What is SRE?

Geoff White

Principle SRE Nexsys Systems LLC

May 1, 2023

What is SRE series…

- **SRE 101**
- All about SLIs, SLOs, Error Budgets and SLAs
- Incidents and Incident Management
- Trends in SRE (CRE, Chaos Engineering, Resilience Engineering)

The goal of this talk…

To allow you to have knowledgeable conversations with customers around SRE.

So What is…

SRE?

SRE can mean…

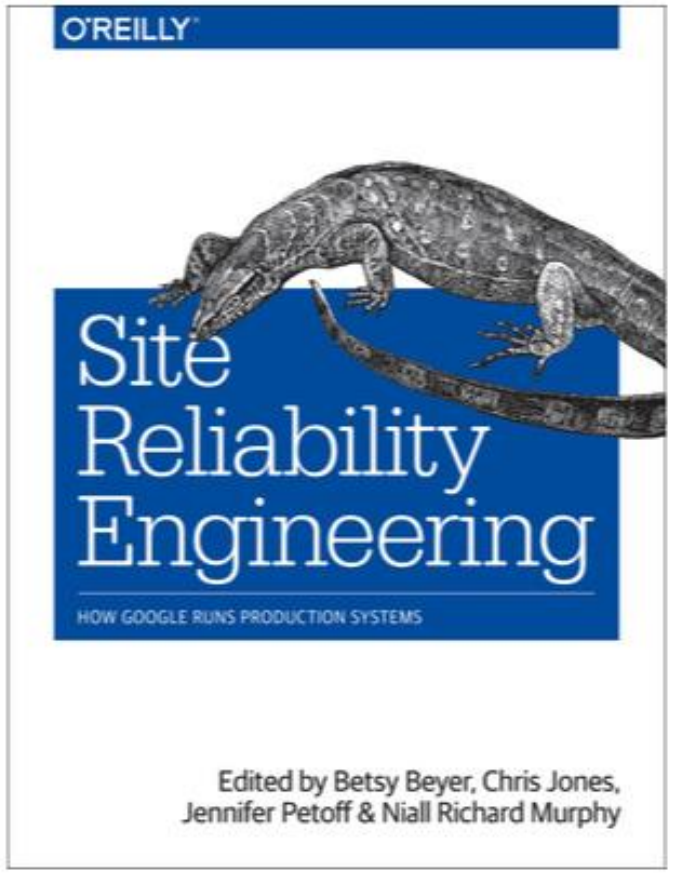
- SRE (the discipline)
- SRE (the role)

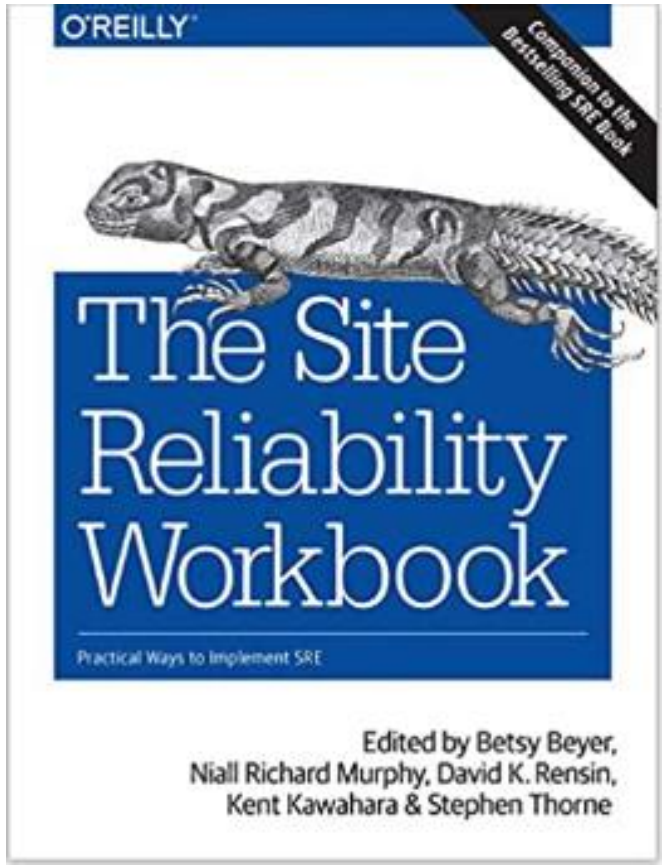
SRE (the discipline) can mean…

- Site Reliability Engineering
- Systems Reliability Engineering
- Systems Resiliency Engineering

