



# SRE 101

## Introduction to Site Reliability Engineering

Seshagiri Sriram





# How to DevOps?

\*Generally

[via DevOps Principles](#)

- Have CI/CD practices
- Shift Left
- Continuous Improvements
- Remove Silos
- Automate
- Shared Responsibilities
- Autonomous Teams



# SRE

## Guiding Principles

- **You can't improve what you can't measure**
  - SLI, SLO, Error Budget
- Embracing Risk
- Eliminate **Toil**
- Implementation agnostic monitoring
- Automate
- Simplicity\*





# Agile Manifesto

Scrum, Kanban, Lean, XP

# DevOps

SRE, Systems Engineer,  
Platform Engineer, Automation  
Engineer, Cloud x Engineer





# SRE vs DevOps

- Non Competing
- Class SRE Implements Interface DevOps
  - <https://goo.gl/CKv3tV>
- SRE is part of whole DevOps Umbrella
  - SRE defines the practices which DevOps suggests
  - And MORE





# SRE Details

## What is SRE toil?

Tasks with the following descriptions



# SRE Details



## SLI

Service Level Indicator

Availability, Throughput, Error Rate

## SLO

Service Level Objectives

E.g.: 99% availability



## Error Budget

the amount of error that your service can accumulate over a certain period of time.

**Tolerance of user happiness**





## SRE Details

**SLA**



SERVICE LEVEL AGREEMENT

the agreement you make  
with your clients or users

**SLOs**



SERVICE LEVEL OBJECTIVES

the objectives your team must  
hit to meet that agreement

**SLIs**



SERVICE LEVEL INDICATORS

the real numbers on  
your performance





## SRE Details

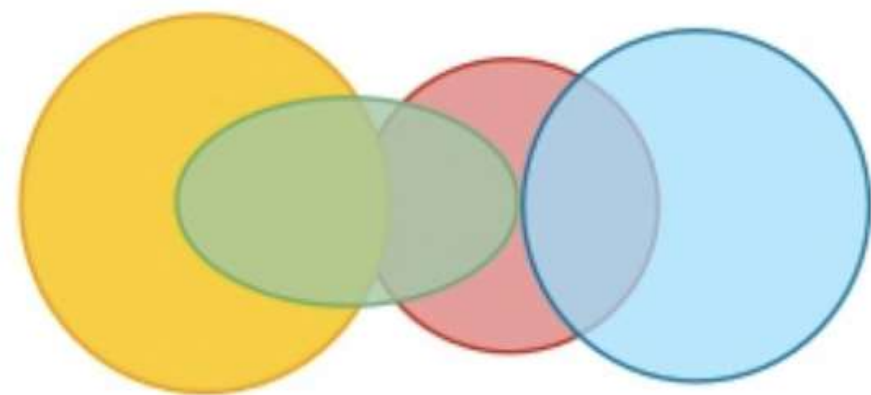


# POSTMORTEM



# SRE Practices

- Remove Toil
- Defining criticalities (System, downtime, unavailability)
- System Designing (DR, Multi or Poly Cloud, Multi-Region Deployments)
- Observability
- Chaos Engineering
- Incident Management
- Measure and Control MTTR and MTBF
- Post Mortems
- Access Controls to Environments



● Dev   ● DevOps   ● SRE   ● Ops

<https://web.devopstopologies.com/>

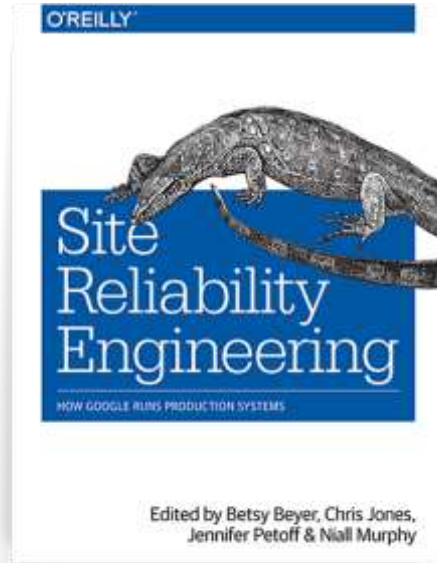
This pattern is used by many organizations that have a high degree of organizational & engineering maturity.

- SRE often act as gate-keepers for production readiness
- Sub-standard software is rejected and SREs provide support in reformatting potential operational issues



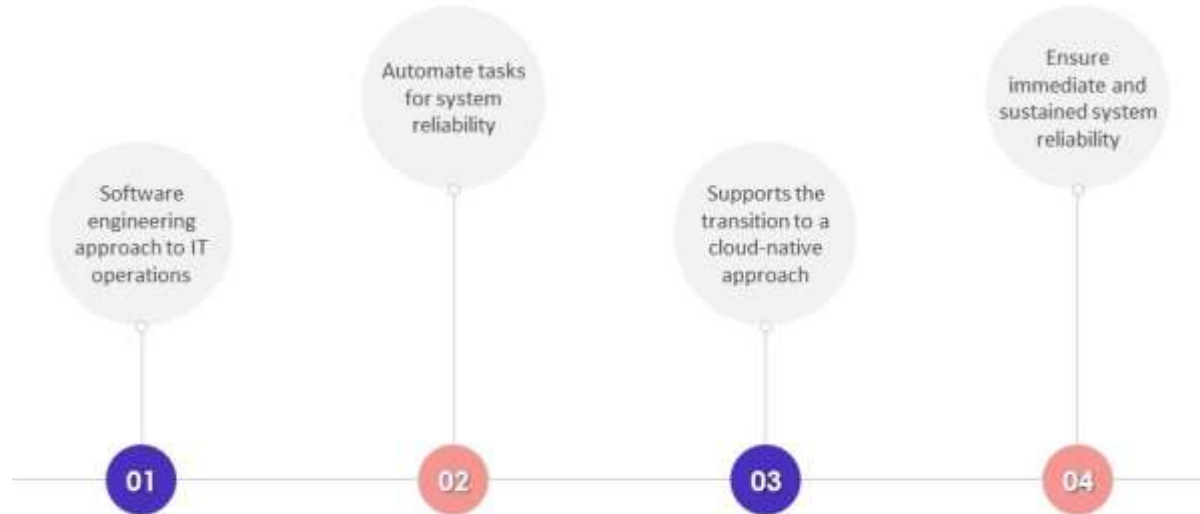
*Mikey Dickerson's Hierarchy of Service Reliability*

