Don’t get out of bed for anything less than an SLO

Joe Blubaugh, Conf42 Devops 2023

Joe Blubaugh

Principal Software Engineer, Grafana

10 years building and operating distributed systems at Google, Twitter, and startups

Love/Hate relationship with On-Call shifts

Agenda

01

On-Call Misery

Bad On-Call Makes People Quit. How can we improve it?

02

Explaining SLOs

A way out of noisy, useless alerting.

03

Effective SLOs

How to define your Indicators and Objectives

04

SLO Alerting

Not too fast, but not too slow, either!

Burnout 

Burnout **kills** careers

Burnout **kills** software teams. People quit

Turnover **kills** engineering departments

75% of software engineers are experiencing symptoms of burnout

What causes burnout?

Unclear expectations

Lack of autonomy

Inability to unplug

Inability to ship

On-call affects burnout

Unclear expectations	**(Who handles the alert?)**
Lack of autonomy	
Inability to unplug	**(Paged at 3am)**
Inability to ship	**(Spent all day on alerts)**

Useful On-Call

The right person for the job
Meaningful alerts
Actionable alerts
Getting paged for
real issues

Stressful On-Call

Frequent alerts
Unactionable alerts
False-positive alerts
Getting paged at 3am

SLOs:
Measure what matters

What are they good for?

Service Level Objectives are a tool for

- Defining the important behaviors of a system
- Understanding how to measure them
- Making decisions about the health of a system

SLOs help you prioritize work and alert on the important stuff

Service Level

What utility does your system provide, and how is it measured?

Objective

What value for the measurement is good enough?

Service Levels

What are the services you're providing to users, clients, customers?

What do your users **want** from the system?

How do you measure their quality?

Define an indicator

Prose

Can users sign into our app?

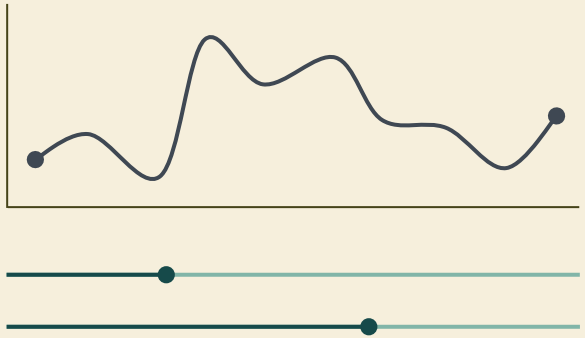
Metrics

login_requests_count

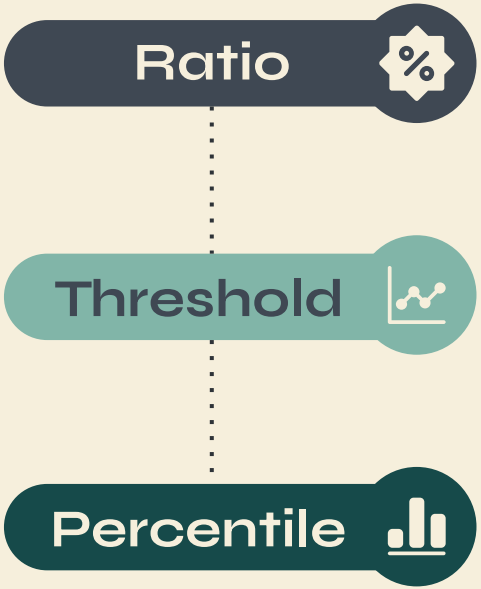
Math

Should be a number between 0 and 1.0

```
sum(
  rate(
    login_requests_count{
      status_code!~"5.."
    }[5m])
) / sum(rate(login_requests_count[5m]))
```