SRE is "what happens when a software engineer is tasked with what used to be called operations." - Ben Treynor Sloss

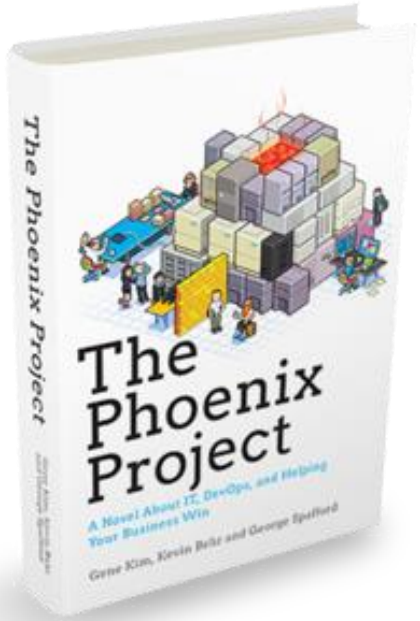
SRE Bibles

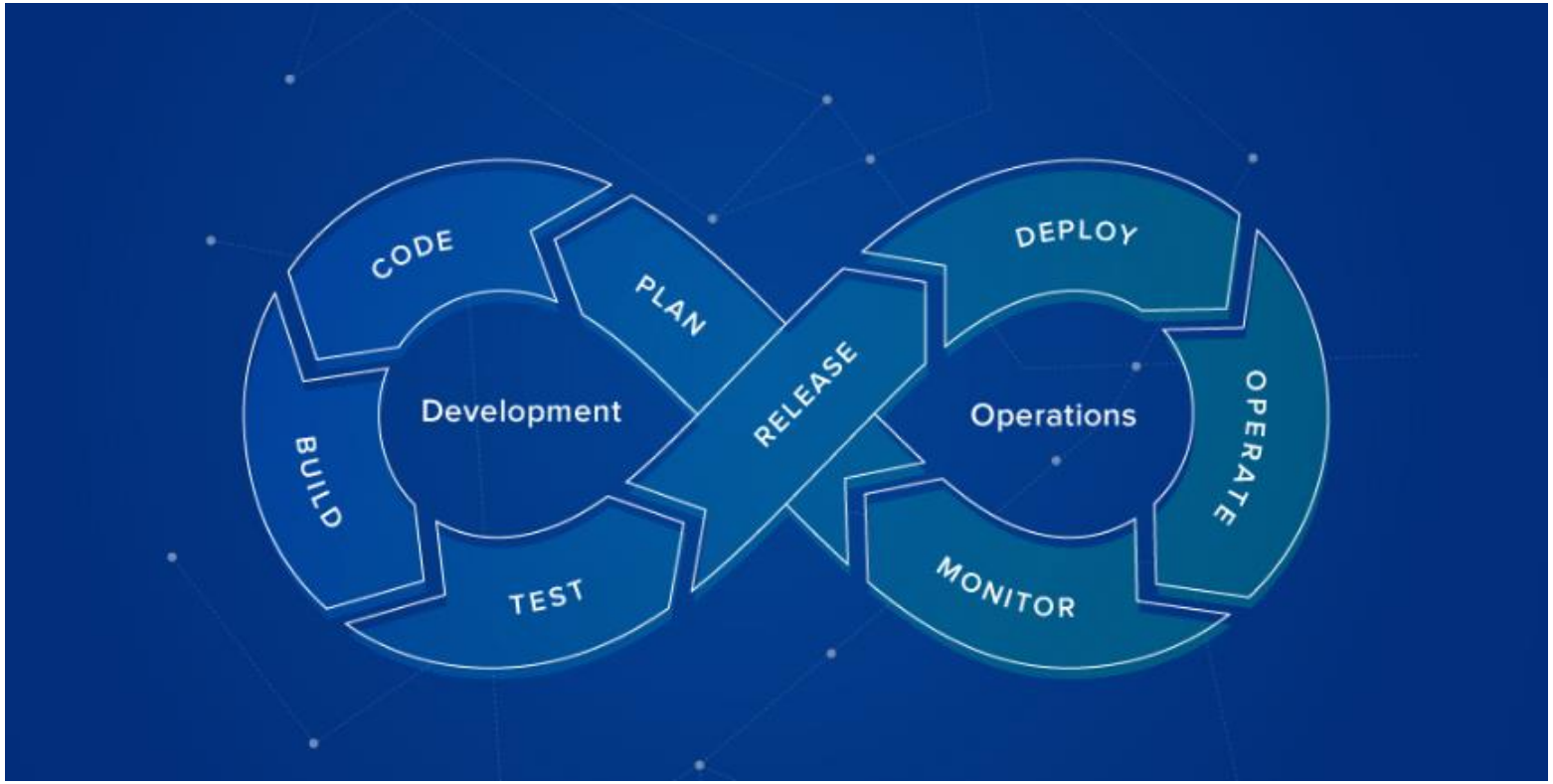

O'REILLY
Site Reliability Engineering
HOW GOOGLE RUNS PRODUCTION SYSTEMS
Edited by Betsy Beyer, Chris Jones, Jennifer Petoff & Niall Richard Murphy

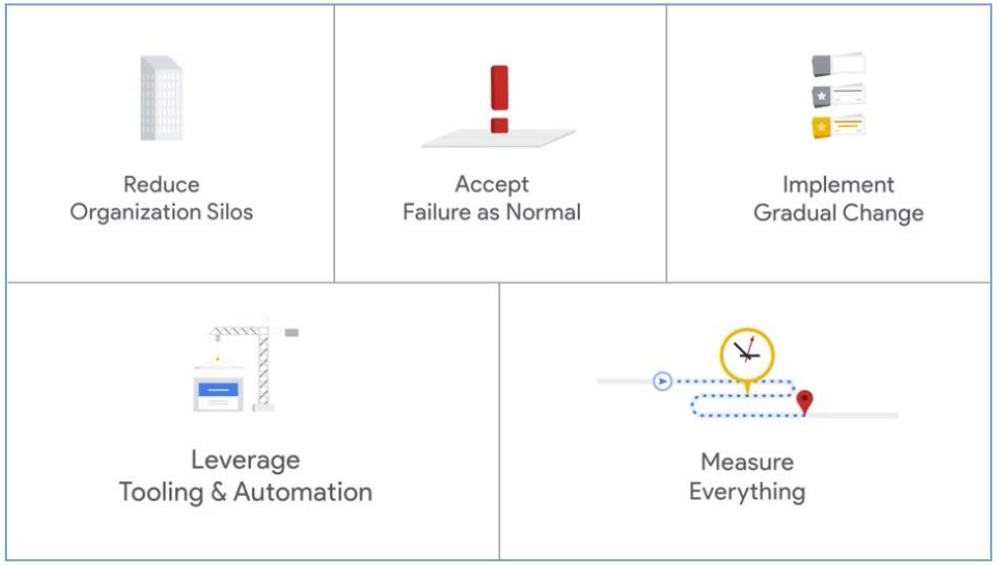

O'REILLY
Companion to the Bestselling SRE Book
The Site Reliability Workbook
Practical Ways to Implement SRE
Edited by Betsy Beyer, Niall Richard Murphy, David K. Rensin, Kent Kawahara & Stephen Thorne

DevOps principles

- Coined in 2009 by **Patrick Debois**
- Started to get traction by 2010
- Sys Admin != DevOps Engineer
- DevOps is a Philosophy not a Role (purist)
- The Phoenix Project…







class **SRE** implements **DevOps**

What makes a successful SRE (the role)?

- SRE's can and do write code.
- Generalists, with one or more specialties.
- Natural curiosity about how systems work.
- Learn from incidents so that the same incident doesn't happen again.

