

Integrating Site Reliability Engineering SRE for Effective Product Development: A Focus on SLAs, SLIs, and SLOs

Akshay Prabhu

Abstract: In today's fast-paced IT landscape, ensuring reliability and performance is paramount. Establishing a Site Reliability Engineering (SRE) mindset early in product development is crucial for aligning operational excellence with business objectives. This journal explores the significance of embedding SRE principles from the outset, focusing on defining Service Level Agreements (SLAs), Service Level Indicators (SLIs), Service Level Objectives (SLOs), and Non-Functional Requirements (NFRs). We delve into the critical role these elements play in managing aspects like data freshness, latency, and overall system performance. Through a use case of launching an API sourced from a database, we highlight potential failure points and the need for a proactive SRE approach to mitigate risks. By converting product requirements into SLIs and utilizing tools like New Relic, teams can create dashboards that monitor key metrics, ultimately avoiding costly pitfalls later in the development cycle. This paper is intended for Product Owners, Tech Leads and Engineers in the process of launching a net new product from the ground up.

Keywords: Site Reliability Engineering, SLAs, SLIs, SLOs, system performance

1. Introduction

Site Reliability Engineering (SRE) is a discipline that incorporates aspects of software engineering and applies them to infrastructure and operations problems. The primary goal of SRE is to create scalable and highly reliable software systems. As organizations strive to deliver consistent user experiences, the importance of establishing an SRE mindset early in the product development lifecycle cannot be overstated. This approach ensures that reliability is not an afterthought but a core component of the product strategy.

Key Concepts in SRE:

Service Level Agreement (SLA): An SLA is a formalized contract between a service provider and a customer that defines the expected level of service. SLAs typically include specific metrics such as uptime, response time, and availability, which are critical for meeting business and customer expectations.

Service Level Indicator (SLI): An SLI is a quantitative measure of some aspect of the service's performance, such as latency, error rate, or throughput. SLIs are used to track the health of a system and provide data that helps in assessing whether the system is meeting its Service Level Objectives (SLOs).

Service Level Objective (SLO): An SLO is a target value or range of values for an SLI that the service provider aims to

achieve over a defined period. SLOs are typically set lower than the SLA to provide a buffer and ensure that the SLA is met consistently.

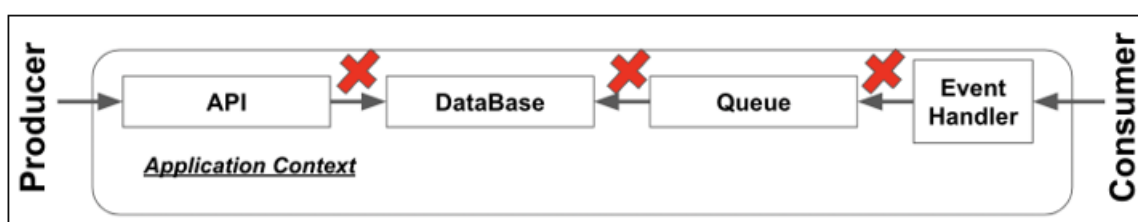
Non-Functional Requirement (NFR): NFRs refer to requirements that specify criteria not related to the specific behaviors or functions of the system but rather to attributes such as performance, security, and reliability. NFRs are critical for defining the quality and usability of the system.

By understanding and defining these key concepts early in the development process, teams can ensure that the system is designed with reliability and performance in mind, avoiding potential pitfalls later in the product life cycle.

Problem Statement

Consider a scenario where a company is preparing to launch an API that interacts with a database. This database, in turn, is populated by an event-driven system. Several potential failure points exist within this system architecture, including issues with data ingestion, processing delays, and database performance bottlenecks. These vulnerabilities highlight the need for clearly defined NFRs, particularly concerning data freshness and latency. Without proper SLAs and SLOs in place, the system could fail to meet user expectations, leading to dissatisfaction and potential revenue loss.

As you can see in the above image, each component will bring in various failure points such as Latency in the API response & Data Freshness Issue with Queue & Event Handler.



Approach to Establishing a SRE Mindset

To establish an SRE mindset, the first step is translating product requirements into measurable SLIs. For instance, if data freshness is critical, an SLI could be the time elapsed between data availability and its reflection in the API. Latency requirements might translate into SLIs that measure the time taken for a request to be processed and the response returned to the user. NFRs should encompass these metrics, setting clear expectations for performance and reliability.

A Failure Modes and Effects Analysis (FMEA) exercise is essential for identifying potential failure points within the system. By understanding where and how failures could occur, teams can proactively implement safeguards, such as redundancy or load balancing, to minimize impact. Once SLIs, SLOs, and NFRs are established, they should be visualized using dashboards. Tools like New Relic can be invaluable for monitoring these metrics in real-time, providing insights into system performance and alerting teams to issues before they affect users.

