Not all minutes are equal

The secret behind SLO adoption failure

Troy Koss + Michael Goins
March 2023

Who are we?

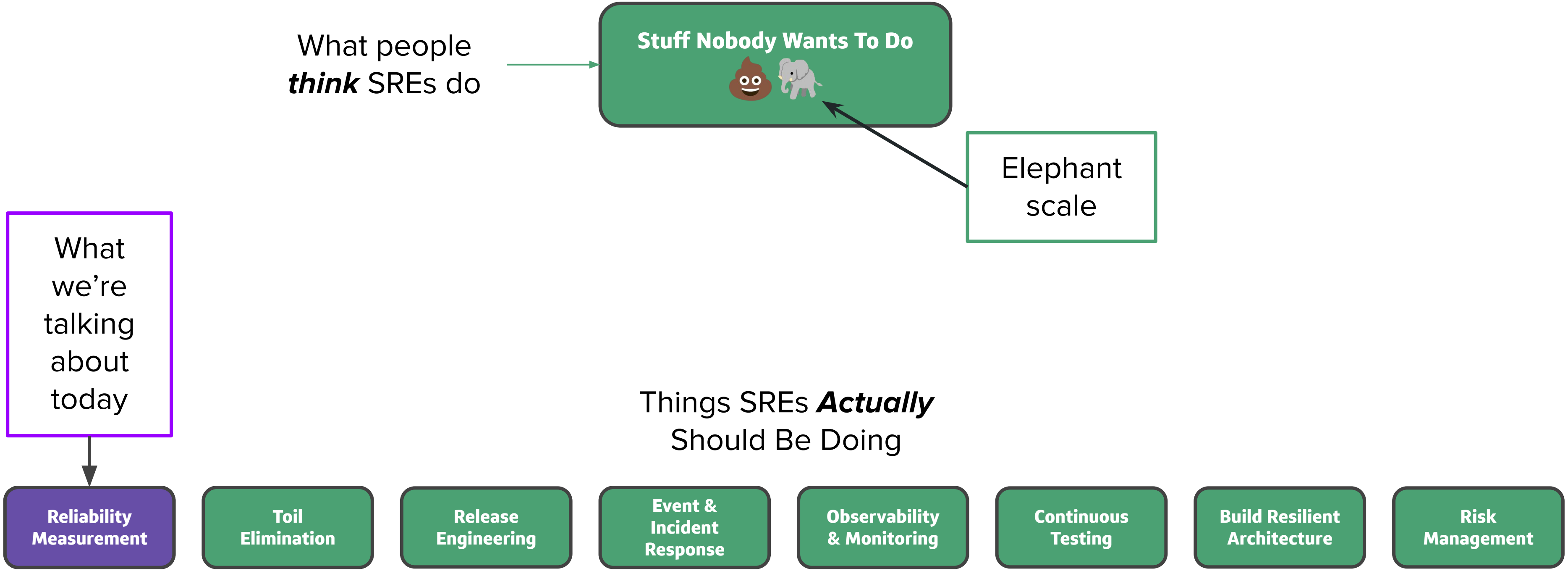
Michael Goins - A problem solver.

Example: My Hands hurt => Used vim => Problem solved.

Troy Koss - A problem creator.

Example: Had a simple service => Used Kubernetes => Problem created.

What SRE’s Do



What people *think* SREs do

Stuff Nobody Wants To Do 💩🐘

Elephant scale

What we’re talking about today

Things SREs ***Actually*** Should Be Doing

Reliability Measurement

Toil Elimination

Release Engineering

Event & Incident Response

Observability & Monitoring

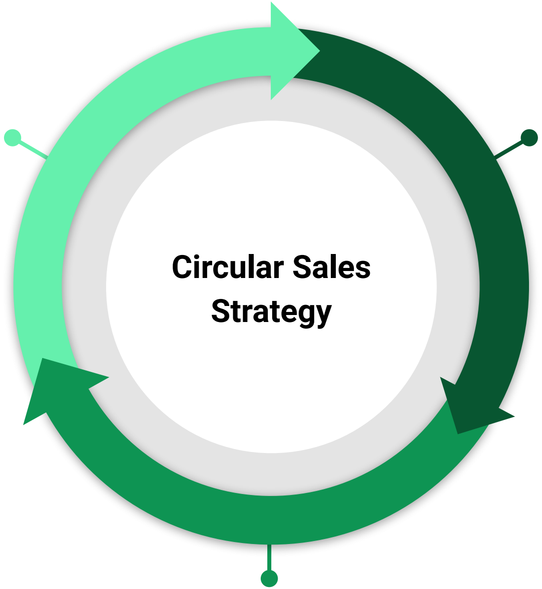
Continuous Testing

Build Resilient Architecture

Risk Management

Just because it’s published doesn’t mean it’s right

Metrics don’t self Justify



SLOs are Awesome

I saw it in a conference / book / article

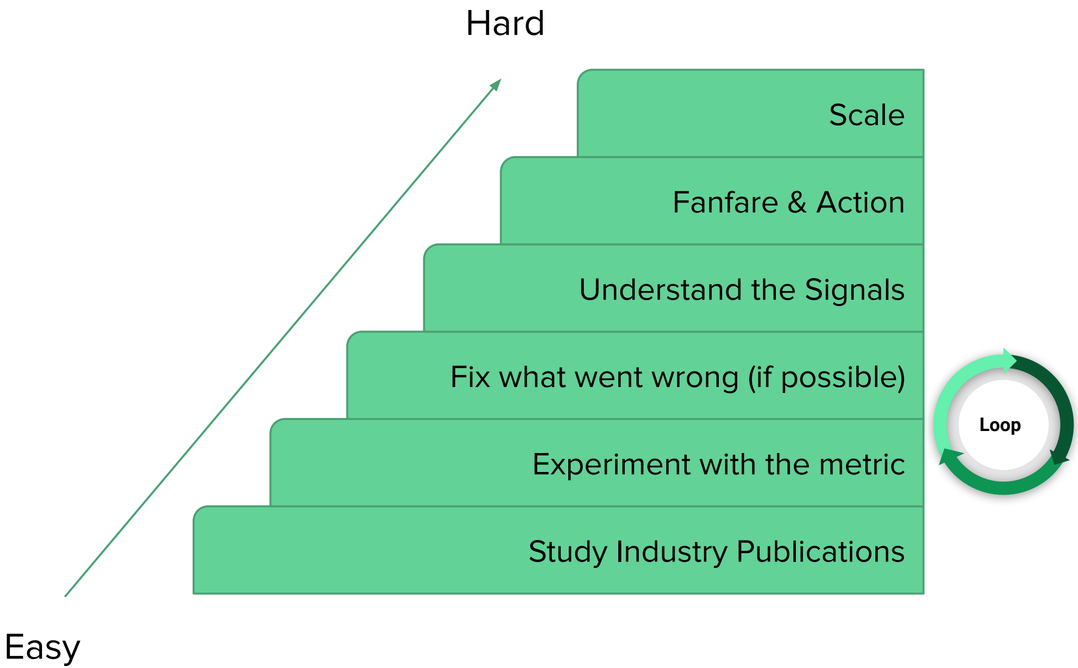
Used an SLO

Set up SLO on a system with much fanfare

The results are Awesome

Because…. SLOs verify that SLOs are great

An actual strategy



Hard

Scale

Fanfare & Action

Understand the Signals

Fix what went wrong (if possible)

