Lecture 3

Site Reliability Engineering

ECE 422: Reliable and Secure Systems Design

Instructor: An Ran Chen
Term: 2024 Winter

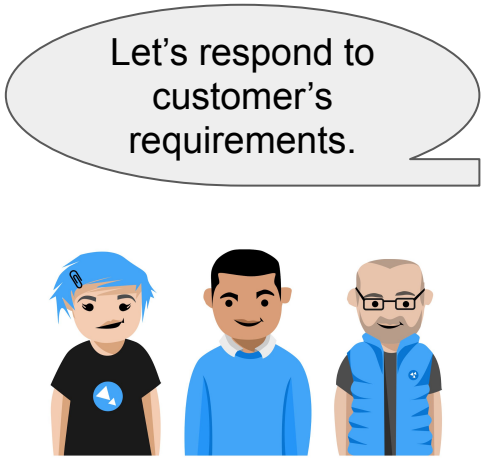
Schedule for today

- Key concepts from last class
- Site Reliability Engineering
  - SLA, SLO, SLI
  - Error Budget
- Availability
- Reliability

DevOps

Traditional software development

- Developer for delivery and agility, Operators for stability and reliability



Developers

Operators

DevOps

Traditional software development

- Developer for delivery and agility, Operators for stability and reliability

DevOps as a collaborative culture to speed up the deployments

- CI/CD for incremental changes and continuous delivery
- Tools and automation to support DevOps principles
  - Example of tools: Jenkins and Docker
- Continuous delivery automates the entire process up to the point of release
- Continuous deployment automates the entire process, eliminating the need for manual approval

SRE (Site Reliability Engineering)

SRE is a practice of automating IT operations tasks.

- Focuses on response time and software reliability
- Operations-oriented automation

Examples of IT operations tasks that are automated:

- Incident response
- Production system management
- Performance monitoring
- Emergency response

SRE (Site Reliability Engineering)

Why? To encourage frequent and small releases while maintaining system reliability

Benefits of SRE:

- Metric reporting
  - Use relevant metrics to discover issues
  - Translate metrics into concrete objectives
    - metric says 99.9% success rate, set objectives to 99% to beat the competition.
- Clarify and evaluate customer expectations
- Better productivity
  - SRE team as a specialized team for reliability
- Detect issues before they reach end-users
  - SRE proactively monitor the system at a higher-level perspective than developers
  - Find and fix issues that only occur during production

SRE (Site Reliability Engineering)

SRE has three key components:

- Service Level Agreement (SLA)
  - Agreement that the business team make with the customer
- Service Level Objective (SLO)
  - Objectives that the SRE team agrees to meet
- Service Level Indicator (SLI)
  - Real metrics on the system performance

Site reliability engineers build Service Level Indicators to drive Service Level Objective which are then used to negotiate the Service Level Agreement with the customer.

Availability

Availability is the percentage of time that a system is operational.

- Expressed in terms of the number of “nines” in the availability percentage

Common levels of availability and its corresponding downtime per year

- 90%, “One Nines”: Around 36 days of downtime per year
- 99%, “Two Nines”: Less than 4 days
- 99.9%, “Three Nines”: Just over 8 hours
- 99.99%, “Four Nines”: Less than an hour of downtime
- 99.999%, “Five Nines”: Only about 5 minutes of downtime

Availability

MTBF (Mean Time Between Failure) is the average time between two consecutive failures.

$$\text{MTBF} = \frac{\text{Operational Time}}{\text{Number of Failures}}$$

MTTR (Mean Time to Repair) is the average time it takes to restore the system after a failure

$$\text{MTTR} = \frac{\text{Total Repair Time}}{\text{Number of Failures}}$$

Availability is the percentage of time that a system is operational

$$\text{Availability} = \frac{\text{MTBF}}{\text{MTBF}+\text{MTTR}}$$

Availability

Given a system with MTBF of 100 hours and MTTR of 2 hours,

based on the availability formula:

$$\text{Availability} = \frac{100}{100 + 2} \times 100\% = \frac{100}{102} \times 100\% \approx 98.04\%$$

The system is estimated to be available 98.04% of the time.

Reliability

Reliability is the likelihood that a system will perform its function without failure.

- Calculated by MTBF (Mean Time Between Failure)

How to define reliability standards?

