Google

Measuring Reliability

...what got you here won't get you there

Štěpán Davidovič, Google

for SRECon EMEA 2022, Oct 25th-27th

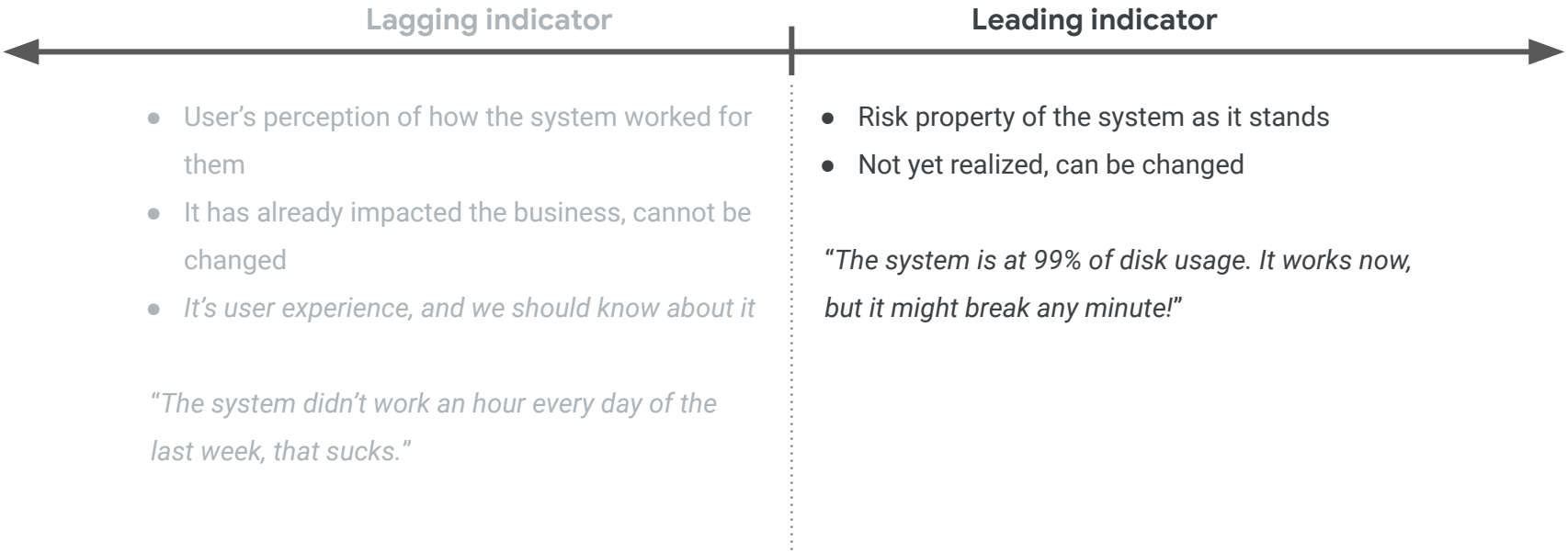
What is “reliability”? It’s a fuzzy concept!

What is “reliability”? It’s a fuzzy concept!

- User’s perception of how the system worked for them
- It has already impacted the business, cannot be changed
- *It’s user experience, and we should know about it*

“The system didn’t work an hour every day of the last week, that sucks.”

What is “reliability”? It’s a fuzzy concept!



What is “reliability”? It’s a fuzzy concept!

Lagging indicator

- User’s perception of how the system worked for them
- It has already impacted the business, cannot be changed
- *It’s user experience, and we should know about it*

“The system didn’t work an hour every day of the last week, that sucks.”

Leading indicator

- Risk
- Not yet realized, can be changed

“The sy
but it will break any minute!

Leading indicator is important, but out of scope here

This slide deck will focus on the ability to *measure what has already happened*, and draw conclusions from that.

Measure reliability?

“Define your SLIs/SLOs!”

...let’s look at that more!

Book covers by O'Reilly Media:

- https://www.oreilly.com/library/view/site-reliability-engineering/9781491929117/
- https://www.oreilly.com/library/view/implementing-service-level/9781492076803/

The SLO Model Recap

Service Level Indicator (SLI)

Time series data which can tell us how good the level of service is. Often from logs or sampled counters.

Examples:

- tuple: `{HTTP 500s, all HTTP responses}`
- ratio: `responses under 200ms / all responses`

The SLO Model Recap

Service Level Indicator (SLI)

Time series data which can tell us how good the level of service is. Often from logs or sampled counters.