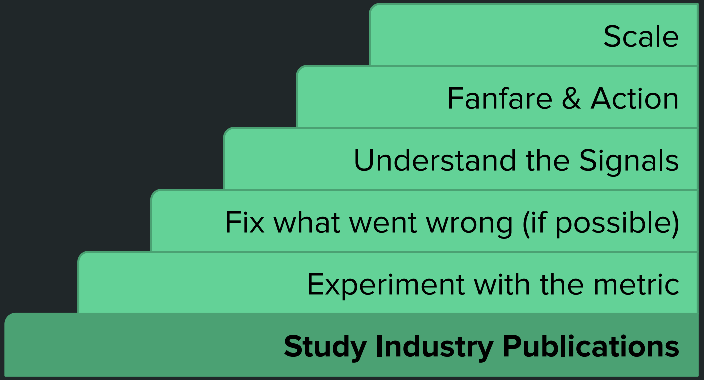
Experiment with the metric

Study Industry Publications

Loop

Easy

Study Industry Publications



Scale

Fanfare & Action

Understand the Signals

Fix what went wrong (if possible)

Experiment with the metric

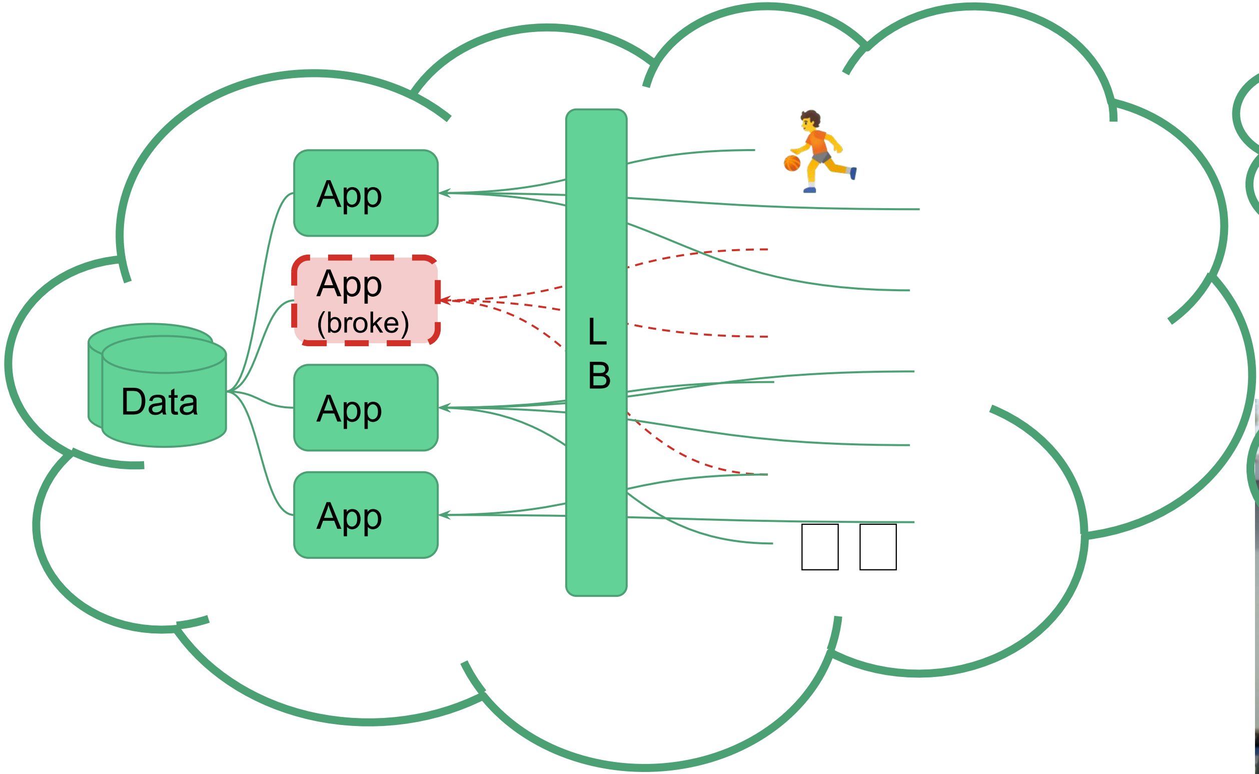
Study Industry Publications

SLOs in Marketing

Error Budget

Obj	Year	Month	Day
90%	36.5d	3d	2.4h
99%	3.65d	7.2h	14.4m
99.9%	8.76h	43.2m	1.44m
99.99%	52.56m	4.32m	8.64s
99.999%	5.256m	25.92s	0.86s

Deep Thought: ...Uptime?



9 / 12 requests = **?**
minutes

(Reenactment)

SLOs in Marketing

$$\overbrace{\underbrace{\boxed{\frac{Good\ Events}{Valid\ Events} * 100}}_{}^{}\!\!{}^{\text{SLI}} + \boxed{0 < Objective < 100}}^{\text{SLO}} = \overbrace{\text{table}}^{\text{Error Budget}}$$

Error Budget

Obj	Year	Month	Day
90%	36.5d	3d	2.4h
99%	3.65d	7.2h	14.4m
99.9%	8.76h	43.2m	1.44m
99.99%	52.56m	4.32m	8.64s
99.999%	5.256m	25.92s	0.86s

~~SLA~~ These are for lawyers

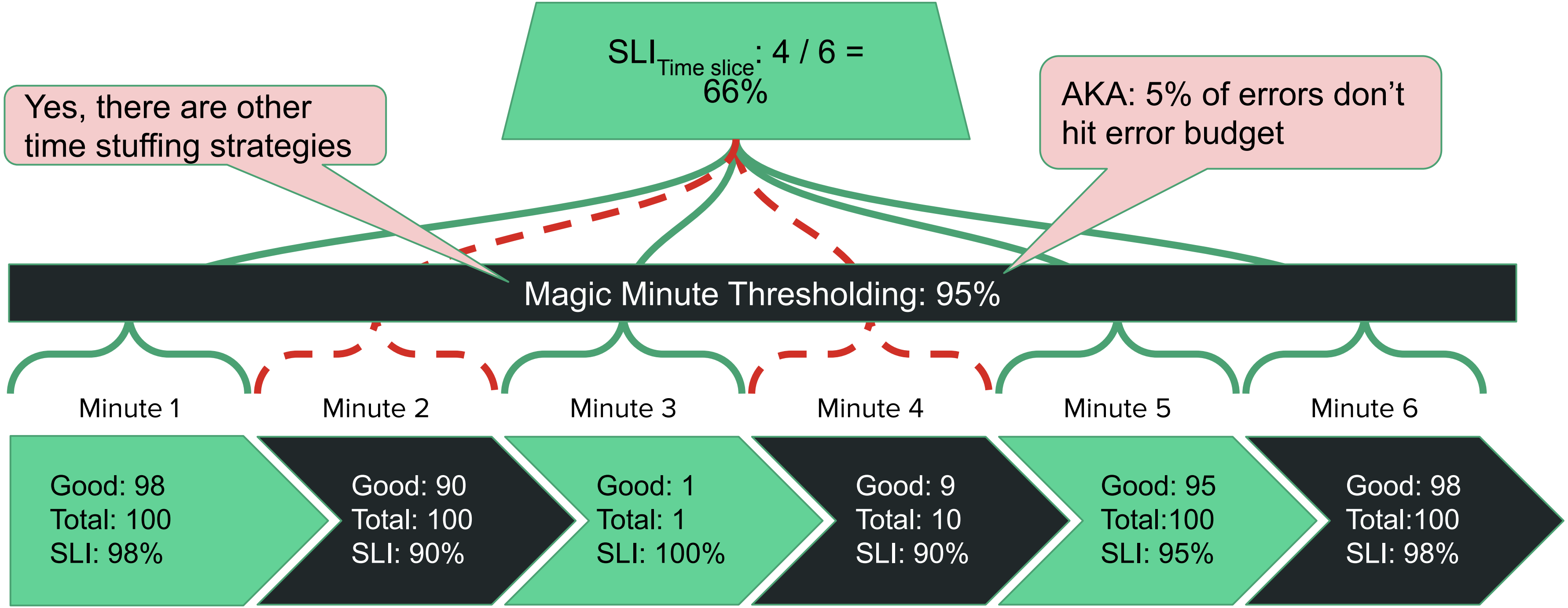
"Answers with inconsistent units are just plain crazy - they could *never* be right."
 - Duke University
https://webhome.phy.duke.edu/~rgb/Class/intro_math_review/intro_math_review/node21.html