- Risks
  - Safety-critical systems vs personal websites
- Customer expectations
  - Translating “it should never fail” into “99.99% chance of successful operation”
  - Function, environment, and probability of failure
    - E.g., Sales transaction should work 99.99% of the time during Black Friday sales
- SLAs (Service level agreements)

Service Level Agreement (SLA)

An SLA is an agreement between a service provider and a customer that outlines the customer's expectations.

- Why? To provide a mutual understanding of service expectations between the business team and the customer
- Who is involved in SLA?
  - Tied to business and production decisions
  - Business team and customer, SREs are not involved
  - SLA is a legal binding agreement
  - But SRE can help the business team in understanding the likelihood and difficulty of meeting the SLOs contained in the SLA
  - SRE also help avoid the consequences of missed SLOs in the SLA

SLA

An SLA helps to:

- To improve trust with customers
- To improve communication with stakeholders
- To improve productivity

Examples of SLAs

- First Response Time (FRT)
- Time to Resolution (TTR)
- Availability

SLA vs KPI

A Key Performance Indicator (KPI) demonstrates how effectively a company archives key business objectives.

Examples of KPI:

- Sales revenue
- Website traffic

A SLA is a contract that outlines the expectations and responsibilities of both parties, and includes performance metrics for services the business offers.

Examples of SLA:

- Availability
- Response time

- SLA is a legal binding agreement.
- SLA for service expectations, KPI for performance
- SLA may contain penalties, KPI does not

SLA challenges

- Hard to track
- Do not always reflect business priorities
- Little flexibility in reporting
  - Yes or no question, either SLA is met or it is not
  - Causing hostility between business and SRE teams

Solution?

- Have the SRE collaborate with the business team to consider real-world scenarios.
- In the SLA, specify the Service Level Objectives (SLOs) which are the individual promises to the customer.

Service Level Objective (SLO)

An SLO is an agreement within an SLA about a specific metric like uptime or response time.

- Why? To provide the SRE team an understanding of whether they are in compliance with the SLA.
- Who is involved in SLO?
  - SRE team
  - Product team (e.g., product marketing, design and management)
  - May also include the development team

SLO opens discussion between teams to create a realistic definition of objectives.

Service Level Objective (SLO)

Examples of SLO:

- Uptime
  - E.g., eClass should be up 99.9% of the time
- Response time
  - E.g., eClass should load within one second for each page request
- Traffic
  - E.g., eClass should support a concurrent student count of 10,000 during course registration period.
- Success
  - E.g., eClass should maintain a success rate of 99.9% for course registration requests.

Service Level Indicator (SLIs)

An SLI is a metric that measures the actual performance and availability of a service.

- Why? We can't have SLOs without SLIs, SLIs measure the compliance of an SLO.
- If the SLO is set to 99.95% uptime, then the SLI is the actual measurement of your uptime.

Examples of SLIs:

- Uptime
  - Time that a service is usable
- Request latency
  - Speed at which the team respond to a request
- Error rate
  - Ratio of bad events (e.g., system failures) to total event

Error Budget

Error budget is how much downtime a system can afford without upsetting customers.

- An error budget is 1 minus the SLO of the service.
  - To respect the SLO of the service
- A 99.9% SLO service has a 0.1% error budget.

Why? To avoid overspending the time on reliability

- How can we tell if we are overspending? By error budget

> The team can “spend” error budget on making unreliable changes.

Error Budget

Example: Given an availability of 95% for SLO, what is your error budget per year?

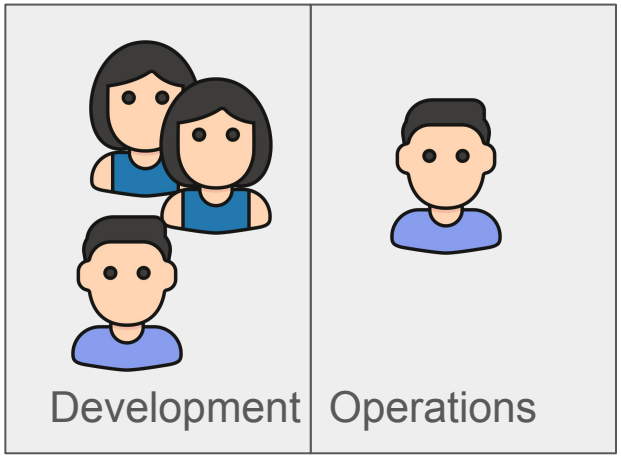
- 95% availability corresponds to 8.4 hours downtime per week
  - Downtime per week = 24*7*.0.05 = 8.4 hours
- 8.4 hours is the error budget you can spend in one week on making the system more reliable or pushing for new features.
- Whether you spend it on making the system more reliable or new features, depends on whether or not you exceed your current budget