Examples:

- tuple: `{HTTP 500s, all HTTP responses}`
- ratio: `responses under 200ms / all responses`

Service Level Objective (SLO)

Predicate on a mathematical function applied on SLI. Has free parameters. Aim is to keep this predicate true.

Mathematical example:

$$\frac{\sum_{window} |successful\ requests|}{\sum_{window} |total\ requests|} \geq objective$$

Organizational example:

- *"We hit/missed our SLO last quarter"*

The SLO Model Recap

Service Level Indicator (SLI)

Time series data which can tell us how good the level of service is. Often from logs or sampled counters.

Examples:

- tuple: `{HTTP 500s, all HTTP responses}`
- ratio: `responses under 200ms / all responses`

Service Level Objective (SLO)

Predicate on a mathematical function applied on SLI. Has free parameters. Aim is to keep this predicate true.

Mathematical example:

$$\frac{\sum_{window} |successful\ requests|}{\sum_{window} |total\ requests|} \geq objective$$

Organizational example:

- *"We hit/missed ... last quarter"*

Many free parameters to choose:

- time window
- objective
- (non-obvious) frequency of predicate evaluation

Answering our reliability questions

When we say "*measure reliability*", we want our data to give us some insight. **We are answering questions, by using our available data.**

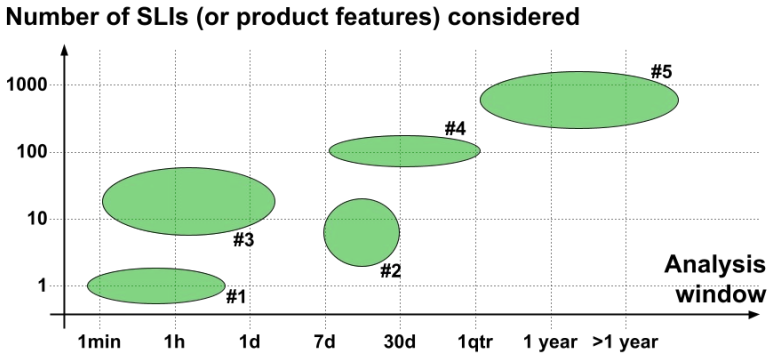
But there is no single reliability question! An engineer oncall is in a different situation than a CEO strategizing.

> *The purpose of computing is insight, not numbers.*
>
> Richard Hamming

Our reliability questions?

Some illustrative examples:

1. Oncall engineer responds to and mitigates an incident. Did their action help?
2. Team manager holds weekly production review team meetings, are there creeping problems?
3. Customer support asks if a customer has problems?
4. SVP wants to understand customer-perceived reliability of product portfolio before meeting with the customer.
5. CEO wants to know if company's reliability is getting worse. Do we need to pivot?



Our reliability questions?

Some illustrative examples:

1. **Oncall engineer responds to and mitigates an incident. Did their action help?**
2. Team manager holds weekly production review team meetings, are there creeping problems?
3. Customer support asks if a customer has problems?
4. SVP wants to understand customer-perceived reliability of product portfolio before meeting with the customer.
5. **CEO wants to know if company's reliability is getting worse. Do we need to pivot?**

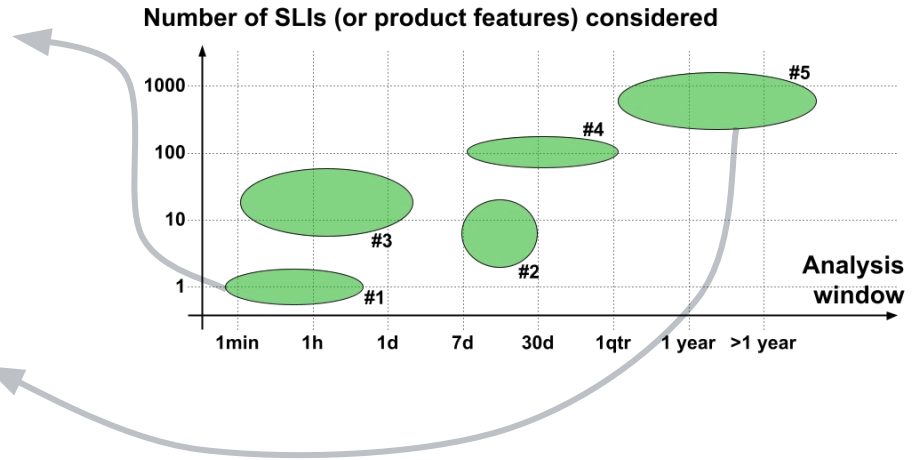


Illustration #1: Oncall engineer

Engineer got paged with the alert “`SLI_Suddenly_Awful`”.

