SITE RELIABILITY ENGINEERING

(SRE)

Site Reliability Engineering (SRE) is a discipline that applies software engineering principles to IT operations in order to build and maintain highly reliable, scalable, and efficient systems. Originating at Google, SRE focuses on automating operational tasks, monitoring system performance, and proactively managing failures to minimize downtime. By using metrics such as Service Level Objectives (SLOs) and Service Level Agreements (SLAs), SRE teams balance system reliability with rapid development. Overall, SRE helps organizations deliver consistent, high-quality services while improving system stability and operational efficiency.

WHY IS SRE

1. SRE is used to improve system reliability and ensure high availability by continuously monitoring applications and preventing failures before they affect users.

2. It automates repetitive operational tasks such as deployments, scaling, and recovery, which reduces manual effort and minimizes human errors.

3. SRE helps balance fast software development with system stability by controlling releases using reliability metrics and error budgets.

4. It minimizes downtime by enabling fast incident detection, structured incident response, and efficient system recovery processes.

5. SRE allows systems to scale smoothly as user demand grows while maintaining consistent performance and efficient resource usage.

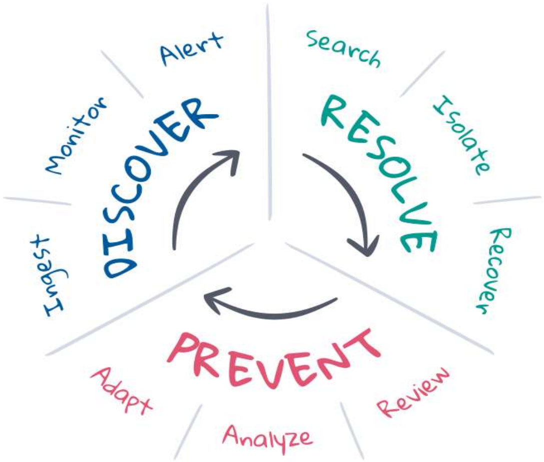

FIG:1 FUNCTIONS OF SRE

WHAT IS SRE

SRE (Site Reliability Engineering) is a practice that uses software engineering and automation to keep applications reliable, available, and scalable. It focuses on monitoring systems, automating operations, and quickly handling failures to reduce downtime. The concept was introduced by Google to manage large-scale systems efficiently.

- SRE defines and tracks **SLIs, SLOs, and SLAs** to measure and enforce service reliability.
- It uses **automation and Infrastructure as Code (IaC)** to manage deployments, scaling, and recovery.
- **Error budgets** are applied to balance system reliability with continuous feature releases.
- SRE implements **monitoring, alerting, and observability** to detect failures in real time.
- Post-incident analysis (**blameless postmortems**) is conducted to prevent recurring failures.

