**Site Reliability Engineering**

1. **NFR (Non-Functional Requirement):** We define and establish the criteria that defines how a system should behave.

The System might work without them but may not meet user and business needs.

While the specifics will vary between products, having a list of these NFR types defined up front provides a handy checklist to make sure we are not missing critical requirements.

**Few types of NFRs:**

* Availability
* Performance
* Scalability
* Recoverability

**Example of NFR:**

* The time during which the application must be available.
* The latency for response time for a function of the product.
* Maximum time which the system can take to recover from failure.
* The highest amount of workload under which the system will still perform as expected.

1. **CUJ (Critical User Journey):** These can be used to represent the reliability and availability of the system. They are the best set of indicators for the overall health of the system.

These are of two types: **Human initiated** and **System initiated**.

A system can have many user journeys but for the purpose of measuring the system reliability, we want to pick just the ones which represent the core logic of the system.

**CUJ Example:**

| CUJ ID | CUJ Type | CUJ Description |
| --- | --- | --- |
| CUJ1 | Data Pipeline | Application consumes streaming messages from the upstream platform transforms the messages and publishes the messages to the destination |
| CUJ2 | UI | Existing users are able to log on to the system |

1. **SLI (Service Level Indicators):** This is the quantifiable metric that measures the performance and availability of a service or system. SLIs should be a measure of user experience. SLIs are expressed as a proportion or percentage, according to this pattern -

SLI= [Good Events/Valid Events]x 100%

SLIs should be used as measures for CUJs and we might aim to have one SLI per CUJ.

**Example for SLIs:**

**Coverage Example:** %of messages that are successfully transformed and written to destination -

SLI= [Messages transformed and published to destination/Total messages received]x 100%

**Availability Example:** % of successful login requests -

SLI= [Login requests with 2xx status response/total login requests] x100%

**Latency Example:** % of login requests serviced within 1s-

SLI= [total 2xx login responses within 1s/total login requests]x100%

1. **SLO (Service Level Objectives):** SLO are created by taking a SLI and adding a threshold and a time period. The SLOs should be set at a level that is an appropriate early warning level, before the customer experiences significant pain in using our service.

**Example of SLO:**

| CUJ ID | CUJ Type | CUJ Description | Measure | SLO |
| --- | --- | --- | --- | --- |
| CUJ1 | Data Pipeline | Application consumes streaming messages from the upstream platform transforms the messages and publishes the messages to the destination | Coverage | 99% messages are successfully published to destination in last 30 days |
| CUJ2 | UI | Existing users are able to login to the system | Availability | 99.5% login requests should result in 2xx in the last 30 days |

1. **Error Budget:** This defines the maximum acceptable amount of unreliability over a given period of time.

Error Budget = [100 - SLO]

**Example for Error Budget:**

**Example 1:**

If SLO is 99.9%

Then the Error Budget will be 0.1%

So, Error Budget will be 0.001\*30 day/month\*24 hours/day\*60 minutes/hour = 43.2 Minutes/Month

That means the total is 43200 Minutes in 30 days and we can afford to have a maximum of 43.2 minutes of error in a month after which our SLO will be breached.

Typically, we set the alerting for 80%, 85% or 90% of the Error budget consumed.

**Example 2:**

If an application processes 100 messages per day. And our SLO is 99% then we have an error budget of 1%.

So, Error budget is 0.001\*3000 = 30 messages/month. If we have set the alert for 90% of Error Budget, then we will get alert after the failure of the 27th message.

**The whole process describes as below:**

**NFR > CUJ > SLI > SLO > Error Budget > Alert.**

1. **Toil Automation:**

Toil is the kind of work that tends to be manual, repetitive, automatable, tactical, devoid of enduring value, and that scales linearly as the service grows.

**Example of Toil:**

* Handling quota requests.
* Applying database schema changes.
* Reviewing non-critical monitoring.

**Steps for Toils Automation:**

* Measure the time spent on different activities.
* Identify toil activities that are candidates for automation.