Common Indicators



Successful requests / All requests

Data backup throughput is over 40 Gbps

The 99th percentile latency is below 500ms

The Objective

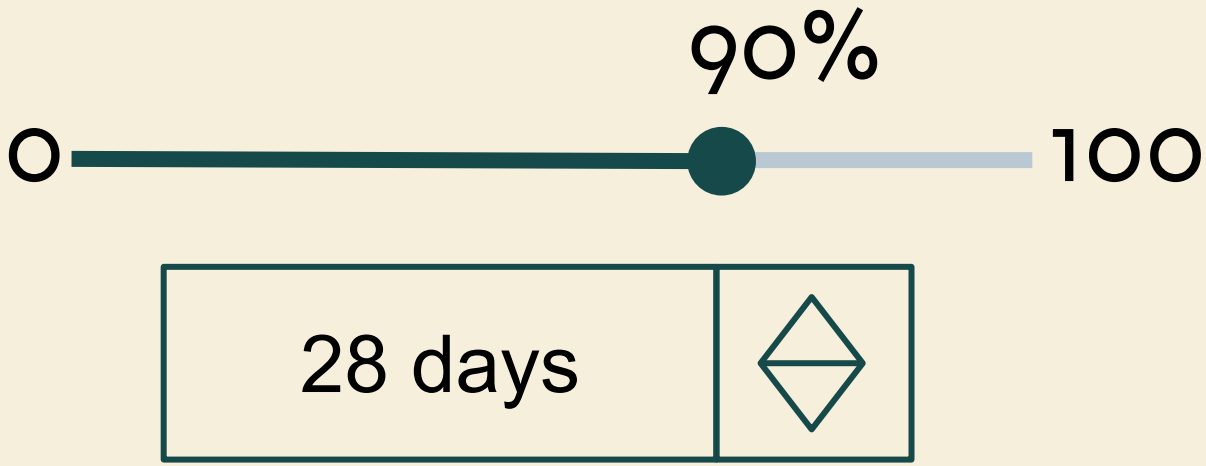
What level of quality is acceptable to you?

Only the

Highest Quality

(of course)

Objective is a target



Permitted Downtime

Time Window	Objective	Downtime
7 days	99%	1 hour, 41 minutes
7 days	99.9%	10 minutes
28 days	99%	6 hours, 43 minutes
28 days	99.9%	40 minutes
28 days	99.999%	25 seconds

Set realistic objectives

It’s better to start with a low objective and get stricter

Leave some safety margin

Look at the indicators over the last few months to choose initial objectives

Blanket, top-down objectives **don’t work**

SLO-based alerts

Monitor system health based on symptoms that users experience. Alert your operators when they're in danger of using up their error budget, and prioritize response time by how quickly they're using it.

Symptoms, not causes

Alerting is not **maintenance**. Alerting is for **emergency triage**

Your systems deserve checkups, where you look at causes like CPU usage, low volumes of 500 responses, etc.

But you shouldn't get up at 3am for that

Not too early, not too late

Here comes the math!

Performance as cash flow

Your objective gives you an **error budget**

You have an error **burn rate**

You need to pay attention when you're **spending too quickly.**

Principles of alerting

Should wake me up

Sustained, high burn rate
Transient, **extremely** high burn

I should look in the morning

Sustained, moderate burn rate

Part of system check-up

Sustained, low burn rate

Shouldn't alert

Transient, moderate burn

Multiple time windows
Multiple rates

Short Window	Long Window	Average Burn Rate	Severity
5 minutes	1 hour	Budget * 14.4	urgent
30 minutes	6 hours	Budget * 6	high
2 hours	1 day	Budget * 3	moderate
6 hours	3 days	Budget * 1	low

Remember

01

On-Call Misery

Improve on-call with actionable, useful alerts

02

Explaining SLOs

A tool for making monitoring more useful

03

Effective SLOs

Define the important behavior
Set realistic objectives

04

SLO Alerting

Not too fast, but not too slow, either!

THANKS!

joe.blubaugh@grafana.com
@joeblubaugh@hachyderm.io