# The SRE Handbook



<https://sre.google/sre-book/introduction/>

## Introduction to site reliability engineering approach



## Importance of site reliability engineering approach




**Customer Experience**

- Prevents errors, prioritizing new features
- Automated development reduces errors



**Operations Planning**

- Plans for failures, minimizing downtime impact
- Better cost estimation



**Collaboration**

- Enhances collaboration for swift updates



# Primary principles of site reliability engineering



## Change implementation

- Embrace small, frequent changes for sustained reliability
- Automate for consistency and efficiency



## Reliability automation

- Embed reliability principles in the pipeline
- Automate for early issue detection and resilient architecture

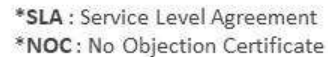
These are some of the core principles.

In later slides, we will see how this impacts

**Stability**

**Reliability**

**Engineering**



# Best practices for site reliability engineering



## Holistic Analysis

SRE promotes comprehensive problem-solving



## User-Centric SLOs

Define SLOs from an end-user perspective for optimized service



## Skill Diversification

Training transforms teams into expert SREs



## Automated Efficiency

Upfront investment saves time, reduces redundancy

# Common site reliability engineering tools



# Introduction to observability in site reliability engineering



Understanding a system's internal state and behavior



Based on its external outputs and interactions



**Main components**

**Metrics**

Measure health of system

**Logs**

Record system events

**Traces**

Records flow of requests

# Observability metrics in site reliability engineering



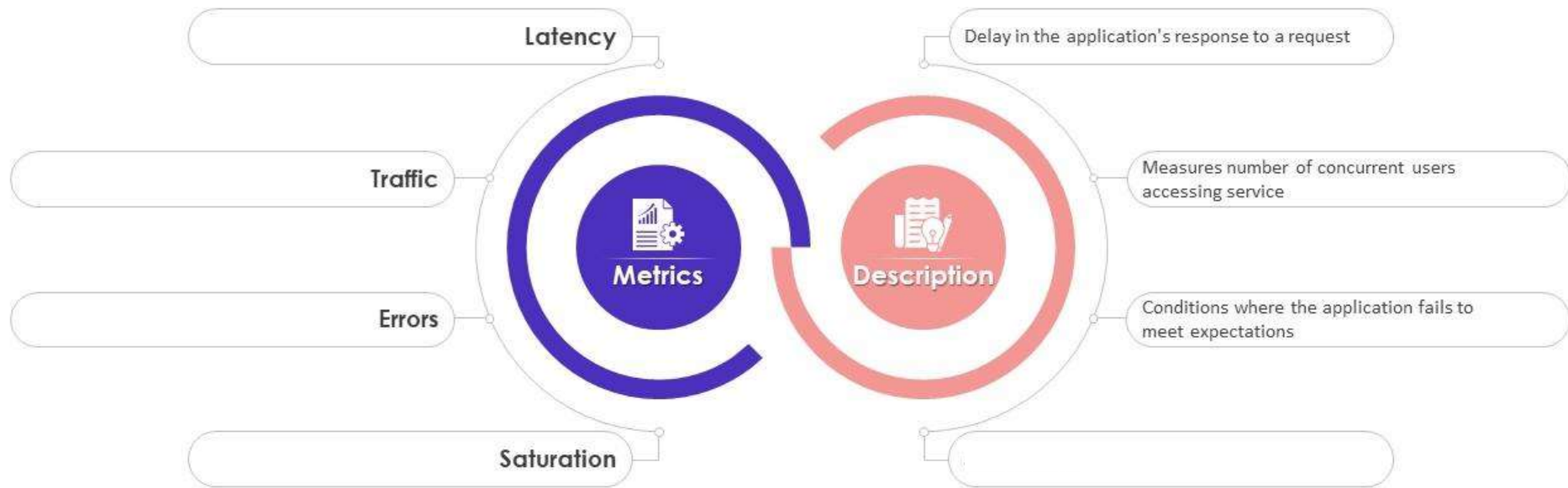
\*SLOs : Service Level Objectives

# Main aspects of monitoring in site reliability





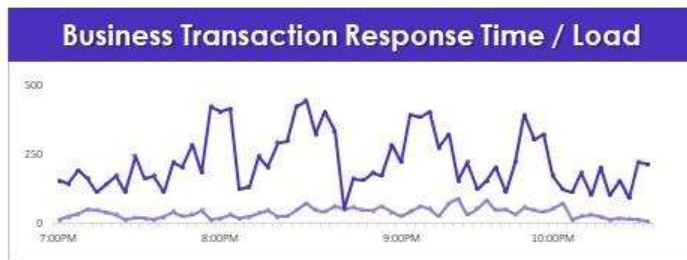
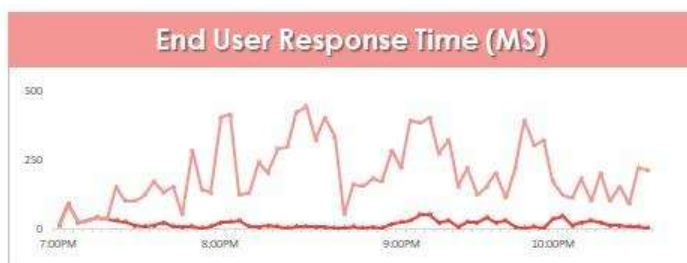
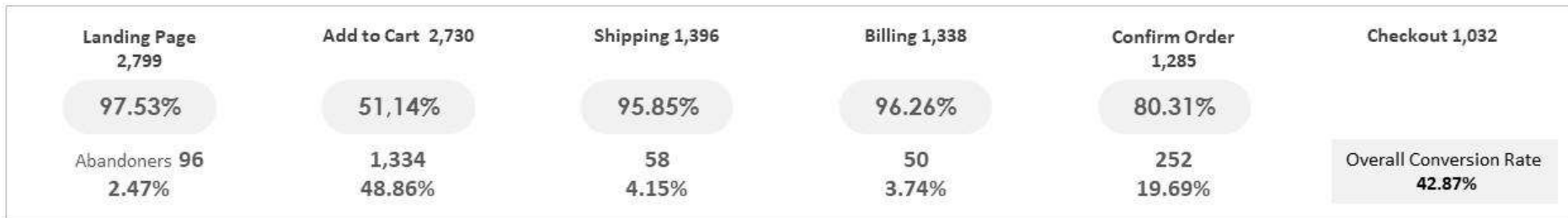
# Monitoring metrics in site reliability engineering



