

Service Levels, Error budgets and why your dev teams should care

Lauri Suomalainen, 2024

verifa

About this talk

- This is a fundamental level talk
- Topics covered:
 - Service Level Agreements, Objectives and Indicators
 - What to measure?
 - Why should the development team care about measurements?
 - Error budgets

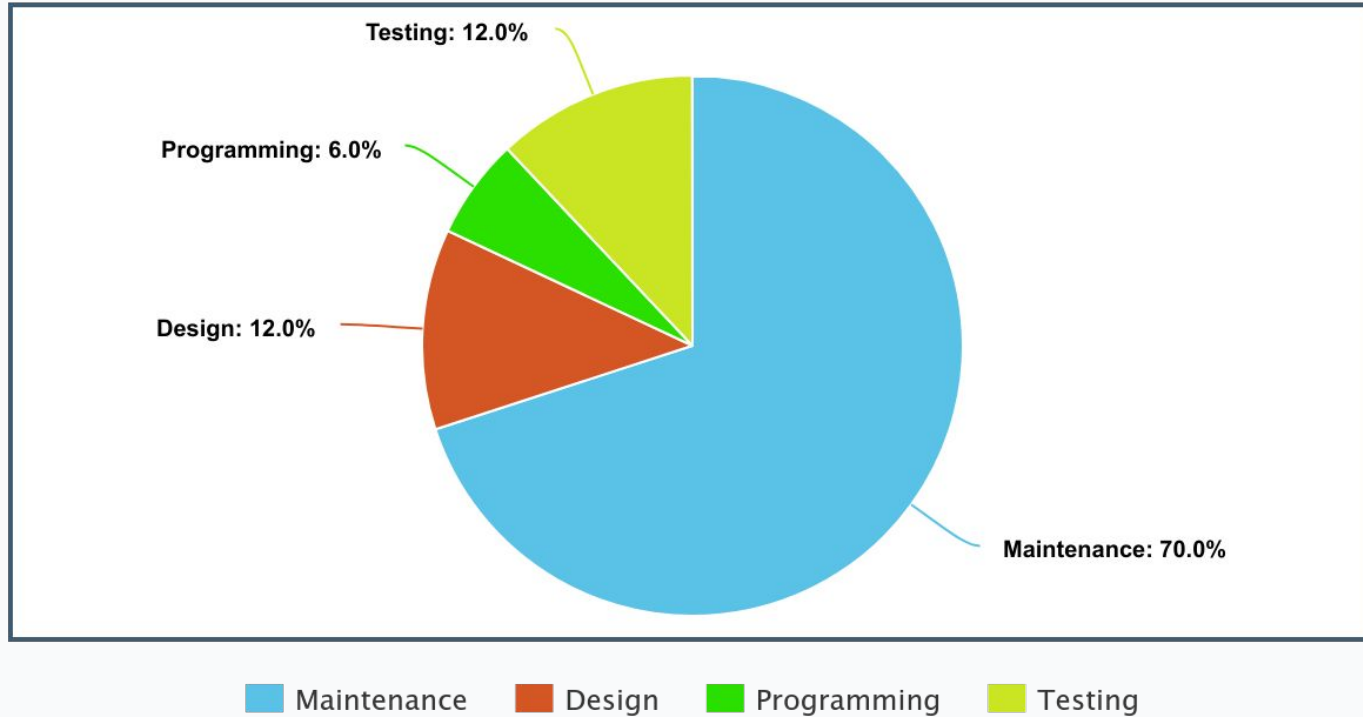
Motivation

- People don't necessarily know how Service Levels relate to development
- Operations side focus
- "I've seen error budgets implemented once or twice" – Ferrix Hovi
 - Do people just not understand them?

**Why Service Levels
matter to software
development teams?**

verifa

Software Lifecycle costs
Code Readability Management of High-level Programming Languages: A Comparative
Study, Tariq et al., 2020



meta-chart.com

verifa

Downtime has real consequences

The Figures

\$129 million

annual cost to a typical large plant through unplanned downtime (up 65% in two years)

\$1.5 trillion

annual downtime losses for Fortune Global 500 firms

70% rise

in losses for Fortune Global 500 firms compared to two years ago

11% of annual revenues

amount lost to unplanned downtime over the Fortune Global 500 companies

Service Levels tell you how your system performs from end-users' perspective

- Service levels can also communicate system health internally
- Key acronyms: SLA, SLO and SLI
- SLIs provide the data to form the SLOs, SLAs are based on the SLOs.