After an hour of debugging, engineer tried a mitigation.

How do they know if it helped?

Illustration #1: Oncall engineer

Engineer got paged with the alert “`SLI_Suddenly_Awful`”.
After an hour of debugging, engineer tried a mitigation.
How do they know if it helped?

If it were me:

1. Wait for a while
2. Look at whether the SLI has recovered to above SLO
   (or maybe even to previous levels)

$$SLI_{last\ 15\ minutes} \geq SLO$$

$$SLI_{last\ 15\ minutes} \approx SLI_{before\ incident}$$

Illustration #1: Oncall engineer

Engineer got paged with the alert "`SLI_Suddenly_Awful`".
After an hour of debugging, engineer tried a mitigation.
How do they know if it helped?

If it were me:

1. Wait for a while
2. Look at whether the SLI has recovered to above SLO
   (or maybe even to previous levels)

We built an intuitive, ad-hoc model to answer our question!

We ignore SLO window

$$SLI_{last\ 15\ minutes} \geq SLO$$

$$SLI_{last\ 15\ minutes} \approx SLI_{before\ incident}$$

We also ignore the objective

Illustration #2: CEO wonders

CEO is wondering whether reliability of the company's product is getting worse, and new reliability work needs to be prioritized.

How do they interpret the reliability data?

Maybe they compare how many SLOs were met, month to month?

Illustration #2: CEO wonders

Comparing how many SLOs were met month to month

Let's say we have 200 SLOs. **Are we getting less reliable?**

	January	February	March
Total SLOs	200	200	200
SLOs violated	3	4	12

Illustration #2: CEO wonders

Comparing how many SLOs were met month to month

Let's say we have 200 SLOs. **Are we getting less reliable?**

Naive answer: Yes, because 12 (6.0% of total) SLOs not met is (a lot) more than 3 (1.5%) or 4 (2%).

	January	February	March
Total SLOs	200	200	200
SLOs violated	3	4	12

Illustration #2: CEO wonders

Comparing how many SLOs were met month to month

Let's say we have 200 SLOs. **Are we getting less reliable?**

~~Naive answer: Yes, because 12 (6.0% of total) SLOs not met is (a lot) more than 3 (1.5%) or 4 (2%).~~

Better answer: We can't tell by interpreting this data this way.

	January	February	March
Total SLOs	200	200	200
SLOs violated	3	4	12

Illustration #2: CEO wonders

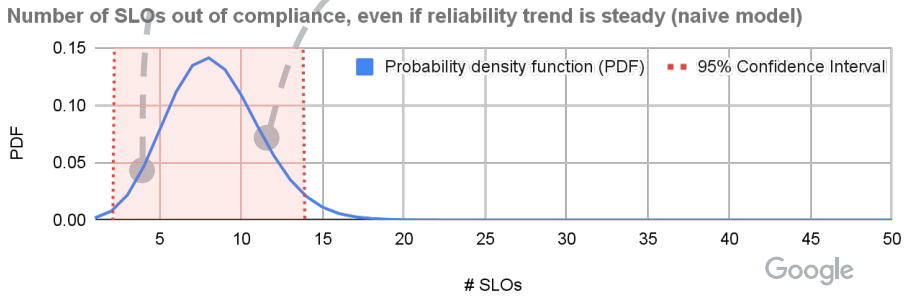
Comparing how many SLOs were met month to month

Illustrating using a binomial model:

- Let's say probability of not meeting SLO each month is 1/24
- Then 95% confidence interval is from 3 to 13 SLOs not met, *even if the average reliability doesn't change*

Naive idea was non-obviously dangerous. **The more impactful (=costly) the decisions, the more important to check your methods!**

	January	February	March
Total SLOs	200	200	200
SLOs violated	3	4	12



This illustration model is *very flawed* (SLOs typically not IID, etc.). It's only an illustration, prompting going beyond the naive answer.

SLI/SLO model got us here, but...

We need more models. The examples show that SLO model alone isn't sufficient. **In practice we build ad-hoc models in our heads, intuitive but sometimes dangerously wrong.**

The SLI/SLO model helped us make good progress in reliability! But **what got us here won't get us there**. The illustrations were strawman, but the problem is real.

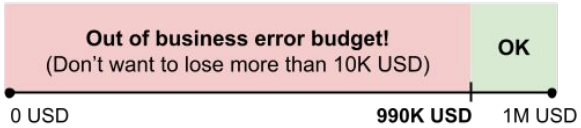
To figure our next steps, **let's understand some of the limitations and assumptions of SLIs and SLOs**.