# Introduction to reliability as a service for EVs



# Site reliability engineering dashboard for web application



\*SLOs : Service Level Objectives

# Service level indicators in site reliability engineering



Metric for system or service performance



Aids data-driven decision-making



Identifies areas for improvement



Facilitates preventive measures

# Service level agreement in site reliability engineering



Contractual agreement between  
business and its customer



Documents and outlines  
service levels



Specifies metrics for  
customer satisfaction

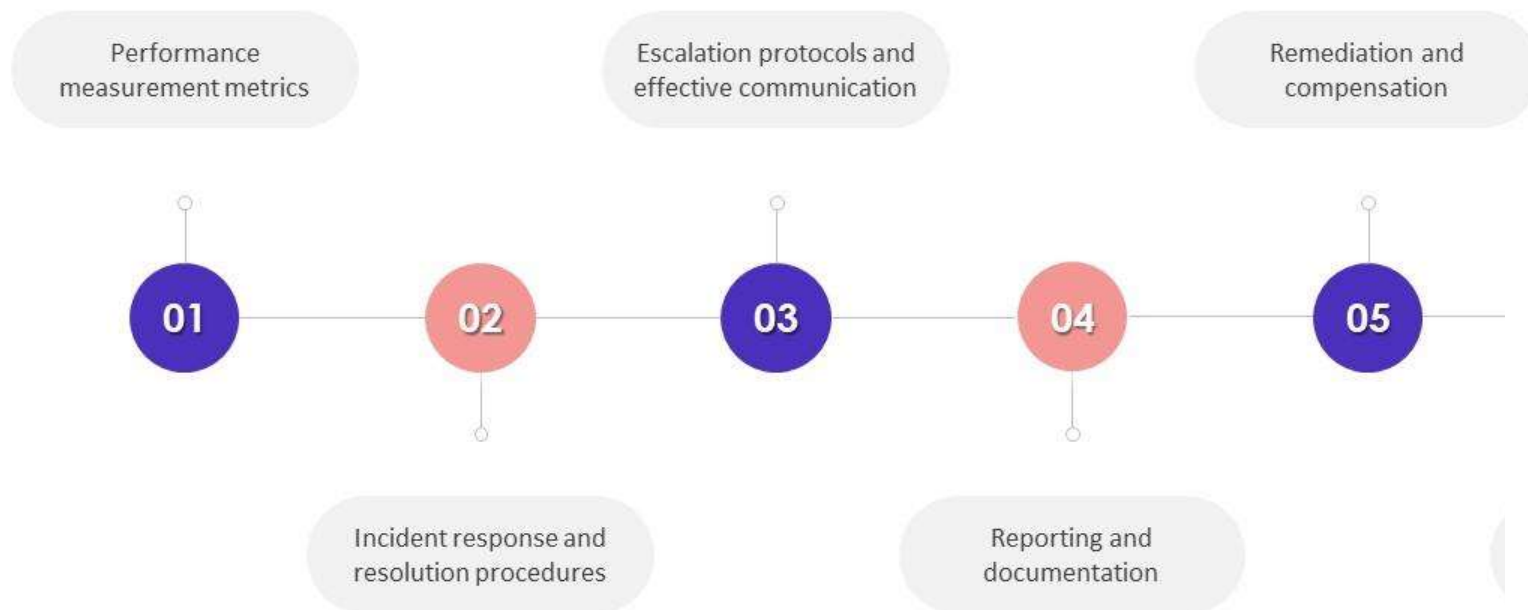


Financial penalties



Contract termination

# Distinct components of typical SLA in SRE



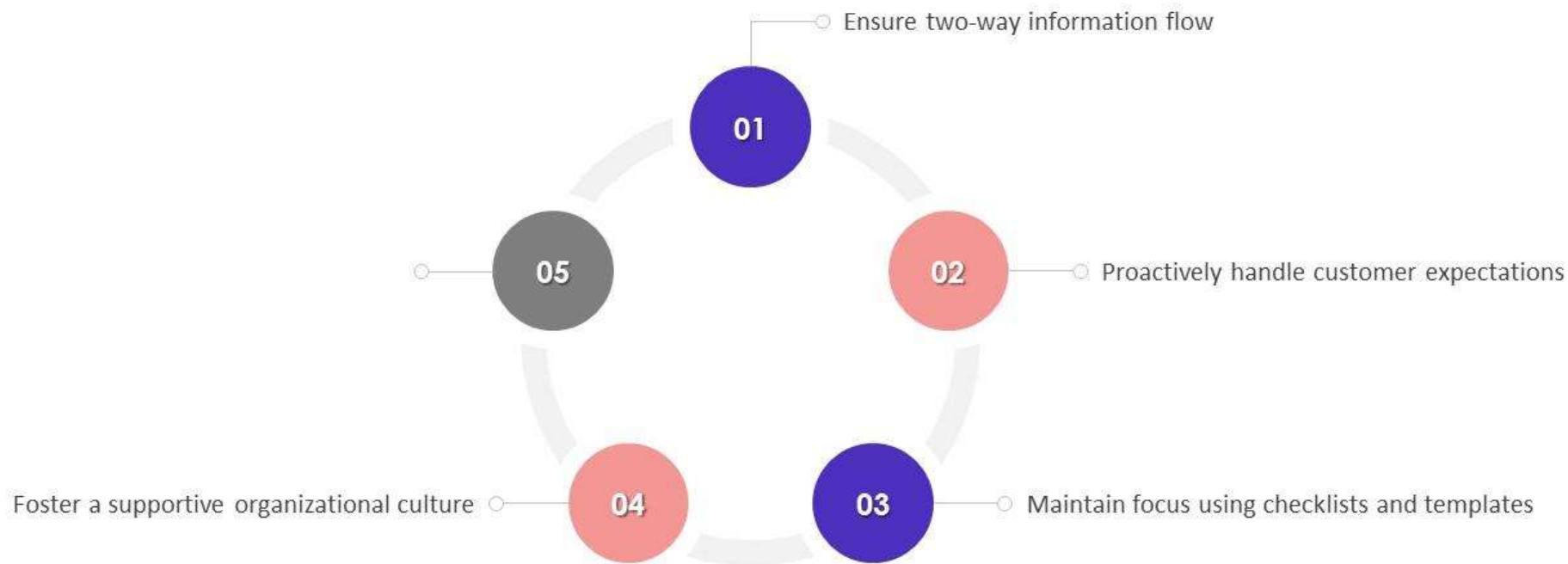