Transitive Property: SREs = SLOs + EB
SLOs + EB = Crazy

SREs = Crazy

Stuffing Events in to Time (Time Slicing)

99.95 SLO for Availability (non 5XXs) at application



$\text{SLI}_{\text{Time slice}}$: 4 / 6 = 66%

Yes, there are other time stuffing strategies

AKA: 5% of errors don’t hit error budget

Magic Minute Thresholding: 95%

Minute 1

Good: 98
Total: 100
SLI: 98%

Minute 2

Good: 90
Total: 100
SLI: 90%

Minute 3

Good: 1
Total: 1
SLI: 100%

Minute 4

Good: 9
Total: 10
SLI: 90%

Minute 5

Good: 95
Total:100
SLI: 95%

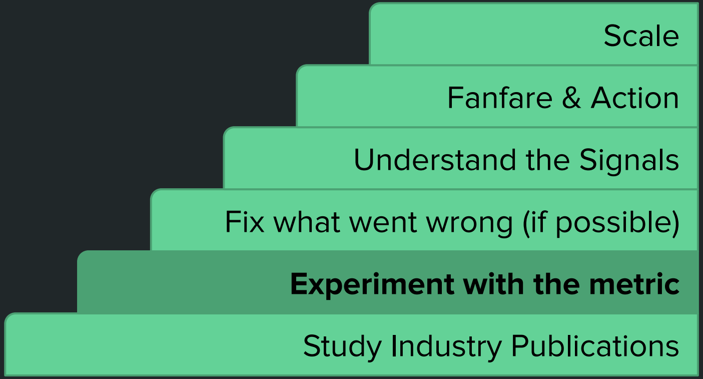
Minute 6

Good: 98
Total:100
SLI: 98%

Problem

Created!

Experiment with the metric



Scale

Fanfare & Action

Understand the Signals

Fix what went wrong (if possible)

Experiment with the metric

Study Industry Publications

Experiment

Question:

Is Error Budget actually in “Events” or are “Events” really in time?

Assertion:

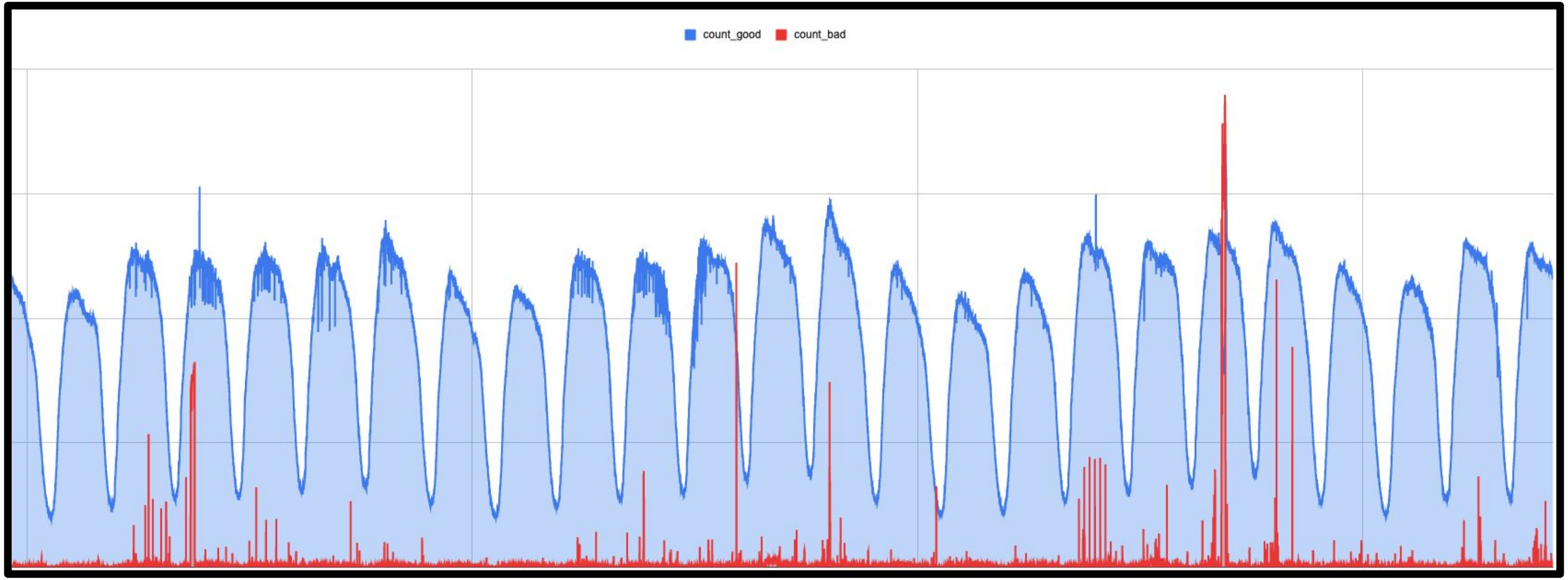
SLOs should *correlate* with what is known currently about system health

Currently Known To Be True (~~always~~ *most of the time*):

- Incidents
- Changes
- Personal Impressions

Event Ratio (500s & non-500s)

GOOD



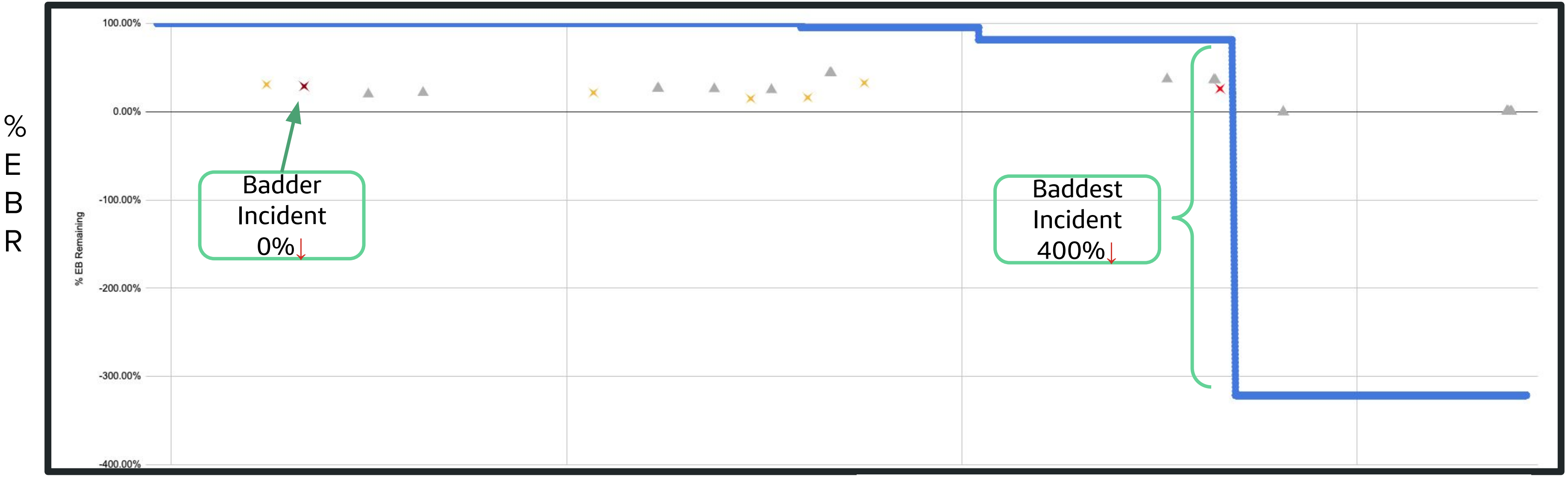
BAD

Time

Time Slice: “L Graph”