Error Budgets Have Error Margins

That's okay! But do you know what yours are?

We establish our "error budget" based on acceptable losses

- 100% availability is fanciful
- Since we make 1M USD, we set a failure budget 10K USD



Error Budgets Have Error Margins

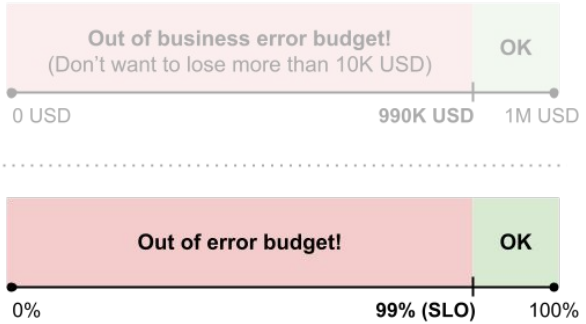
That's okay! But do you know what yours are?

We establish our "error budget" based on acceptable losses

- 100% availability is fanciful
- Since we make 1M USD, we set a failure budget 10K USD

We set our SLO to correspond to our error budget

- We make 1M USD/yr, so 10K implies 99% SLO

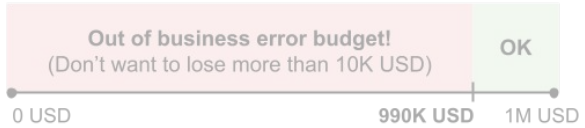


Error Budgets Have Error Margins

That's okay! But do you know what yours are?

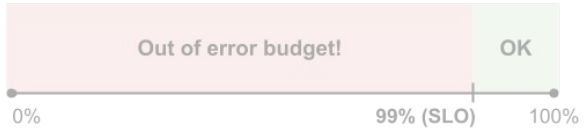
We establish our "error budget" based on acceptable losses

- 100% availability is fanciful
- Since we make 1M USD, we set a failure budget 10K USD



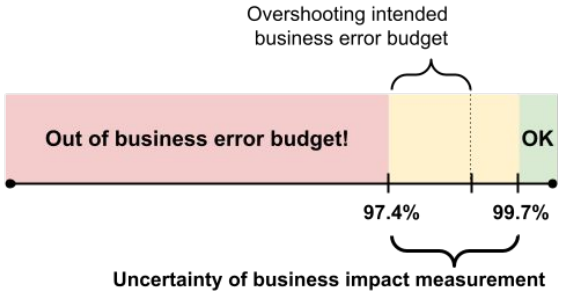
We set our SLO to correspond to our error budget

- We make 1M USD/yr, so 10K implies 99% SLO



We measure incident impact — but inaccuracy is a problem!

- Impact assessment may be inaccurate, e.g. order of magnitude!
- Estimate your inaccuracy: Ask three *independent* incident reviewers for impact estimate, and observe variance
- *This is a problem even without any black swan events!*



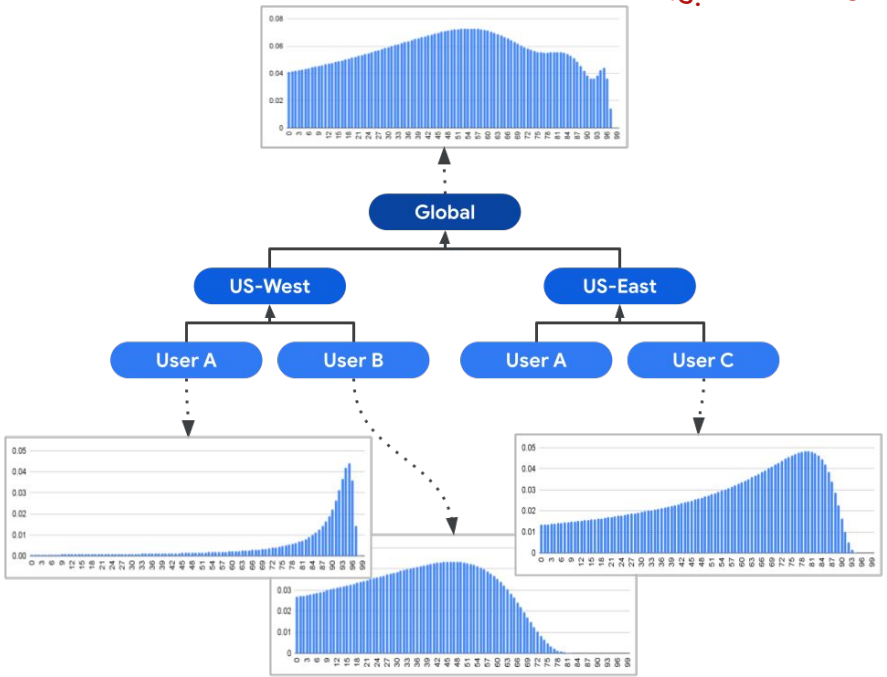
SLO Model Assumes Linearity

...in time and space!