\*SLA : Service Level Availability

# Major design principles for reliability in cloud

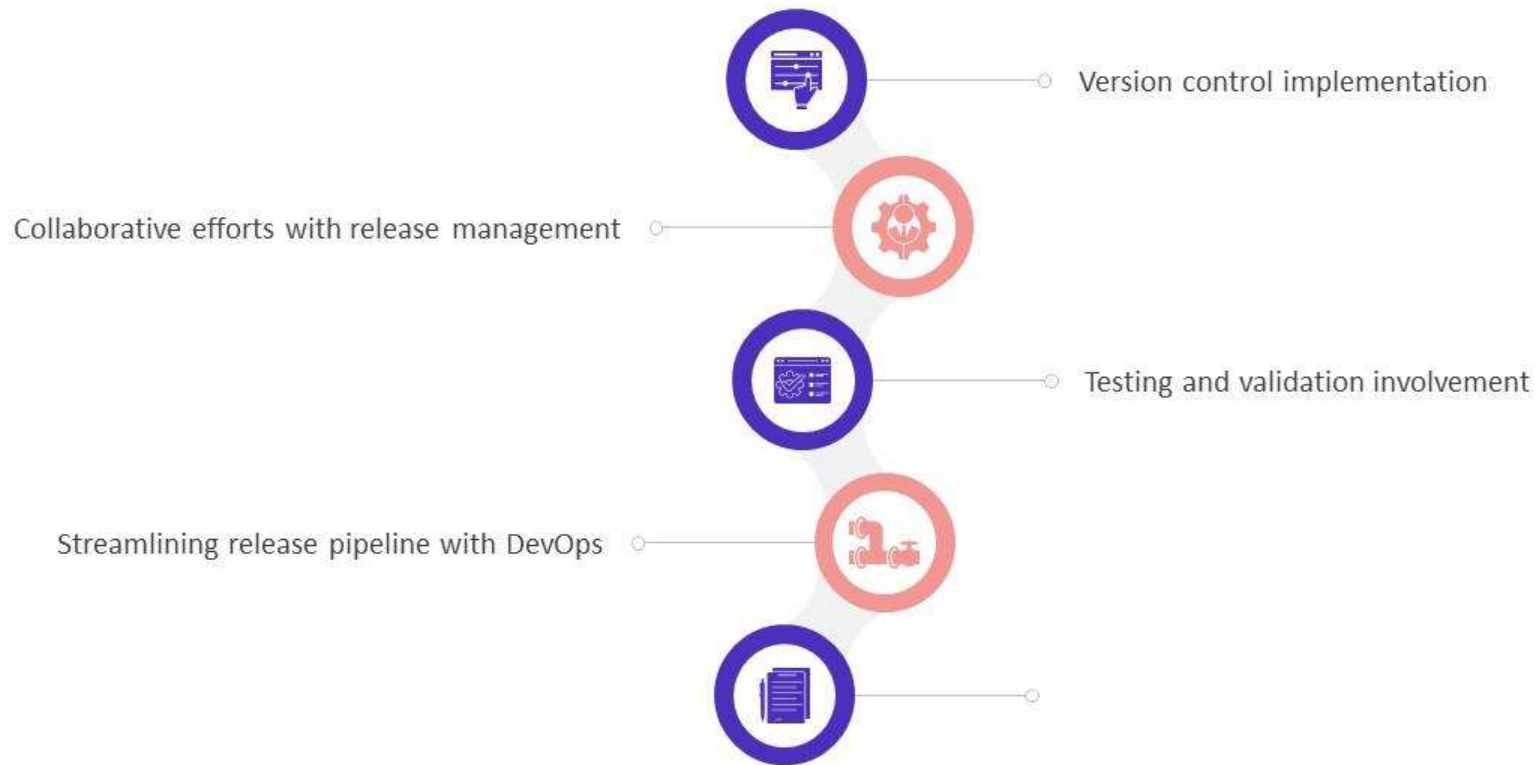




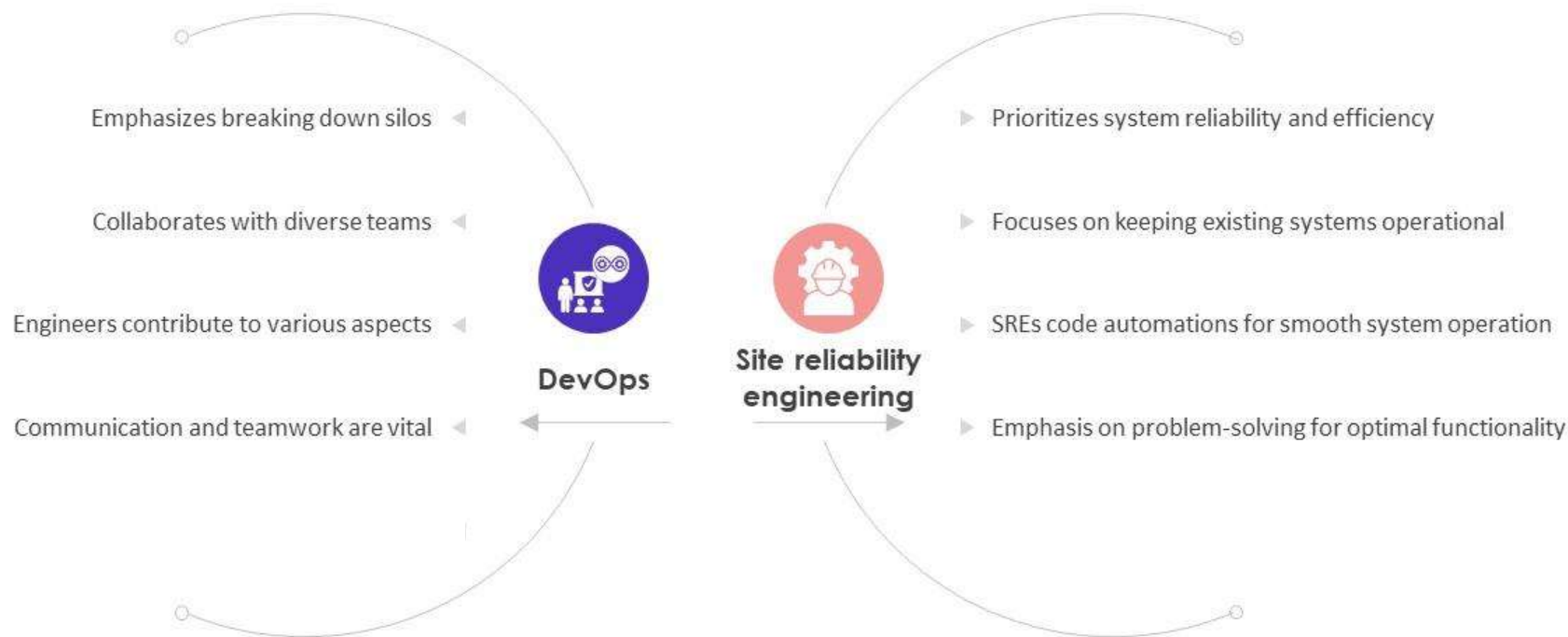
# Effective strategies to achieve reliable service



# Release engineering best practices in product management



# Comparison between DevOps and Site reliability engineering



# Site reliability engineering checklist to track progress



Tasks	Description	Status
Monitoring	Set up alerts for critical systems	
Capacity planning	Track and strategize for future capacity needs	
Automation	Employ automation tools to streamline tasks	
Documentation	Maintain detailed system documentation	