Error Budget

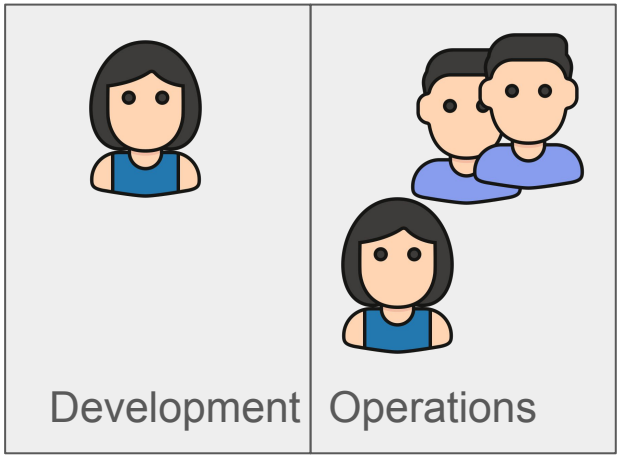
What happen if you overspend your error budget?

- Shift the priorities from development to improving the reliability of the system
- Once your error budget has recovered, you can work on functionalities again

When error budget is underspent



When error budget is overspent



SRE best practices

- Use non-technical language in SLAs
  - Keep SLA language simple to avoid misunderstandings with the customer
- Fewer SLOs as possible
  - Commit to fewer SLOs, and focus on the objectives that matter to the customer
- Build an error budget
  - Accept failures, leave room for agility (changes and new features)
- Don’t shoot for the moon
  - Better to underpromise and overdeliver
- Do not include every metric as an SLI
  - Pick the SLIs based on your core SLOs

SRE vs DevOps

SRE focuses on reliability

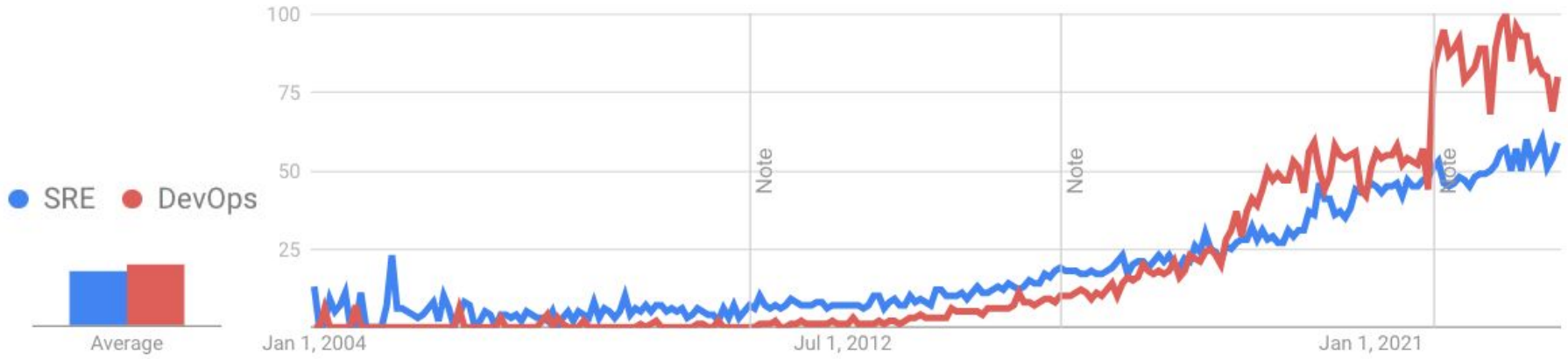
- Operating and monitoring
- Incident response
- Capacity planning

DevOps focuses on delivery

- CI/CD
- Release automation
- Configuration management

Note: SRE is to DevOps what Scrum is to Agile; one implementation of a philosophy and principles, but not a 100% subset.