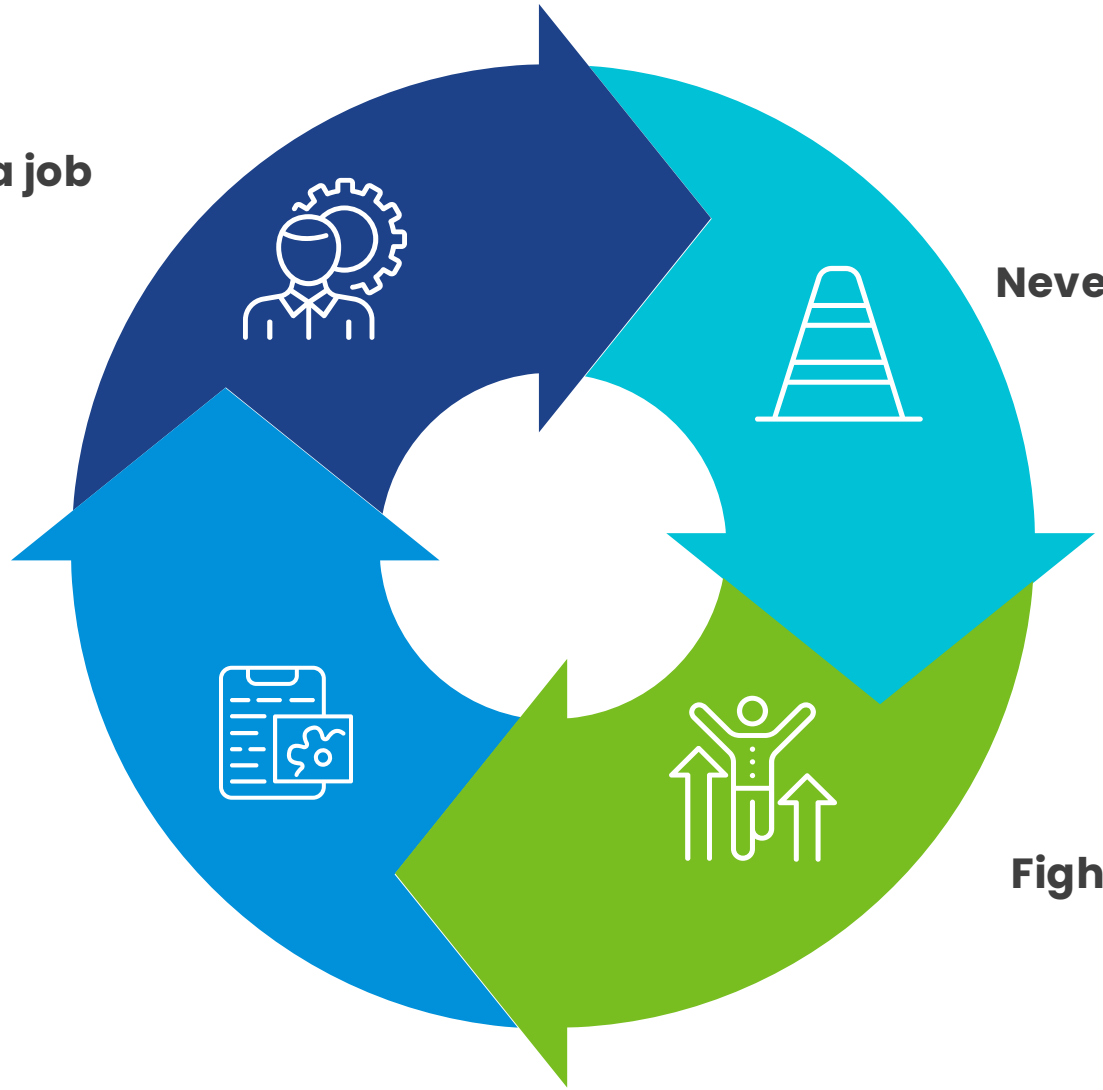
SRE Lore

http://bit.ly/3BVfWwa

What SREs do...