Service Level Indicators

- Every SLI is a metric, but not every metric can be SLI.

Queries per second?

Nope!

95th percentile latency?

Yep!

Number of pods in a
Kubernetes cluster?

Nope!

HTTP Error codes in a
minute?

Nope... but what about %?



Service Level Objectives

- Internal targets
 - Some choose to publish these to end-users to manage expectations
- Thus, should correlate with user expectations.
- Breaking SLO should warrant action.

Service Level Agreements

- The NINES!
- A contractual promise to customers
 - Breaching has consequences
- Usually business and/or legal oriented. Can be a selling feature.
- SLAs should be more relaxed than SLOs.
- Not all services need SLAs.

What to measure?

- Things to measure:
 - Error code frequency
 - Latency
 - Data availability and durability
 - Correctness(?)
- “Availability” is not necessarily a good metric.

Simplified
reliability

$$\frac{\text{Available time}}{\text{All time}} = \text{Reliability \%}$$

User centric
reliability

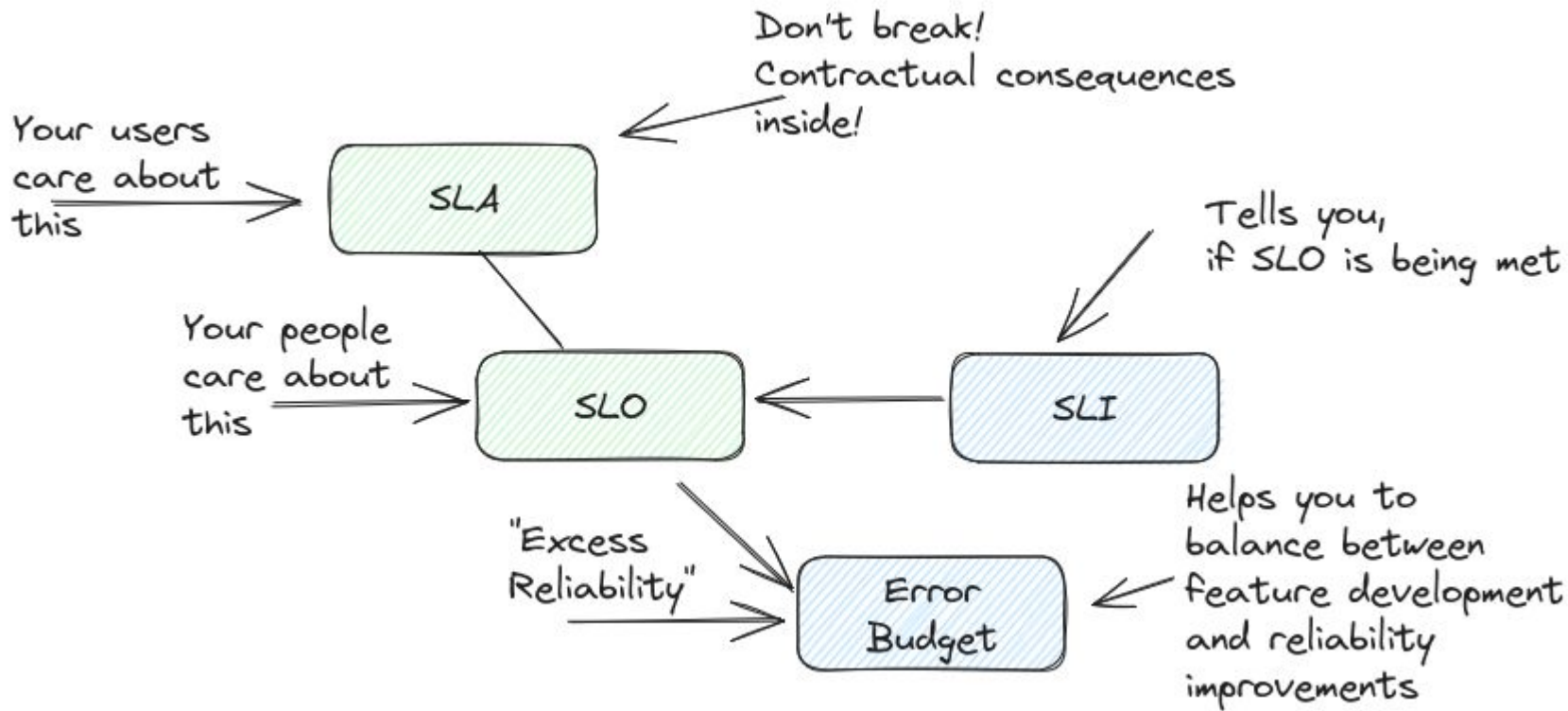
$$\frac{\text{Good interactions}}{\text{All interactions}} = \text{Reliability \%}$$