Trends over time in Canada

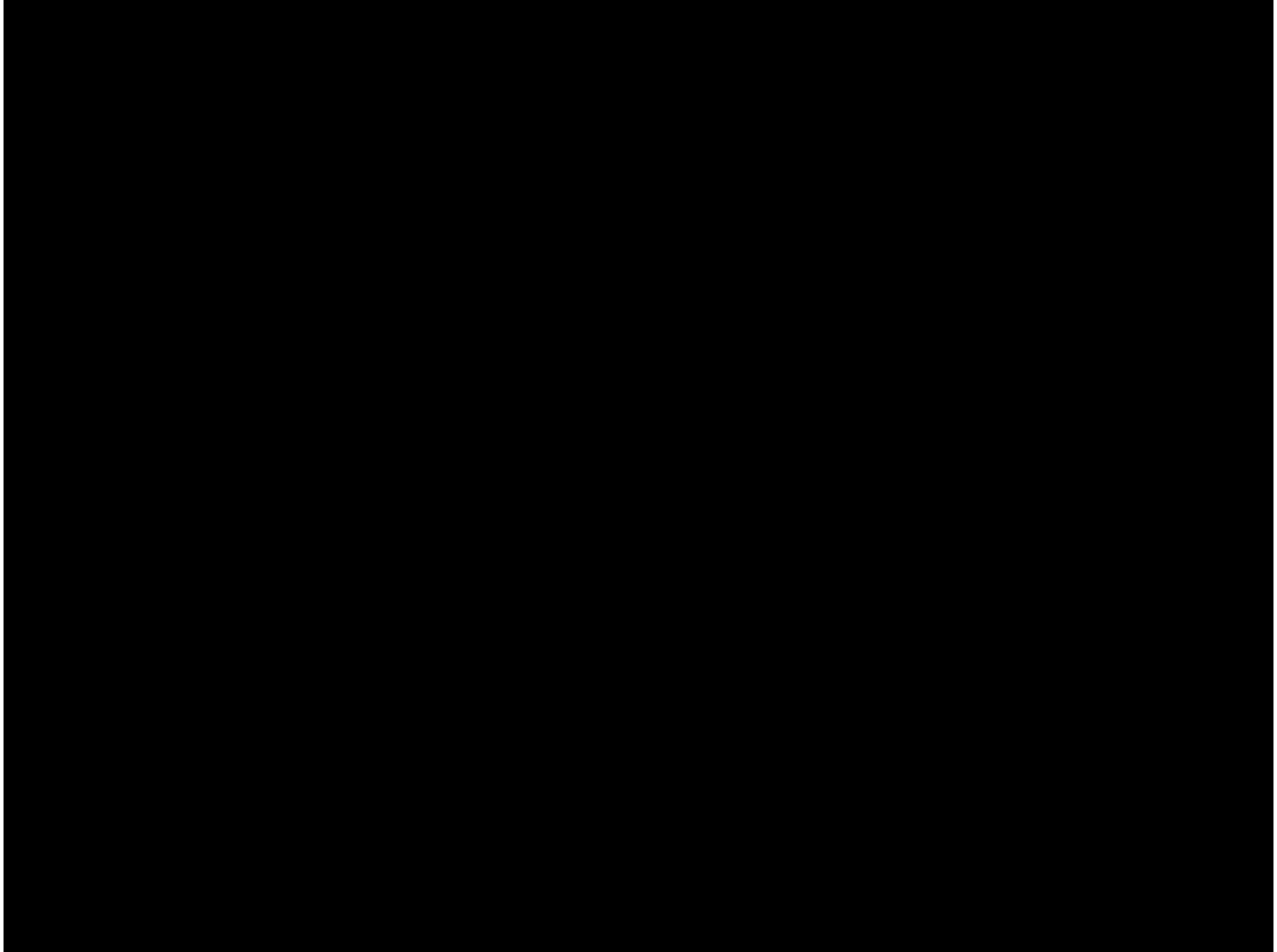


DevOps and SRE are not two competing methods, but rather similar approaches to design and deliver better software faster.

- SRE as one implementation of DevOps best practices

Google SRE team explains SRE

- Article: Lessons Learned from Twenty Years of Site Reliability Engineering
- Video: Site reliability engineers at Google explaining SRE



Google SRE team explains SRE

- SRE is a discipline that include software, hardware, networking, and human
- SRE as a new role, a team that is highly-skilled in monitoring and incident responses
- As a SRE team, the goal is to
  - Improve the system overtime
  - Mitigate the issues
  - Understand and learn from the issues

From a SRE at Shopify

Team structure

- Operations engineer only (including the manager is an operation engineer)
- Vs. DevOps that is a cross-functional team that includes developer, designer, data scientists, product manager

Daily tasks

- Maintain the infrastructure so that it fulfills SLO (service-level objective) under different loads