# Training plan for site reliability engineering

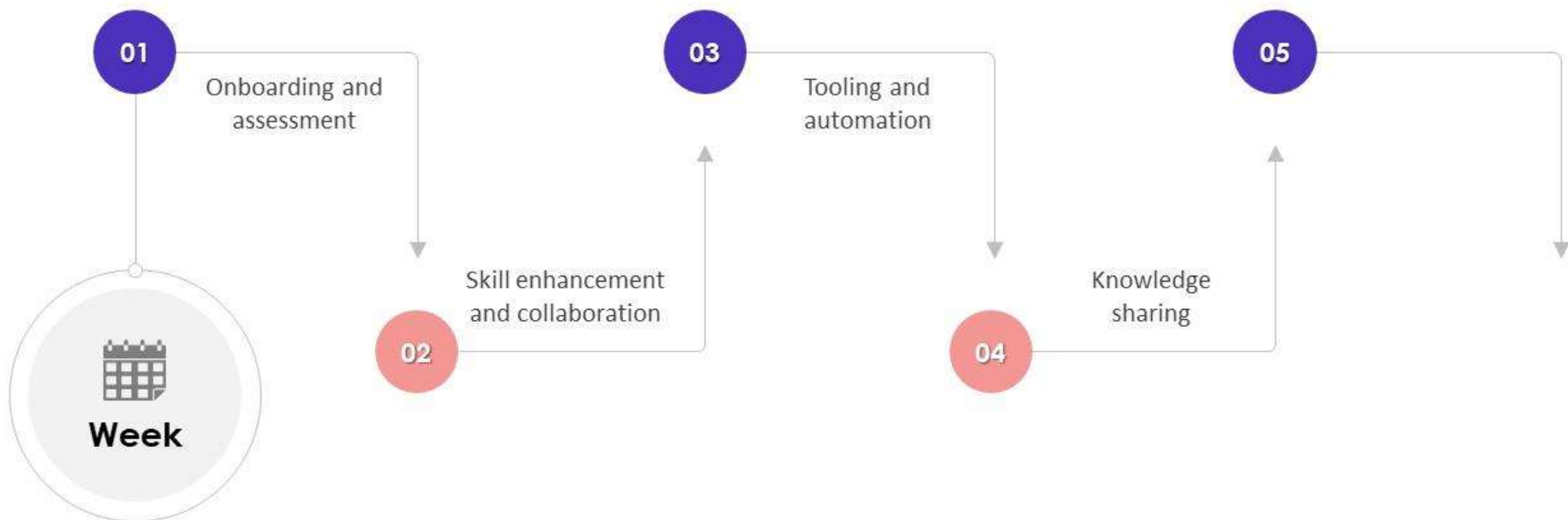
 Module	 Duration	 Mode of Training	 Key Speaker
01 Introduction to SRE	1 week	Instructor-led	SRE Lead
02 Linux Fundamentals	2 weeks	Hands-on workshop	Linux Expert
03 Monitoring and Alerting	2 weeks	Interactive sessions	Monitoring Specialist



# SRE Practices

- Other Plans
  - Implementation Budgets
  - Training plans
  - Schedules
  - Risk Management
  - Costing