99.95 SLO with 95% “Threshold” for good/bad minute

**(1 - Threshold): % of Bad Traffic to ignore each minute (doesn’t hit EB)



%
E
B
R

Badder Incident 0%↓

Baddest Incident 400%↓

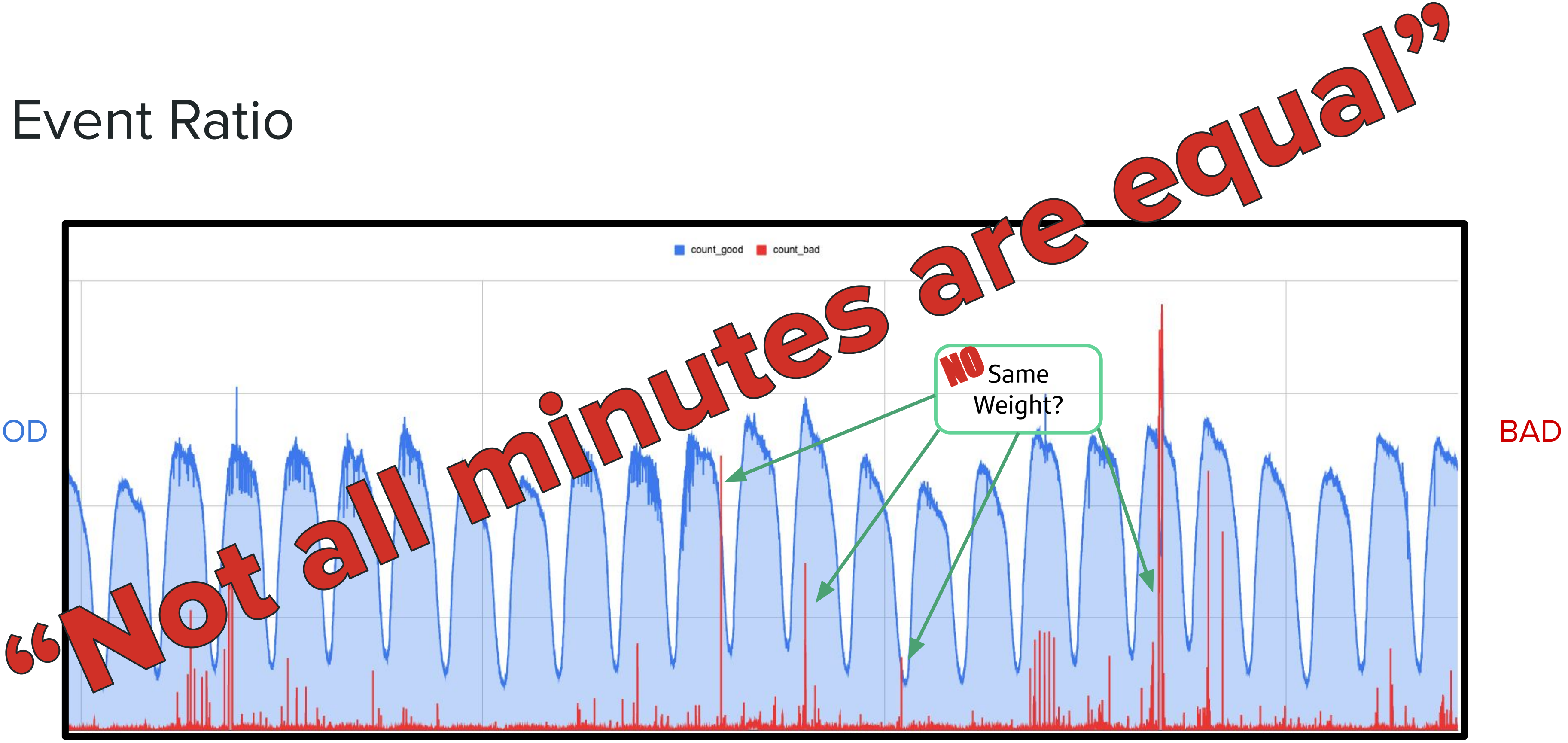
Time

Incidents

● EBR ✕ Baddest ✕ Badder ✕ Bad ▲ Change

Things people believe currently

Event Ratio



Time

Fix what went wrong (if possible)

Scale

Fanfare & Action

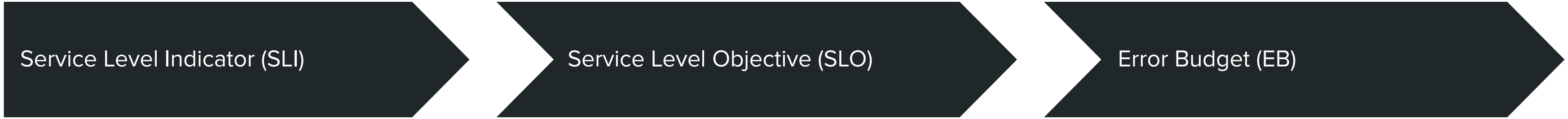
Understand the Signals

Fix what went wrong (if possible)

Experiment with the metric

Study Industry Publications

Formulaic SLO definition



Service Level Indicator (SLI)

Definition:

Ratio of Good over Total through a period of time measured through observability tools

Formula:

SLI = Good / Total * 100

Example:

Good: 99,990 requests in 30 days

Total: 100,000 requests in 30 days

99.99% = 99,990 / 100,000 * 100

Service Level Objective (SLO)

Definition:

Expected system reliability from a resourcing perspective

Formula:

0% < Objective < 100%

Example:

Objective = 99.5%

Error Budget (EB)

Definition:

The amount of permissible impact on users

Formulas:

Error Budget = (1 - Objective) * Total

% EB Remaining = ((EB - Bad) / EB) * 100

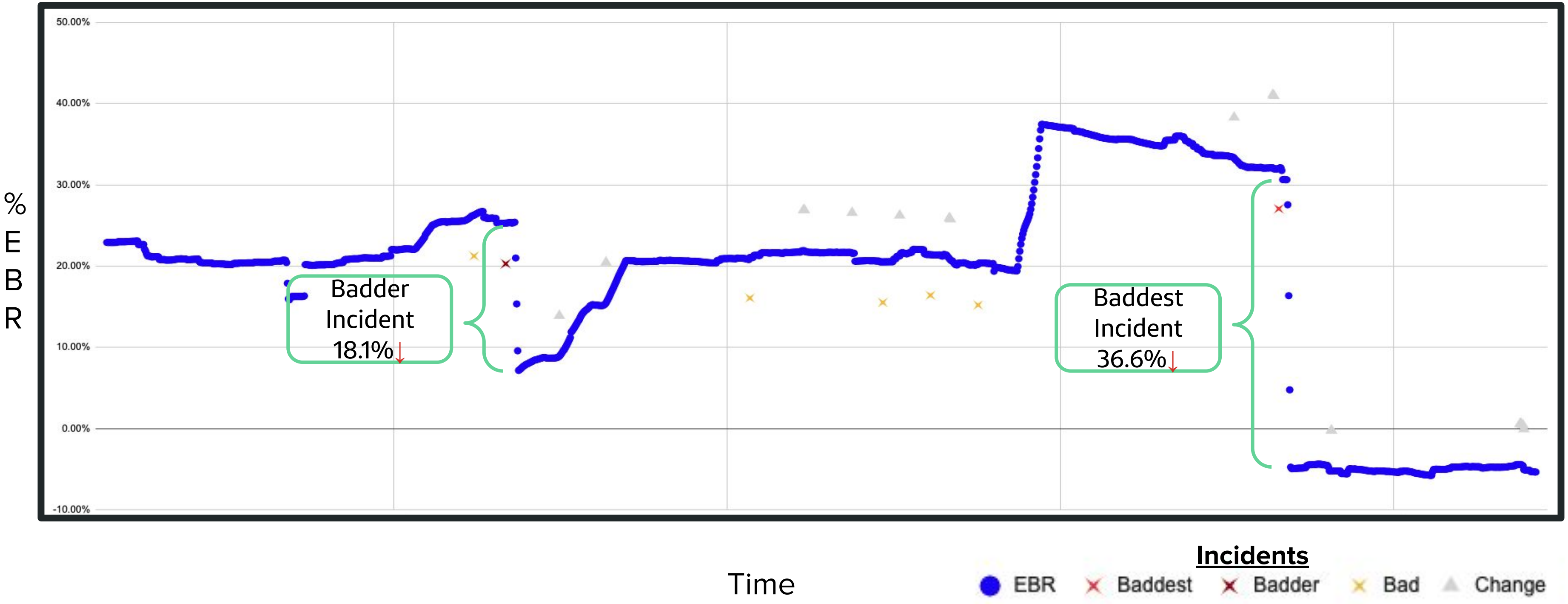
Examples:

500 Allowed Errors= (1 - .995) * 100,000

98% EBR = ((500 - 10 Bad) / 500) * 100

Event Based: “Money Chart💰💸🤑”

99.95 SLO



Badder Incident 18.1%↓

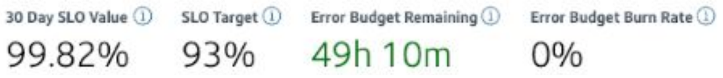
Baddest Incident 36.6%↓

Problem

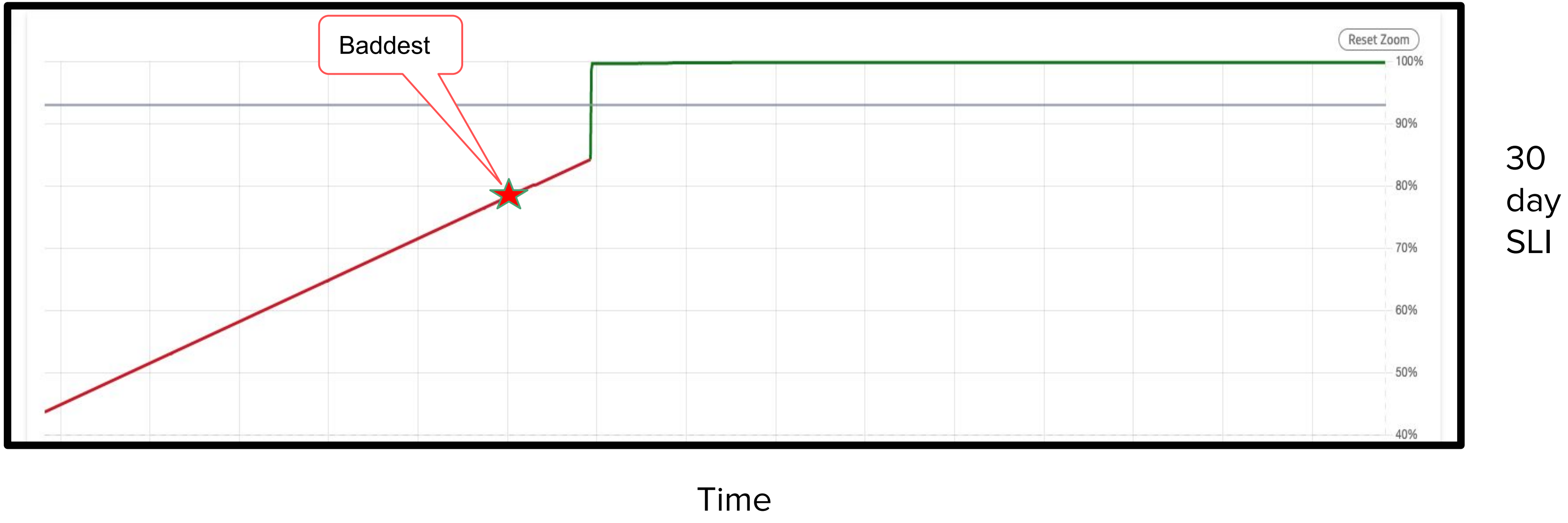
Solved!

Other reasons to test…

Coding mistakes: “Testing matters”



30 Day SLO Value	SLO Target	Error Budget Remaining	Error Budget Burn Rate
99.82%	93%	49h 10m	0%

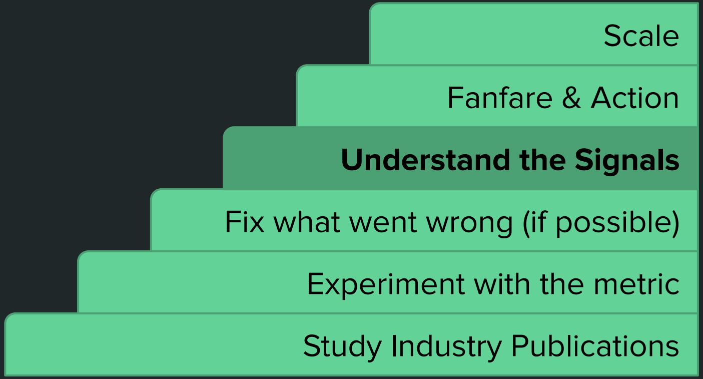


Another

Problem

Created!