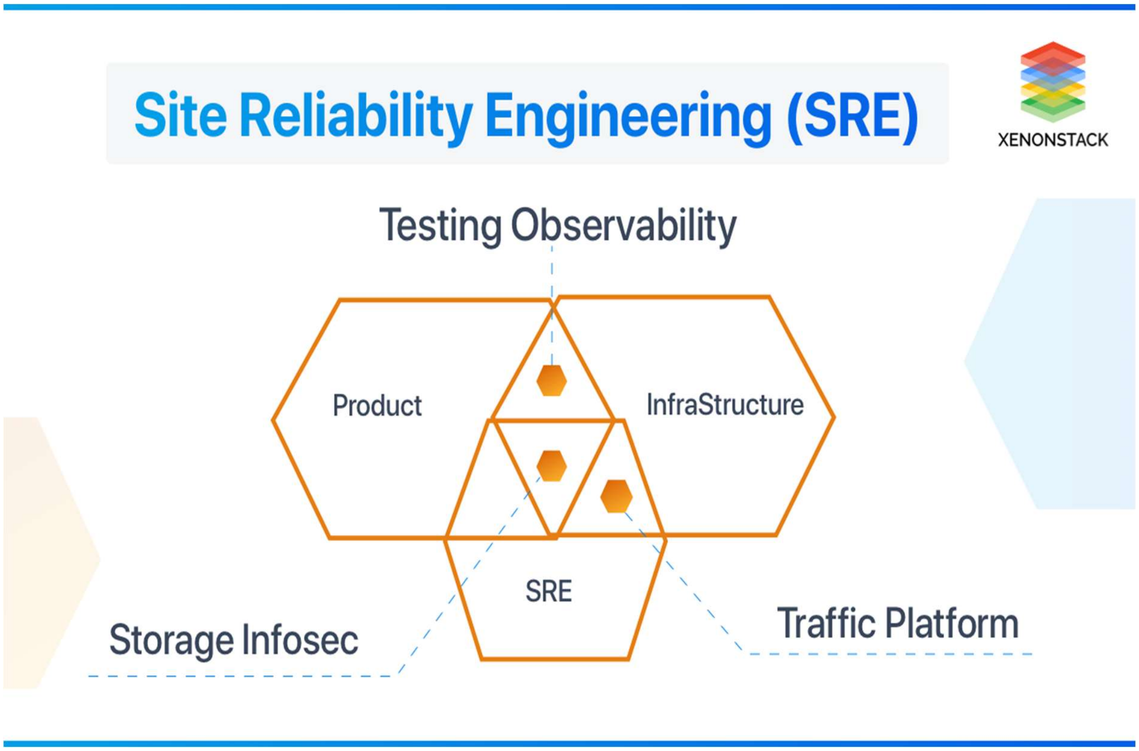

FIG:2 USE OF SRE

USE OF SRE

- Define reliability targets by setting and tracking Service Level Indicators (SLIs), Service Level Objectives (SLOs), and Service Level Agreements (SLAs) to measure system health.
- Automate infrastructure and operations using scripts, CI/CD pipelines, and Infrastructure as Code (IaC) to reduce manual work and configuration errors.
- Monitor and observe systems through metrics, logs, and traces to detect performance issues, latency, and failures in real time.
- Handle incidents and outages by responding to alerts, coordinating recovery, and restoring services quickly to minimize user impact.
- Implement error budgets to balance reliability with rapid feature releases, ensuring controlled and safe deployments.
- Improve system scalability through load testing, capacity planning, and performance optimization as traffic grows.
- Conduct blameless postmortems after failures to identify root causes and prevent similar issues in the future.
- Optimize costs and resources by efficient utilization of servers, cloud services, and scaling strategies.