Potential SLIs that can be instrumented in this scenario

Given the scenario, the following SLIs can be defined to monitor the system's performance:

- **Data Freshness SLI:** Measure the time elapsed between when data is ingested into the event-driven system and when it becomes available in the database for the API to access. This could be tracked by logging the timestamp when data is received and comparing it to the timestamp when the data is available in the database.
- **Corresponding New Relic Metric:** Custom event or transaction timing, tracking the duration from data ingestion to availability in the database.
- **API Latency SLI:** Measure the time taken for the API to process a request and return a response to the user. This is critical for user experience and can be measured at different stages of the API request lifecycle, including time to connect to the database, query execution time, and response time to the client.
- **New Relic Metric:** Track API response times using APM (Application Performance Monitoring) features to monitor the latency of each API endpoint.
- **Database Query Performance SLI:** Measure the execution time of database queries. Slow queries can be a major bottleneck, impacting overall system performance and user experience.
- **New Relic Metric:** Monitor slow SQL queries, indexing issues, and database connection pool usage via New Relic's Database Monitoring features.
- **Event Processing Time SLI:** Measure the time it takes for events to be processed and written to the database. This is critical for ensuring that the system can handle the load and that data is up-to-date.
- **New Relic Metric:** Implement custom events or use New Relic's distributed tracing to track the time taken by each component in the event-processing pipeline.

Now Set SLOs Based on SLIs:

Once SLIs are identified, set realistic Service Level Objectives (SLOs) to ensure the system meets user expectations:

- **Data Freshness SLO:** For example, 95% of data should be available in the database within 2 minutes of ingestion.
- **API Latency SLO:** For example, 99% of API requests should have a response time of under 500 milliseconds.
- **Database Query Performance SLO:** For example, 98% of database queries should execute in under 100 milliseconds.
- **Event Processing Time SLO:** For example, 99% of events should be processed and available in the database within 3 minutes.

Configure New Relic Alerts:

Use New Relic's alerting system to create notifications based on the SLOs. For instance:

- **Data Freshness Alert:** Trigger an alert if the time for data to become available in the database exceeds the SLO threshold.
- **API Latency Alert:** Set up alerts for high latency, especially if it breaches the SLO.
- **Database Query Performance Alert:** Alert when slow queries exceed a certain threshold, indicating potential performance bottlenecks.
- **Event Processing Time Alert:** Notify the team if event processing times exceed acceptable limits, which could impact data freshness.

Visualize SLIs with Dashboards:

Create custom dashboards in New Relic to visualize SLIs in real-time. For example:

- **Data Freshness Dashboard:** Show the distribution of data availability times.
- **API Latency Dashboard:** Display real-time API response times, with breakdowns by endpoint.
- **Database Performance Dashboard:** Highlight slow queries, database connection usage, and other relevant metrics.
- **Event Processing Dashboard:** Visualize the end-to-end processing times for events.

By setting up these metrics and dashboards in New Relic, teams can continuously monitor the system's health and proactively address issues before they impact the user experience. This approach not only ensures that the system meets its SLAs but also provides the necessary visibility to maintain high reliability and performance throughout the product lifecycle.

This section details how to implement and monitor key SLIs using New Relic, ensuring that the system's reliability and performance are effectively managed.

2. Conclusion

Establishing an SRE mindset early in the product development cycle is essential for avoiding pitfalls related to system reliability and performance. By defining SLAs, SLIs, SLOs, and NFRs from the outset, teams can ensure that they are building systems that meet user expectations and can scale effectively. This proactive approach, supported by continuous monitoring and analysis through tools like New Relic, enables organizations to mitigate risks, reduce downtime, and deliver a superior user experience.

In the above paper, With a few simple steps we were able to clearly convert product requirements into SLIs, SLOs as well as New Relic Metrics which should give us increased confluence prior to product launch.

As IT projects grow increasingly complex, embedding SRE principles early on will be key to maintaining competitive advantage and ensuring long-term success.

References

- [1] Adhikari, S. et al. "Service Level Objectives and Indicators: A Key to Reliable Systems." *Journal of Software Engineering**, vol. 45, no. 3, 2020, pp. 233-245.
- [2] Mikoluk, K. "The Importance of SLAs in IT Project Management." *IT Professional Magazine**, vol. 32, no. 1, 2019, pp. 12-19.
- [3] Sharma, A. and Jain, R. "Implementing SRE Principles in Product Development." *International Journal of Computer Science and Technology**, vol. 51, no. 2, 2021, pp. 77-89.
- [4] Tanenbaum, A.S., and Van Steen, M. "Distributed Systems: Principles and Paradigms." *Prentice Hall**, 2007.