Reliability percentage

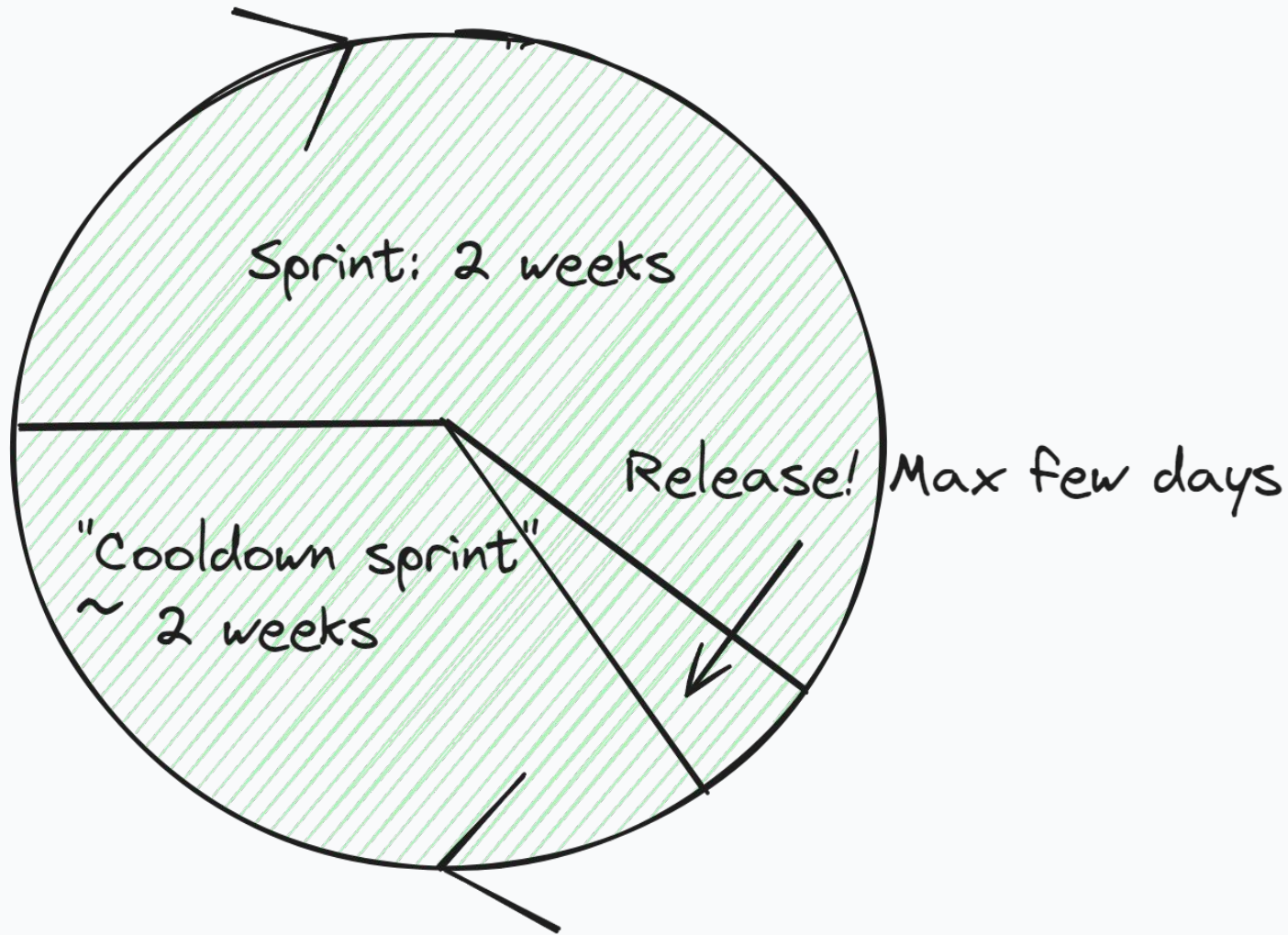
verifa

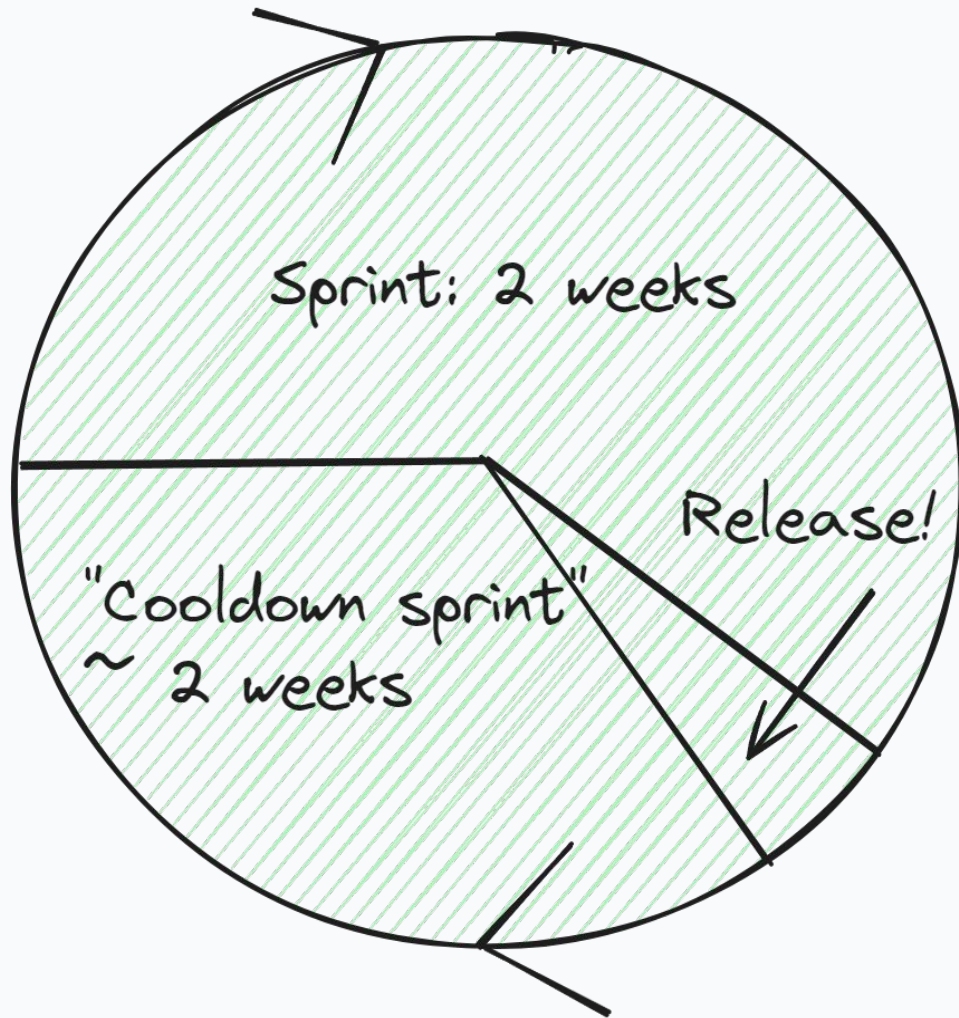
Tips for making best use of Service Levels

- Build from bottom up.
- Focus! Have as few as SLOs as possible.
- SLOs: 100% Reliability is not feasible.
- Don't overachieve.
- Don't aim for perfection.
- Nothing is static.