Understand the Signals



Scale

Fanfare & Action

Understand the Signals

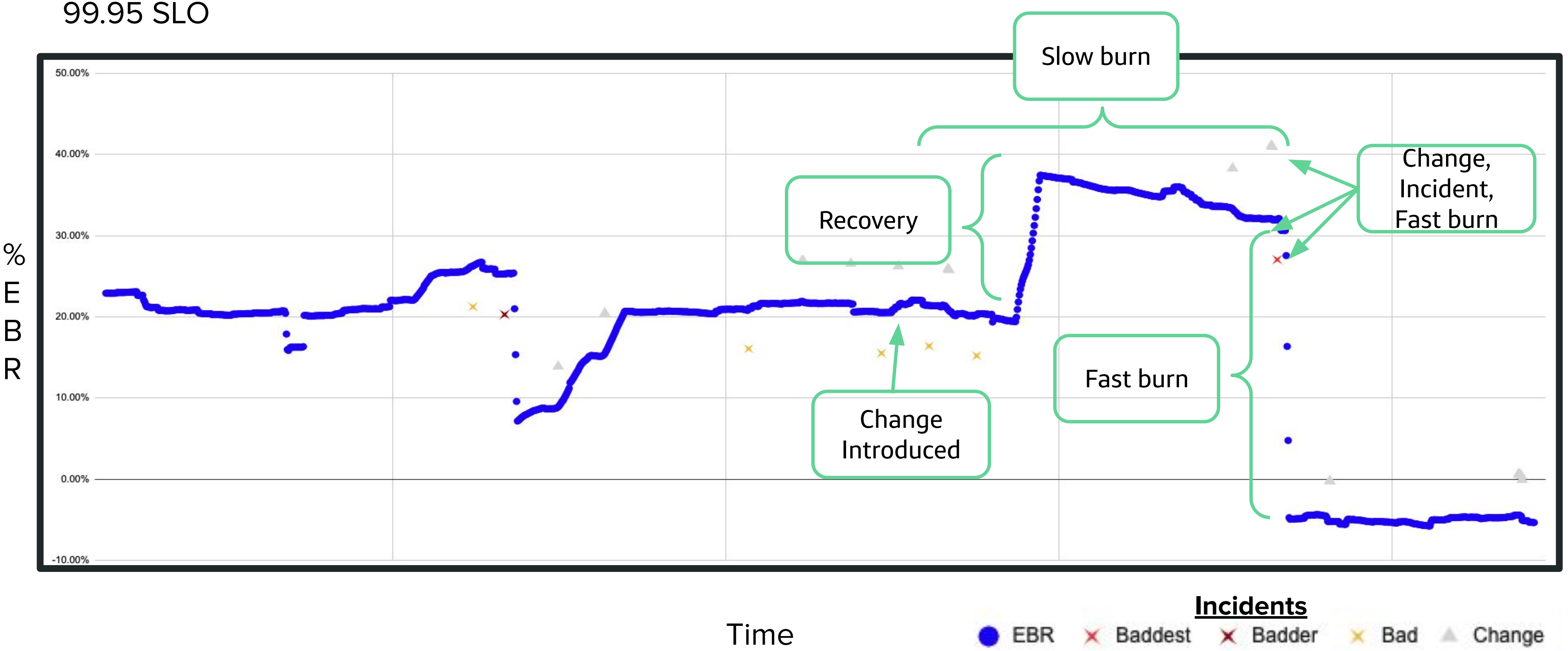
Fix what went wrong (if possible)

Experiment with the metric

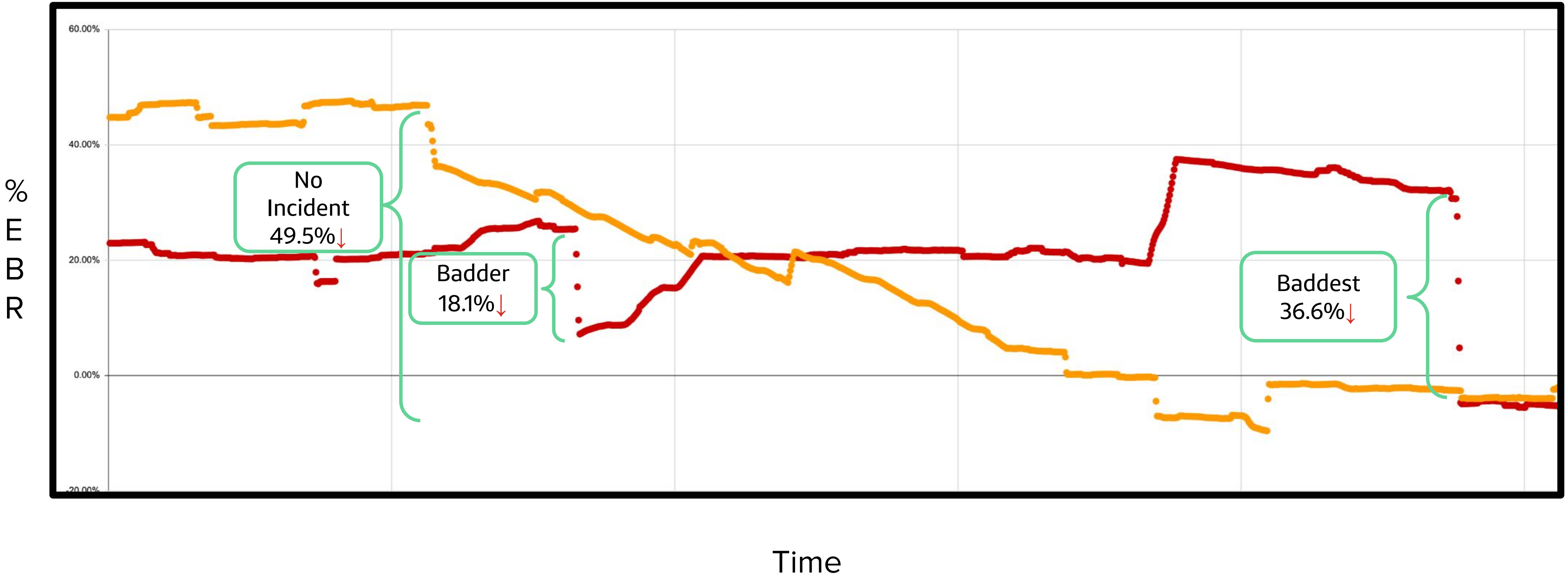
Study Industry Publications

Signals: Interpretation

99.95 SLO



Signals: “Feel the Bern”



No Incident 49.5%↓

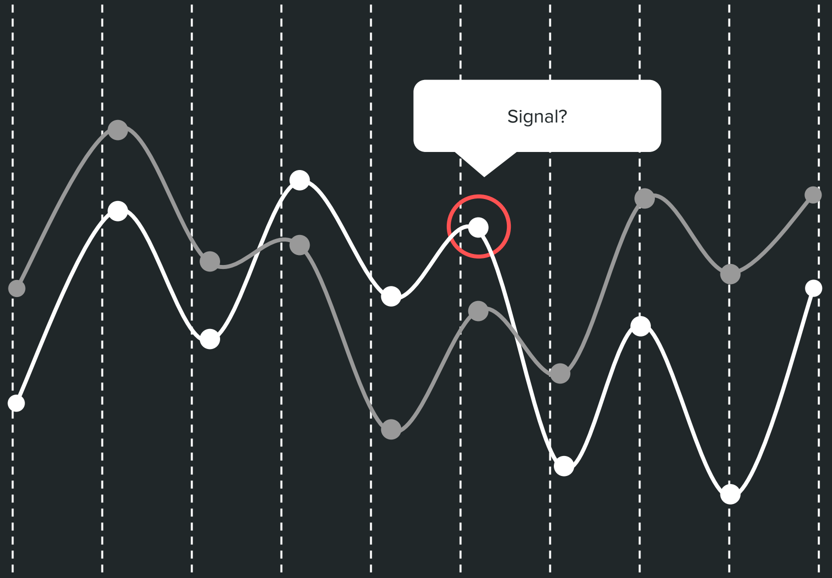
Badder 18.1%↓

Baddest 36.6%↓

Great the data is believable, now what?

Error Budget Signals

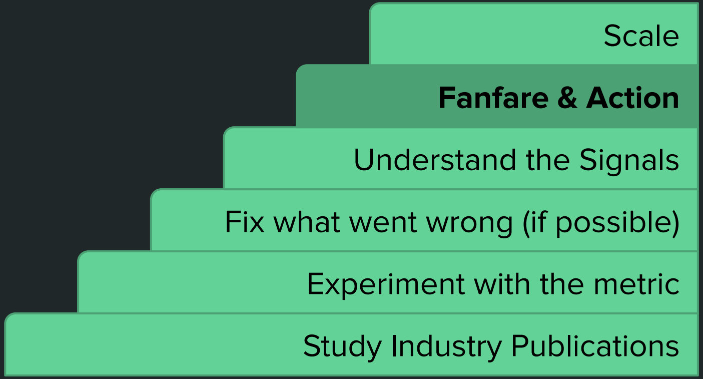
- Movement:
  - Slow Burns
  - Fast Burns
  - Recoveries
- Associations:
  - Changes
  - Incidents (*including missing incidents*)
  - Bug
  - Alerts
- Action:
  - 5 Ws, & H



Remember **M**ichael's Sheep "MAA"......
🐑🐑🐑🐑

Fanfare & Action

We are here
"And there was much rejoicing (yaaaaaaaay)"


Scale

Fanfare & Action

Understand the Signals

Fix what went wrong (if possible)

Experiment with the metric

Study Industry Publications

Bootstrapping Teams w/Defaults

You get a SLO!