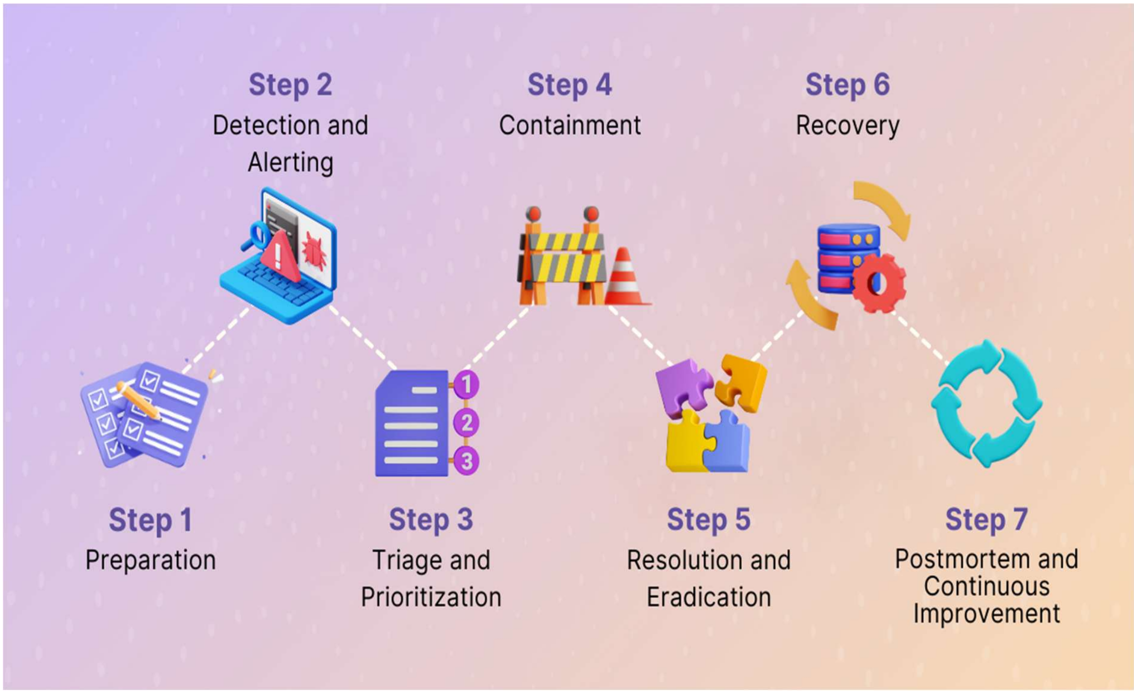

FIG :3 MAIN USE OF SRE

DEVOPS VS SRE

DevOps	SRE (Site Reliability Engineering)
A cultural and technical approach to improve collaboration between development and operations teams	An engineering discipline that applies software engineering to operations
Faster and continuous software delivery	High reliability, availability, and system stability
CI/CD, automation, collaboration	Reliability, monitoring, incident management
Practices, tools, and mindset	Engineering-driven and metrics-based
Deployment frequency, lead time, MTTR	SLIs, SLOs, SLAs, error budgets
Fix issues after deployment	Prevent failures and manage risk before release
Automates build, test, and deployment	Automates operations, recovery, and reliability
Industry-wide movement	Introduced by Google

PRINCIPLE OF SRE

1. Reliability is treated as an important feature of the system.
2. Failures are allowed within limits using error budgets.
3. Manual work is reduced by automating repeated tasks.
4. Systems are continuously monitored to detect problems early.
5. Systems are designed to handle failures and recover quickly.
6. Issues are reviewed without blame to learn and improve.

.

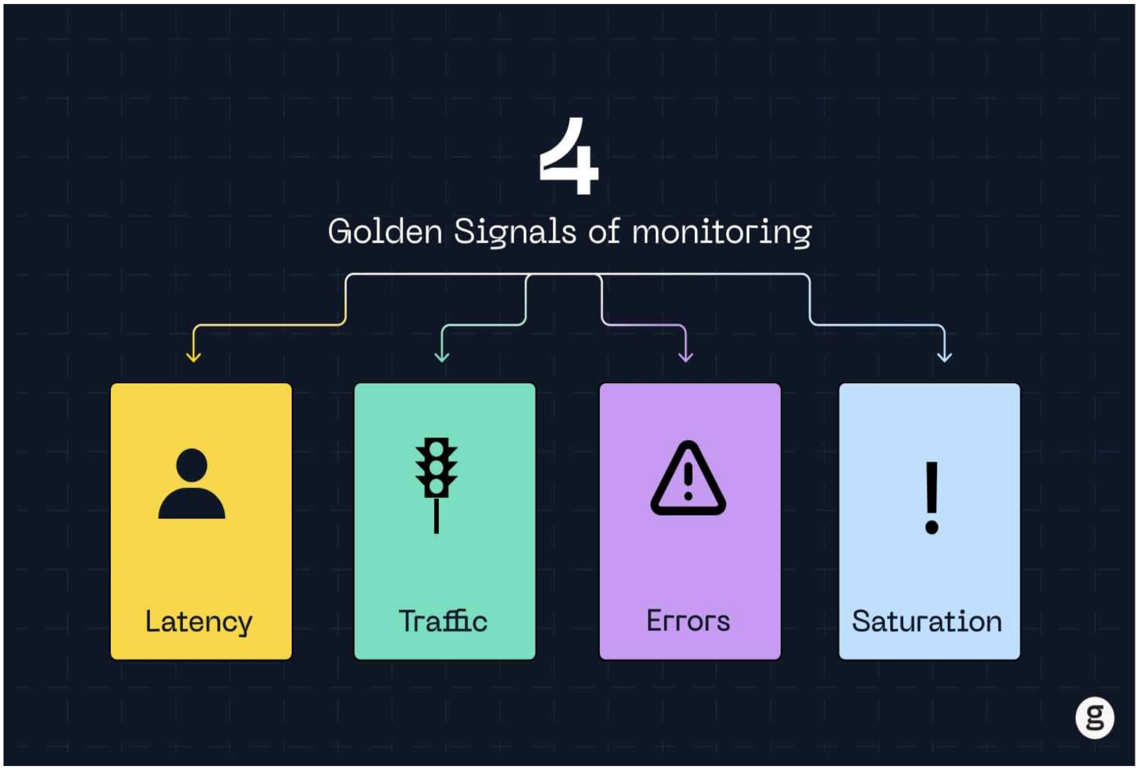

FIG:4 GOLDEN RULES OF SRE