# Site reliability engineering implementation timeline

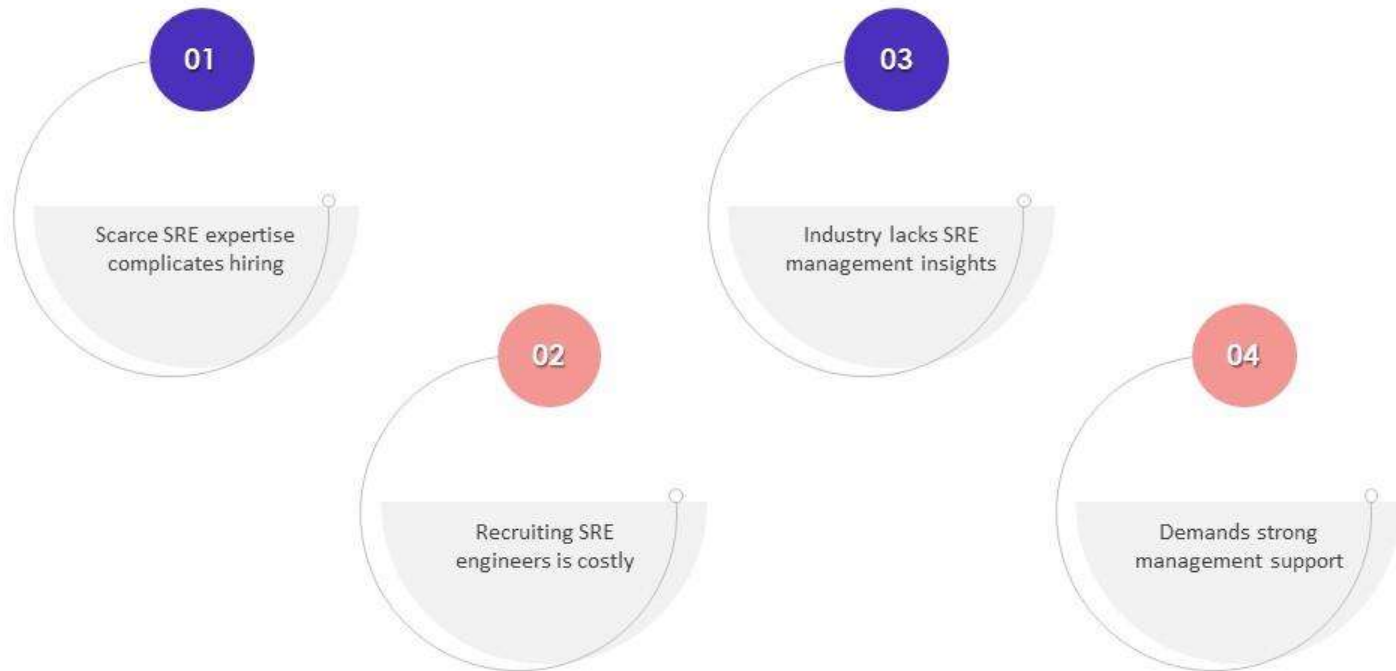




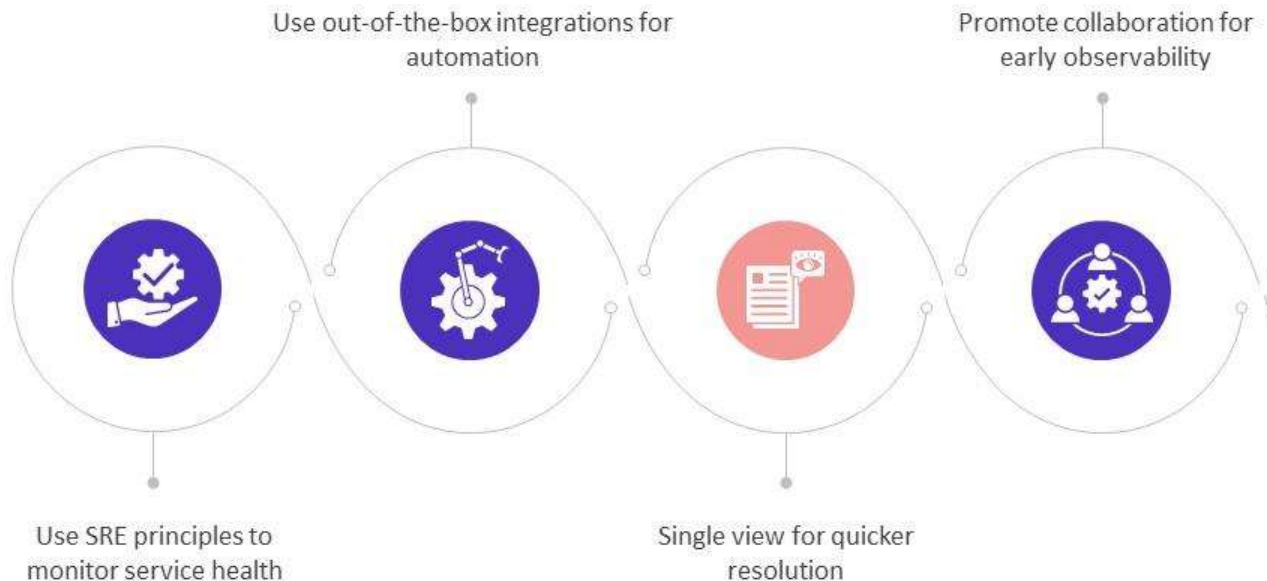
# Site reliability engineering before vs after impact



# Major limitations of site reliability engineering



# Features of Site reliability tools and resources





# SRE Practices

- Timelines
- Financials
- Targets