And You get a SLO!

What should my objective be?

98.5?

99.5?

99.9?

99.99999?

64.5?

Another, Another

Problem

Created!

Default SLOs

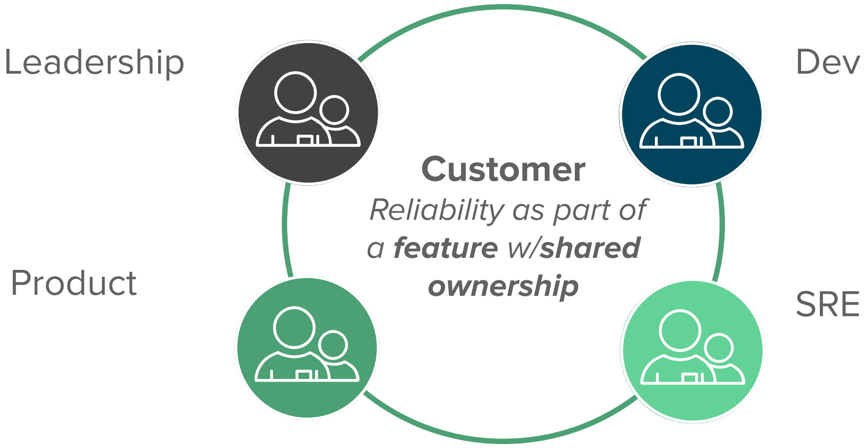
$$SLO_{default} = \left(\frac{bad}{(EB\% - 1)Total} + 1 \right) * 100$$

Given a target %EBR calculate an Objective
(Useful for when people say “I don’t know what my objective should be”)

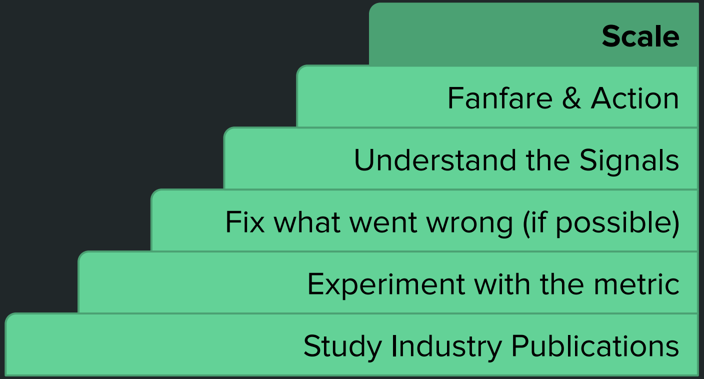
That’s cool, what do we do with it?

Error Budget Policy

- Who does what, when?
- Accountability
- Reinforcement
  - Team member changes
  - Re-org
- Enhancements to the SLO
  - Changes to SLI
  - Changes to Objective


Leadership
Dev
Customer
*Reliability as part of a **feature** w/**shared ownership***
Product
SRE

Scale



Scale

Fanfare & Action

Understand the Signals

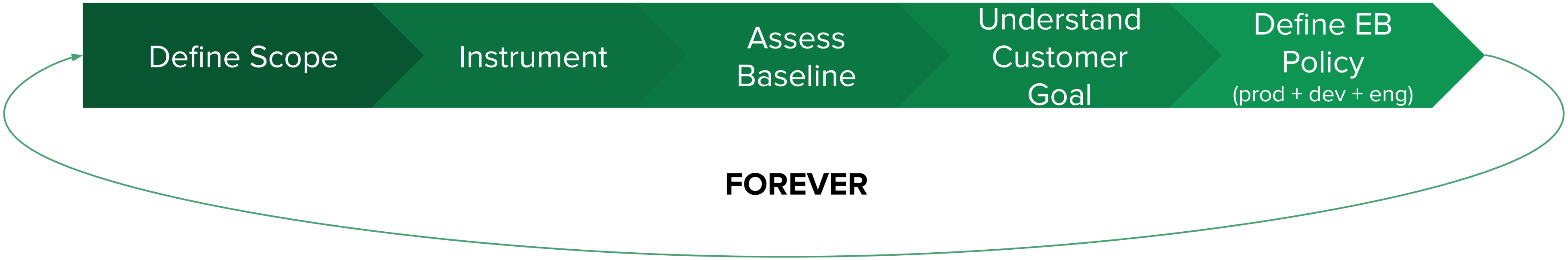
Fix what went wrong (if possible)

Experiment with the metric

Study Industry Publications

Easy as 1, 2, 3... 4 - And then recursive algorithms?

1. **Start**: SLIs Availability (5xx) & Latency (sec) to provide a baseline
2. **Inspect Different Layers**: Services, APIs, Edge, Client to equip SLIs
3. **Add edge cases**: SLIs for (ex. 400, 408, 409, client-side javascript errors, retry logic, etc)
4. **Map to customer interactions**: Authenticate/Log In, Complete a {{customer-action}}, etc.



Define Scope

Instrument

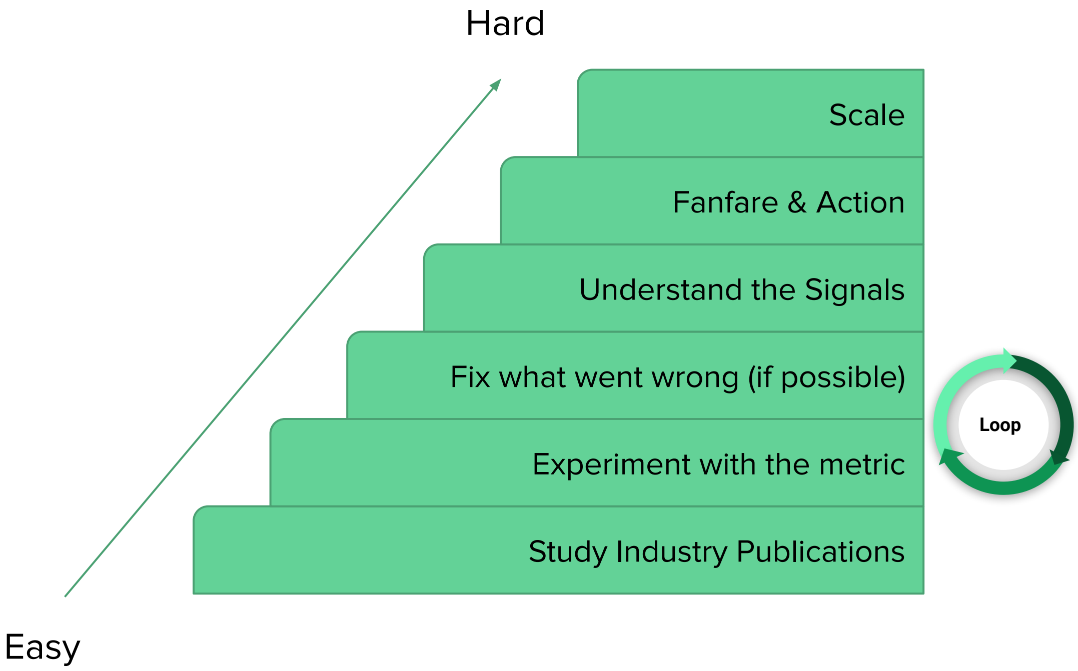
Assess Baseline

Understand Customer Goal

Define EB Policy (prod + dev + eng)