Aggregating upwards hides bad behavior

- What if your product *never works* for *a handful of users*?
- For SLI/SLO, it's the same as if it didn't work *a little bit* for *all users*!

Aggregation to global (or zonal) SLO can hide severe problems!



Synthetic illustration data

SLO Model Assumes Linearity

...in time and space!

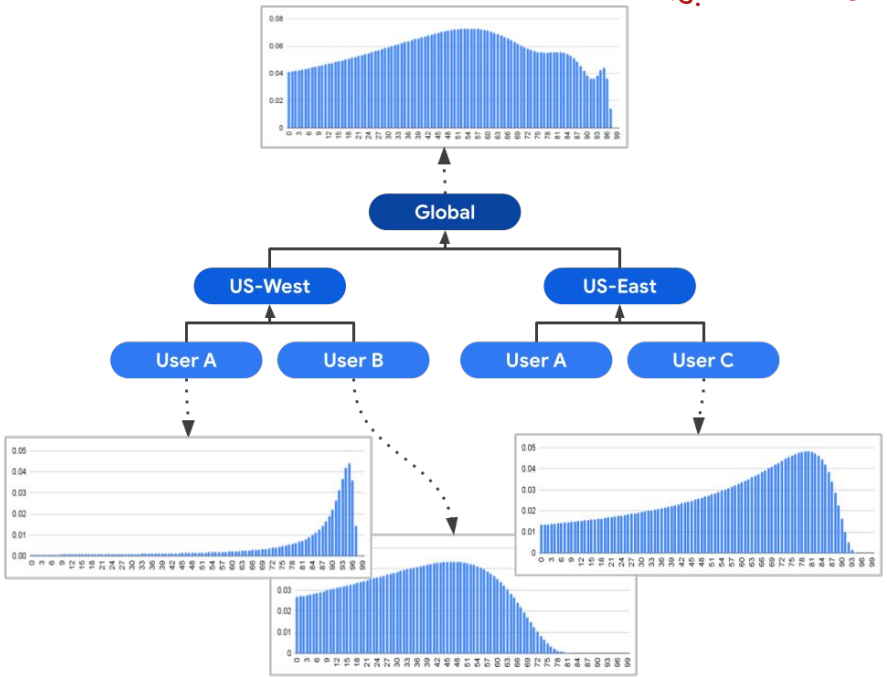
Aggregating upwards hides bad behavior

- What if your product *never works* for *a handful of users*?
- For SLI/SLO, it's the same as if it didn't work *a little bit* for *all users*!

SLIs assume all requests are equal

- They can have different costs, user utility, or revenue
- Human-curated SLI grouping helps, e.g. group by API call or location

Aggregation to global (or zonal) SLO can hide severe problems!



Synthetic illustration data

SLO Model Assumes Linearity

...in time and space!

Aggregating upwards hides bad behavior

- What if your product *never works* for *a handful of users*?
- For SLI/SLO, it's the same as if it didn't work *a little bit* for *all users*!