SLO Hierarchy





Here be things that rustle:

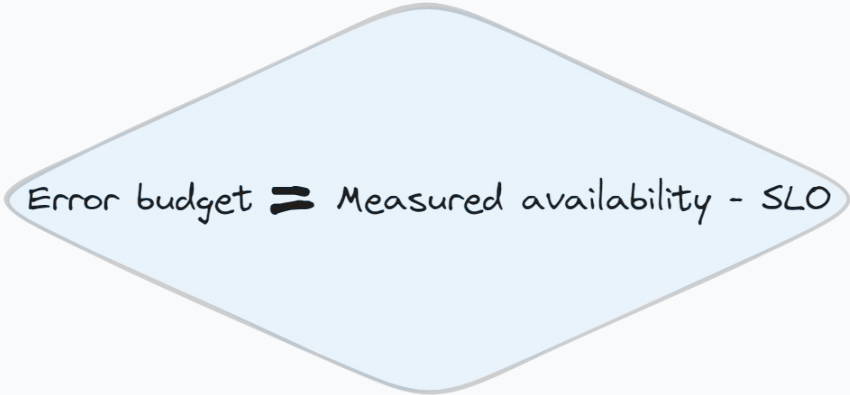
- SLOs are not OKRs
- Cooldown sprints are arbitrary like this!

Release! Max few days



verifa

Error budgets



Error budget = Measured availability - SLO

- Communication medium between Dev and Ops.
- Budget can be used to do risky things:
 - More frequent pushes
 - Bigger Canary Groups
 - Relaxed testing?

Availability Cheat Sheet

Availability level	Downtime per year	Downtime per quarter	Downtime per month	Downtime per week	Downtime per day	Downtime per hour
90%	36.52 days	9.13 days	3.04 days	16.80 hours	2.40 hours	6.00 minutes
95%	18.26 days	4.57 days	1.52 days	8.40 hours	1.20 hours	3.00 minutes
99%	3.65 days	21.91 hours	7.30 hours	1.68 hours	14.40 minutes	36.00 seconds
99.5%	1.83 days	10.96 hours	3.65 hours	50.40 minutes	7.20 minutes	18.00 seconds
99.9%	8.77 hours	2.19 hours	43.83 minutes	10.08 minutes	1.44 minutes	3.60 seconds
99.95%	4.38 hours	1.10 hours	21.91 minutes	5.04 minutes	43.20 seconds	1.80 seconds
99.99%	52.59 minutes	13.15 minutes	4.38 minutes	1.01 minutes	8.64 seconds	0.36 seconds
99.999%	5.26 minutes	1.31 minutes	26.30 seconds	6.05 seconds	0.86 seconds	0.04 seconds

Calculations are based on the average Gregorian year: 365.2425 days

<https://availability.sre.xyz/>

verifa

Learn availability math with Lauri!

- SLO 95% availability = 1,52 days downtime per month = 72 minutes per day
- Our availability is 97!
 - Software FUBAR only mere 43,2 minutes each day!
- Our error budget is $97\% - 95\% = 2\%$
- $72 \text{ minutes} - 43,2 \text{ minutes} = 28,8 \text{ minutes}$ per day.



verifa

**DEPLOYED
TO PRODUCTION.
IT'S AN
OPS PROBLEM NOW.**



**ME
REMEMBERING THAT
I'M A
DEVOPS ENGINEER**



Instructions unclear, halved my error budget

- Downtime from ~ 43 minutes per day to 58 minutes per day
- Availability from 97% to 96%

BUT DO NOT FEAR!

- Error budget is yours to spend, not to save.
 - Even now, you would still have 14 minutes 24 seconds of budget PER DAY.
- Scheduled downtime? Cooldown sprint?
- What if our daily downtime was over 72 minutes?

Conclusions

- Service Levels and Error budgets allow operation and developers to align their goals
- They enable proactive decision making.
- Help to steer and prioritise software development and strike the balance between change and stability.
- Delivering software is a customer service job. SLOs focus on the end user.

Thank you!

Service Levels, Error budgets and why your dev teams should care

[Lauri Suomalainen](#), 2024
lsuomalainen@verifa.io

verifa

Based on my blogpost at verifa.io

References & Further reading:

- [Lifecycle cost piechart](#)
- [Siemens: True cost of Downtime](#)
- [Google's SRE book](#)
- [This google course goes deeply into the topic.](#)