FOREVER

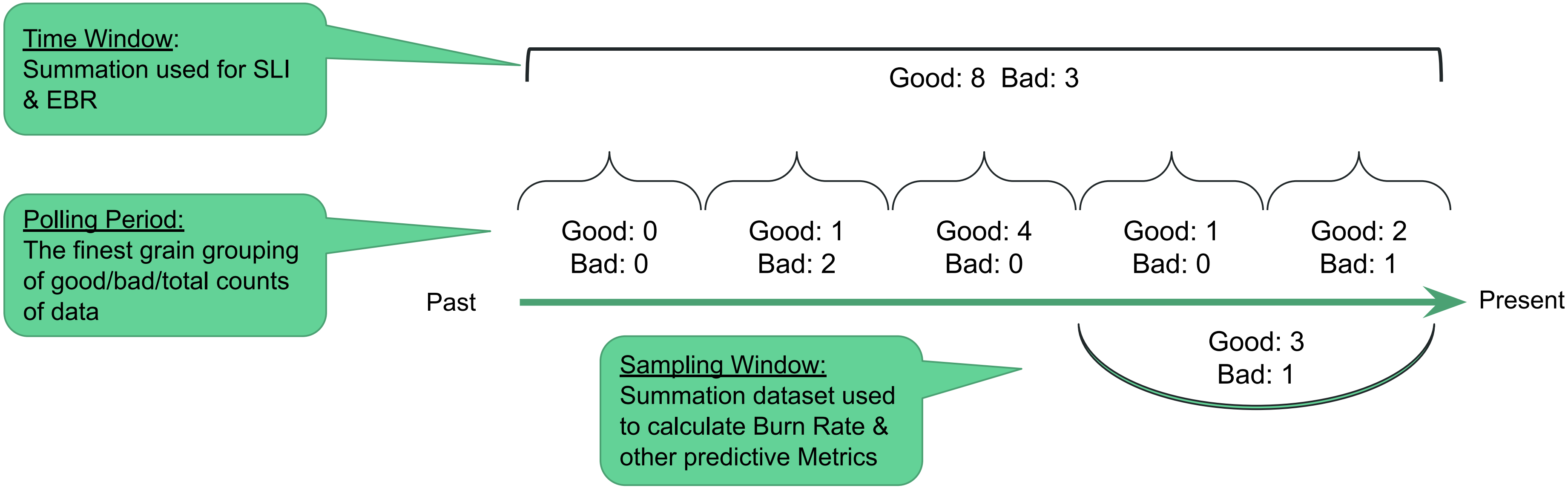
An actual strategy - Completed! 💥👏👍😜👾


Hard
Scale
Fanfare & Action
Understand the Signals
Fix what went wrong (if possible)
Loop
Experiment with the metric
Study Industry Publications
Easy

Appendix

A Vestigial Organ

Time Windows



Time Window: Summation used for SLI & EBR

Polling Period: The finest grain grouping of good/bad/total counts of data

Sampling Window: Summation dataset used to calculate Burn Rate & other predictive Metrics

SLI & Error Budget

$$SLI_{TW} = \frac{Good\ Events_{TW}}{Total\ Events_{TW}} * 100$$

$$EB\ Current_{TW} = EB\ Max\ -\ (BadEvents_{TW})$$

$$EB\ Max_{TW} = (1\ -\ Objective) * (Total\ Events_{TW})$$

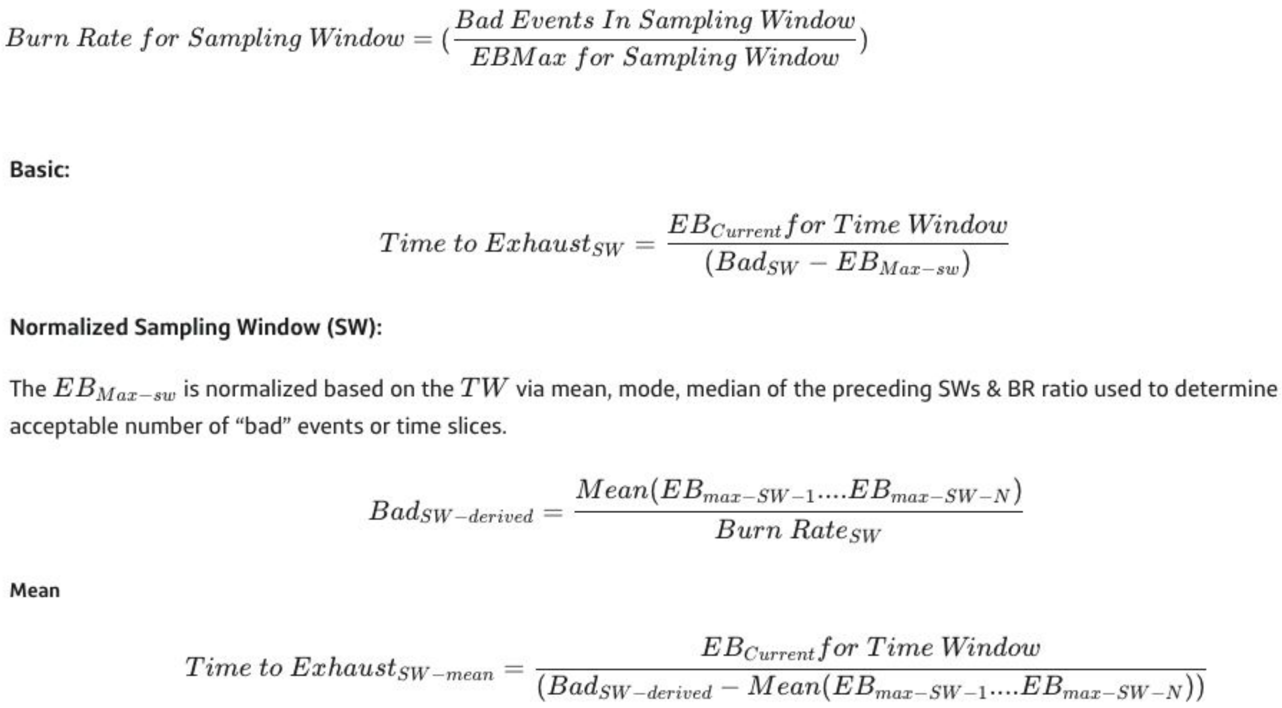
$$\%EB\ Remaining\ (EBR\%) = \frac{EB_{Current-TW}}{EB_{Max-TW}} * 100$$

$$SLO_{default} = (\frac{bad}{(EB\% - 1)Total} + 1) * 100$$

Given a target %EBR calculate an Objective (Useful for when people say “I don’t know what my objective should be”)

Basic Predictive Math

(Remember: Published != “right”)



$$Burn\ Rate\ for\ Sampling\ Window = (\frac{Bad\ Events\ In\ Sampling\ Window}{EBMax\ for\ Sampling\ Window})$$

Basic:

$$Time\ to\ Exhaust_{SW} = \frac{EB_{Current} for\ Time\ Window}{(Bad_{SW} - EB_{Max-sw})}$$

Normalized Sampling Window (SW):

The EB_{Max-sw} is normalized based on the TW via mean, mode, median of the preceding SWs & BR ratio used to determine acceptable number of “bad” events or time slices.

$$Bad_{SW-derived} = \frac{Mean(EB_{max-SW-1}....EB_{max-SW-N})}{Burn\ Rate_{SW}}$$

Mean

$$Time\ to\ Exhaust_{SW-mean} = \frac{EB_{Current} for\ Time\ Window}{(Bad_{SW-derived} - Mean(EB_{max-SW-1}....EB_{max-SW-N}))}$$

Q & A

Quesadillas & Avocados