on companion to the bestselling Site Reliability Engineering book and uses concrete examples to show how to put SRE principles and practices to work. This book contains practical examples from Google's experiences and case studies from Google's Cloud Platform customers. Evernote, The Home Depot, The New York Times, and other companies outline hard-won experiences of what worked for them and what didn't.

- Google Cloud Consulting Services

  When you choose a Google Cloud consultant, you'll be working hand in hand with experts who will educate your team on best practices and guiding principles for a successful implementation. Our deep technical expertise and services help you unlock business value from the cloud across a range of solutions—including infrastructure, application modernization, data management and analytics, machine learning, and security.

- Site Reliability Engineering: Measuring and Managing Reliability (Coursera)

  This course teaches the theory of service level objectives (SLOs), a principled way of describing and measuring the desired reliability of a service. Upon completion, learners should be able to apply these principles to develop the first SLOs for services they are familiar with in their own organizations.
  Learners will also learn how to use service level indicators (SLIs) to quantify reliability and error budgets to drive business decisions around engineering for greater reliability. The learner will understand the components of a meaningful SLI and walk through the process of developing SLIs and SLOs for an example service.

- DORA DevOps Quick Check

  Measure your team's software delivery performance and compare it to the rest of the industry by responding to five multiple-choice questions. The quick check takes less than a minute to complete, and we don't store your answers or personal information. Immediately compare your team's performance to others.