**CRITICAL TO BE DEFINED AND MEASURED**

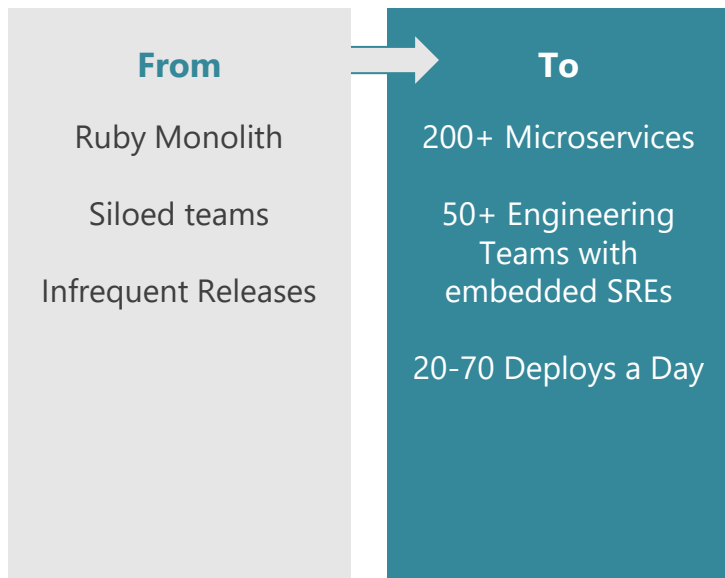


# Defining the Principles, Habits, and Practices of Site Reliability Engineering

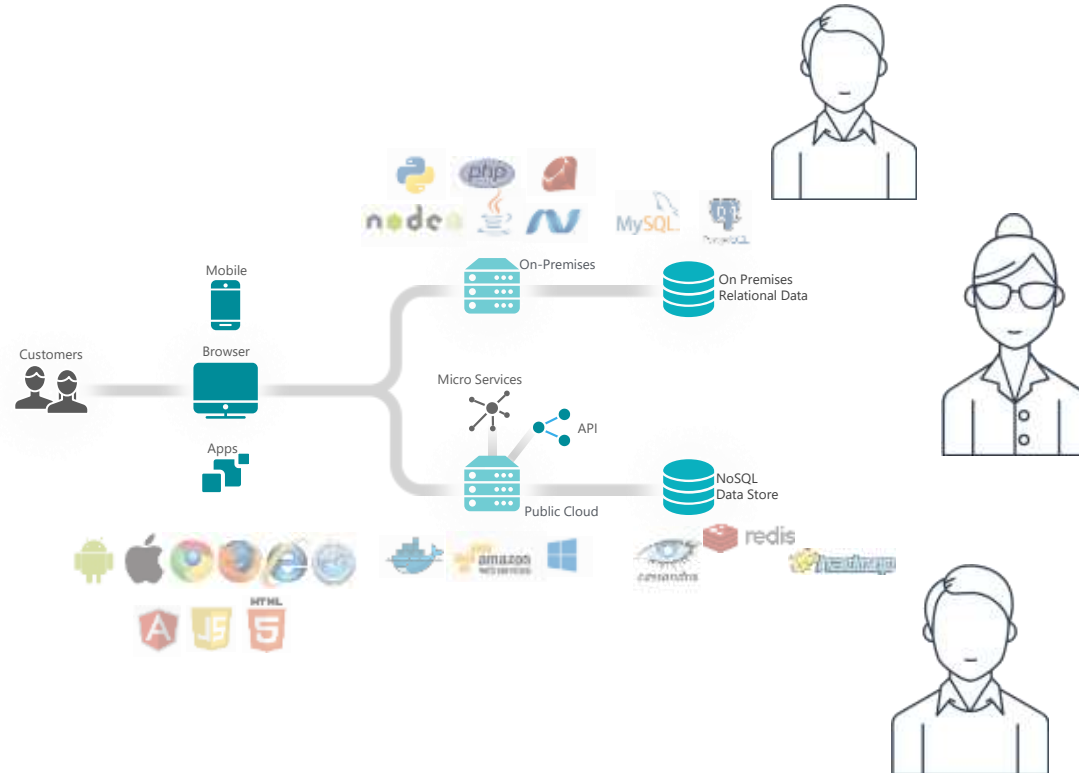
Tori Wieldt, Developer Advocate, 08.06.2018



# A Little Background about New Relic



# How it was



# We Asked Our Stakeholders

Why do we have SREs at New Relic?