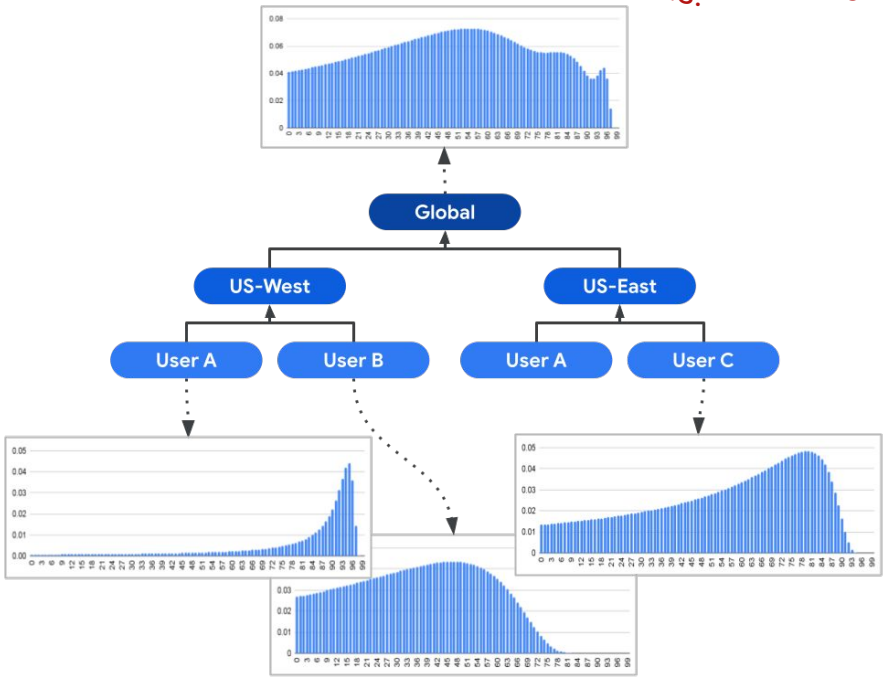
SLIs assume all requests are equal

- They can have different costs, user utility, or revenue
- Human-curated SLI grouping helps, e.g. group by API call or location

Human-designed grouping is not always possible

- Example: Free-form SQL queries for a database, or arbitrary input video formats for video encoder

Aggregation to global (or zonal) SLO can hide severe problems!



Synthetic illustration data

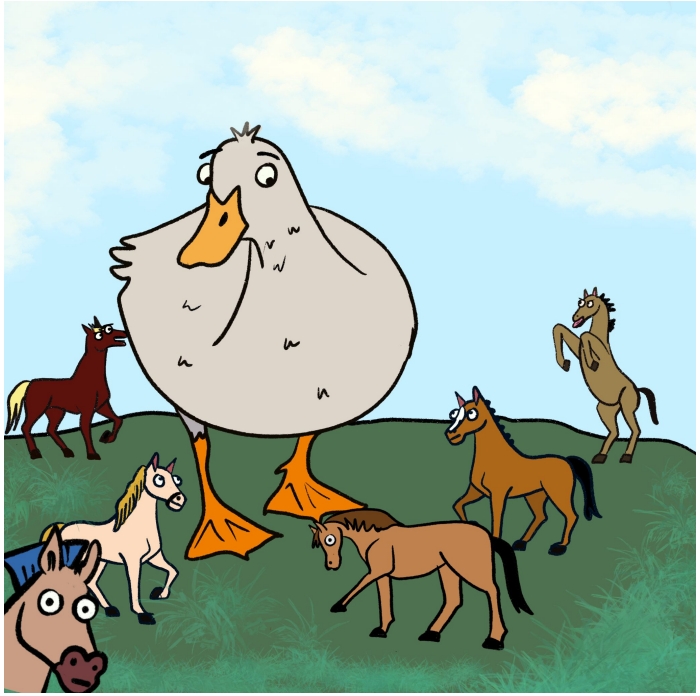
SLO Model Assumes Linearity

...in time and space!

SLO model aggregates over time

- 1000x 1-minute full outages is equal to...
- ...1x 1000-minute full-outage?
- ...2x 1000-minute half-outages?

To your users and your business, this difference may matter



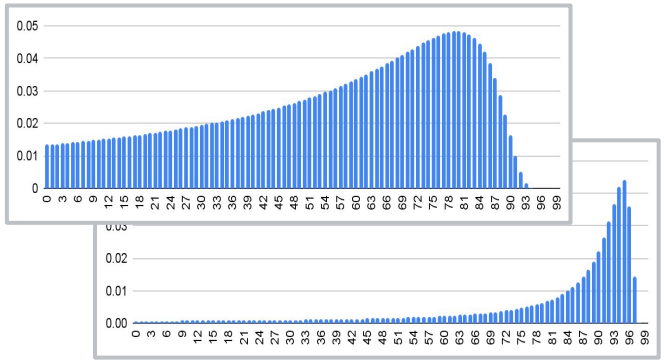
WOULD YOU RATHER FIGHT 1 HORSE-SIZED DUCK OR 100 DUCK-SIZED HORSES?

SLIs Aren't The Best Data

...they are the easiest data

Practical SLIs are often limited to sources which have:

- High sample rate
- Low cost to sample (and interpret)
- Low sampling latency



SLIs Aren't The Best Data

...they are the easiest data

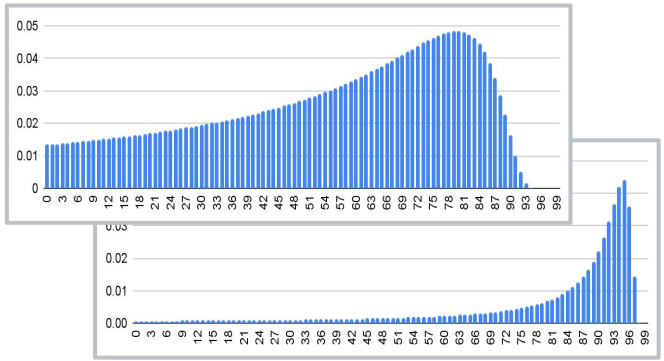
Practical SLIs are often limited to sources which have:

- High sample rate
- Low cost to sample (and interpret)
- Low sampling latency

We should integrate other useful sources:

- Complaints on Twitter
- Crowdsourced outage reporting
- Direct customer feedback

Valuable signals worth not ignoring! Could we integrate them into our day-to-day reliability management?



Know Your Tools

This talk isn't to say SLIs or SLOs are bad — or good. **The SLOs ecosystem is only a tool.** The only question for a tool is if it fits the need.

You should understand your needs. They might be answering questions with data, or organizational design, or many more!

This talk is about understanding when your tools apply for answering questions, and building new ones if you need them**.**

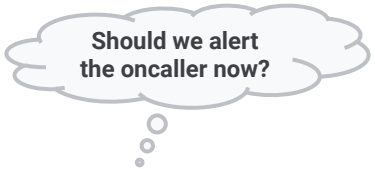
Operationalization

*"**...defines a fuzzy concept** so as **to make it** clearly distinguishable, **measurable**, and understandable by empirical observation..."*
(Wikipedia, May 2022)

Reliability Measurement Models

...operationalization using three "simple" steps

1. **Identify your key reliability questions**
    - Some are generic (e.g. need to alert), many aren't
    - Be precise, and think of cost of consequent action

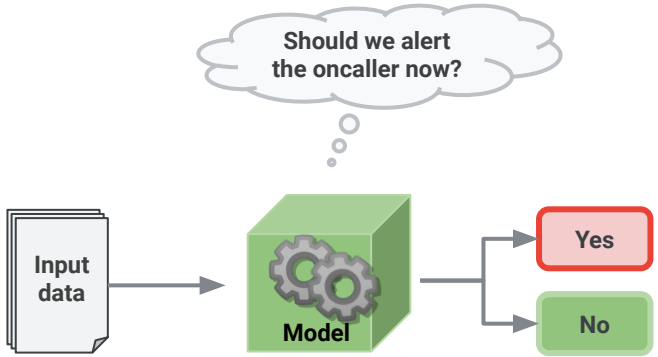


Reliability Measurement Models

...operationalization using three "simple" steps

1. **Identify your key reliability questions**
   - Some are generic (e.g. need to alert), many aren't
   - Be precise, and think of cost of consequent action

2. **Build a model for each question**
   - This is creative, and hard work
   - Consider how hard it is to agree on a model to answer the question "*given this data, should we alert someone?*"
   - ML techniques are tempting, but beware their caveats



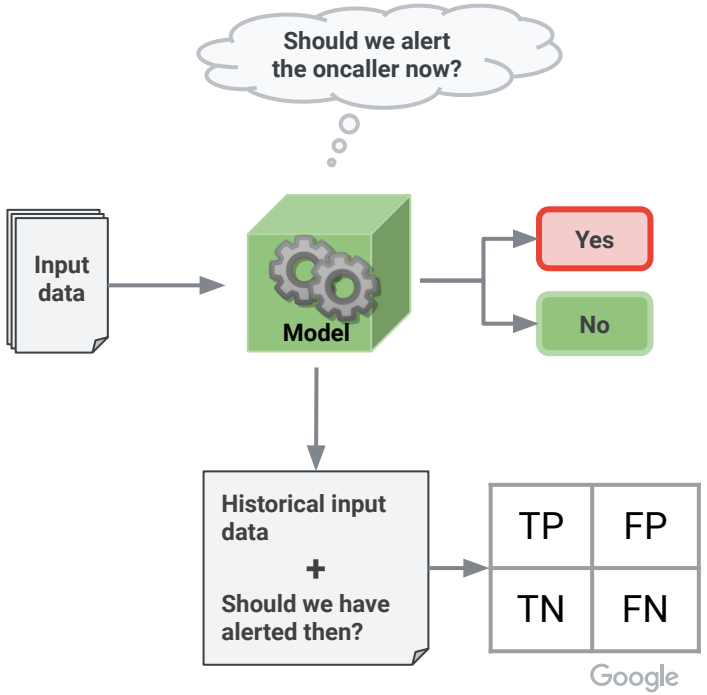
Reliability Measurement Models

...operationalization using three "simple" steps

1. **Identify your key reliability questions**
   - Some are generic (e.g. need to alert), many aren't
   - Be precise, and think of cost of consequent action