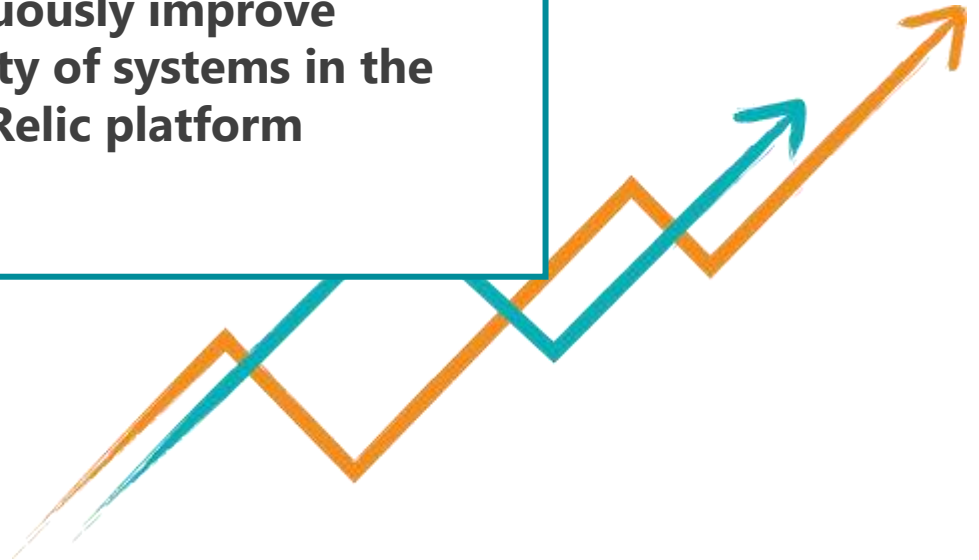
What's the vision for our SRE team?

How can SREs most effectively contribute to the future of our platform?



# One Goal

**Continuously improve  
the reliability of systems in the  
New Relic platform**



# Two Roles



## "Pure" SRE

**Build and support our core internal platform:**

Container Fabric  
Networking Systems



## Embedded SRE

**Partner with Eng Teams**

**Domain Experts in:**

Reliability  
Tooling  
Scaling

# Three Spheres



**Stability**



**Reliability**




**Engineering**

# What SREs Do

- ✓ Champion reliability best practices.
- ✓ Guide designs and processes with an eye toward resilience and low toil.
- ✓ Reduce technical complexity and sprawl.
- ✓ Drive the usage of tooling and common components.
- ✓ Implement software and tooling to improve resilience and automate operations.

# SRE Tasks

 <p>Work with teams to adopt operational best practices</p>	<ul style="list-style-type: none"><li>• Work with teams to update their risk matrices; audit for missing or outdated runbooks; influence teams to prioritize the most important reliability work.</li><li>• Work with teams to hold “game days” to test the resilience of their systems against injected fault conditions.</li></ul>
<p>Stay current on our pipeline and build process, and know the top risks for their team(s)</p>	<ul style="list-style-type: none"><li>• Meet with architects and SREs on other teams to discuss concerns and changes.</li><li>• Use state-of-production knowledge to guide team risk matrices, operational processes, and priorities.</li></ul>

# More SRE Tasks

Building, or helping teams adopt, core shared internal platform components



- Work with teams to migrate systems into a new version of our shared deployment pipeline.
- Contribute code or tools to our container runtime platform.
- Limit technical sprawl by guiding teams to select appropriate existing tools rather than building new ones.

Improve the monitoring and observability of the New Relic platform

- Work with teams to clean up noisy unused alerts and ensure that important problems are alerted on.
- Build an integration to our software to create new visibility into our platform.

# Even More SRE Tasks

Implement automation, tooling, and application code to improve reliability and reduce **toil**.



- Identify a commonly used manual runbook and automate it with software.
- Identify a common failure pattern for new deployments and implement a system to automatically detect and roll back that type of failed deploy.
- Work with teams on the design of new services to ensure those services will be scalable and robust.
- Update an application's DB connection pool to use a more reliable library.

Mentor less senior SREs and grow the SRE community and practice at our company

- Have a meeting, or lunch, once a week with a less senior SRE to discuss work challenges and solutions.
- Pair with other SREs experiencing problems you've previously encountered or solved.
- Document and share novel solutions and other effective strategies.

# And Lastly

Perform task-based operational work (toil)

- Unblock teams with operational needs where automated or self-service solutions do not yet exist
- Track down hardware defects on servers.
- Provision new network endpoints.
- Run Ansible playbooks.



# Keys to SRE Success

**Reliability** is a feature



Query your stakeholders

Reliability depends on **shared understanding**



Develop clear, specific guidelines

SRE is a **challenging, cross-disciplinary** practice



Build a strong SRE community

1

## **Determine Your Goal**

Example:

Continuously improve  
the reliability of the  
systems of our  
company's platform.

2

## **Establish Roles**

Examples:

Pure SRE  
Embedded SRE

3

## **Focus Areas**

Examples:

**Stability**  
Reliability  
Engineering