2. **Build a model for each question**
   - This is creative, and hard work
   - Consider how hard it is to agree on a model to answer the question "*given this data, should we alert someone?*"
   - ML techniques are tempting, but beware their caveats

3. **Backtest your models against historical data**
   - For boolean questions, you can get a confusion matrix
   - Identify model shortcomings, and iterate



Good news: We're doing it already!

- **"SLO alerting" is an example of building a fresh model**
  - Input is SLI data, output is a boolean answer
  - Frequent topic of articles and discussions
  - Alerting decision is made frequently

Good news: We're doing it already!

- **"SLO alerting" is an example of building a fresh model**
  - Input is SLI data, output is a boolean answer
  - Frequent topic of articles and discussions
  - Alerting decision is made frequently

- **Models for identifying unusual behaviors, such as:**
  - Anomaly detection in monitoring solutions
  - SRECon'21 talk "Beyond Goldilocks Reliability"
  - But beware: "*unusual*" is not automatically "*bad*"!
  - Cost of being wrong drives accuracy requirements

Good news: We're doing it already!

- **"SLO alerting" is an example of building a fresh model**
  - Input is SLI data, output is a boolean answer
  - Frequent topic of articles and discussions
  - Alerting decision is made frequently

- **Models for identifying unusual behaviors, such as:**
  - Anomaly detection in monitoring solutions
  - SRECon'21 talk "Beyond Goldilocks Reliability"
  - But beware: "*unusual*" is not automatically "*bad*"!
  - Cost of being wrong drives accuracy requirements

- **However, models for high-level decision are hard**
  - Typically very infrequent decisions
  - Not always clear what should've been done, even in hindsight

Conclusion

What got you here won't get you there!

SLI/SLO model is a helpful hammer, but not everything is a nail

- ➔ Understand what questions you need answered!
- ➔ Match your tool to that, don't start with a tool

Conclusion

What got you here won't get you there!

SLI/SLO model is a helpful hammer, but not everything is a nail

- ➔ Understand what questions you need answered!
- ➔ Match your tool to that, don't start with a tool

Build models, and backtest them (...and publish them?)

- ➔ Start with just three reliability questions
- ➔ Backtesting is sometimes hard, "*what should we have done?*" not always accurate or available
- ➔ Think of the cost of the answer being wrong, be ready

Conclusion

What got you here won't get you there!

SLI/SLO model is a helpful hammer, but not everything is a nail

- ➔ Understand what questions you need answered!
- ➔ Match your tool to that, don't start with a tool

Build models, and backtest them (...and publish them?)

- ➔ Start with just three reliability questions
- ➔ Backtesting is sometimes hard, "*what should we have done?*" not always accurate or available
- ➔ Think of the cost of the answer being wrong, be ready

Include new or external data in your reliability day-to-day practice

- ➔ Complaints on Twitter as a regularly measured quantity? :-)
- ➔ Ideally: used as input to your regularly exercised models

Conclusion

What got you here won't get you there!

SLI/SLO model is a helpful hammer, but not everything is a nail

- ➔ Understand what questions you need answered!
- ➔ Match your tool to that, don't start with a tool

Build models, and backtest them (...and publish them?)

- ➔ Start with just three reliability questions
- ➔ Backtesting is sometimes hard, "*what should we have done?*" not always accurate or available
- ➔ Think of the cost of the answer being wrong, be ready

Include new or external data in your reliability day-to-day practice

- ➔ Complaints on Twitter as a regularly measured quantity? :-)
- ➔ Ideally: used as input to your regularly exercised models

See also:

- Incident Metrics In SRE (O'Reilly, 2021)
- The VOID Report (Verica, 2021)
- ML for Operations (USENIX ;login:, 2020)
- How to Measure Anything (Wiley, 2020)

Thanks to: Ben Appleton, Kristina Bennett, Brent Bryan, Brendan Gleason, Paul Holden, Jennifer Mace, Jake McGuire, Niall Richard Murphy, Courtney Nash, Alex Rodriguez, Dylan Vener, Salim Virji

For their review and thoughts on this (or preceding) material

Illustrations by: Allyssa Jill Olivan (@kleinebean)
allyssajillolivan.myportfolio.com