WRITE YOUR SUBTITLE HERE

Automate yourself out of a job



Never let a good incident go to waste

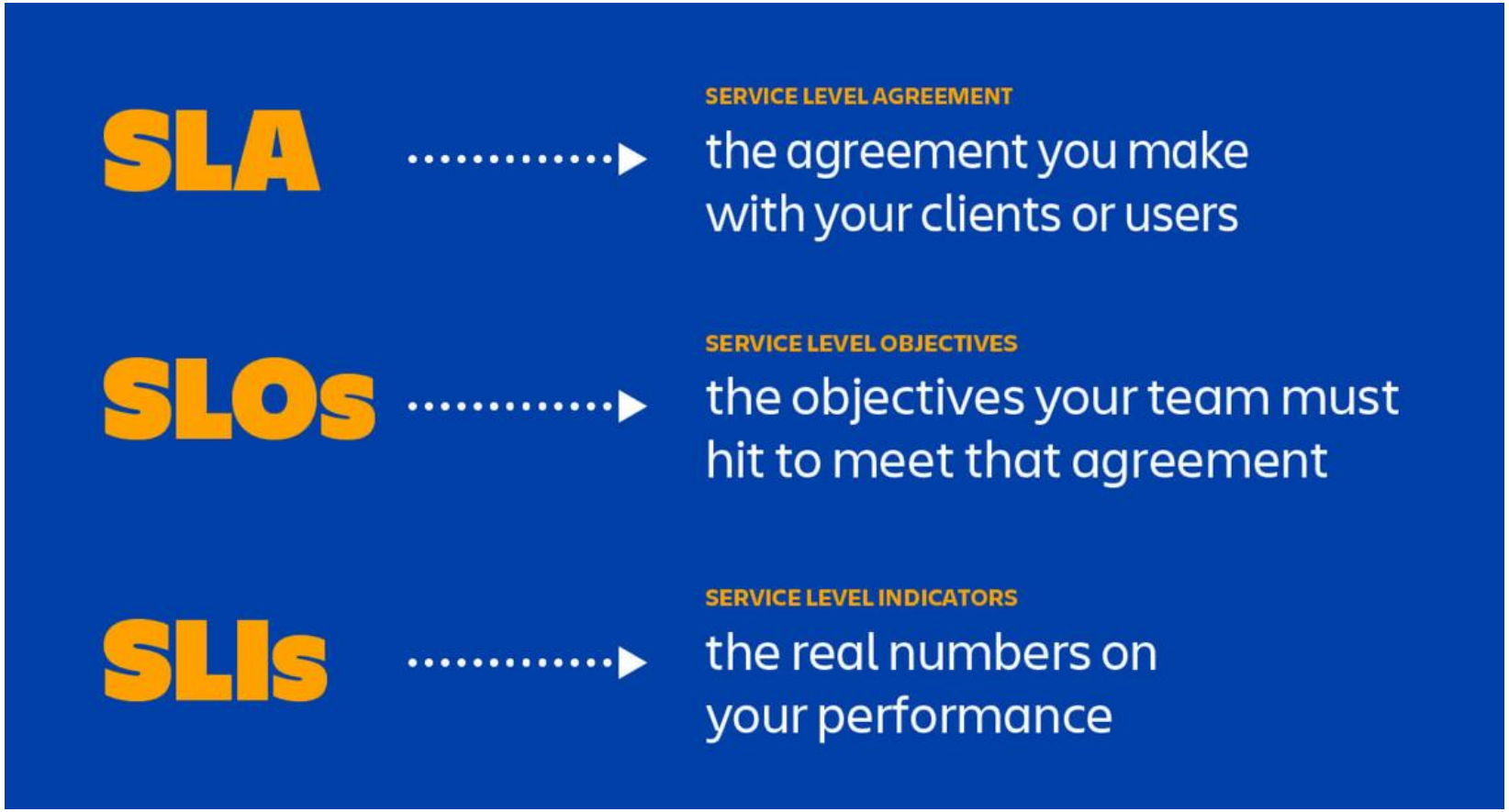
Measure Everything

Fight Burnout

What do SREs Do?...

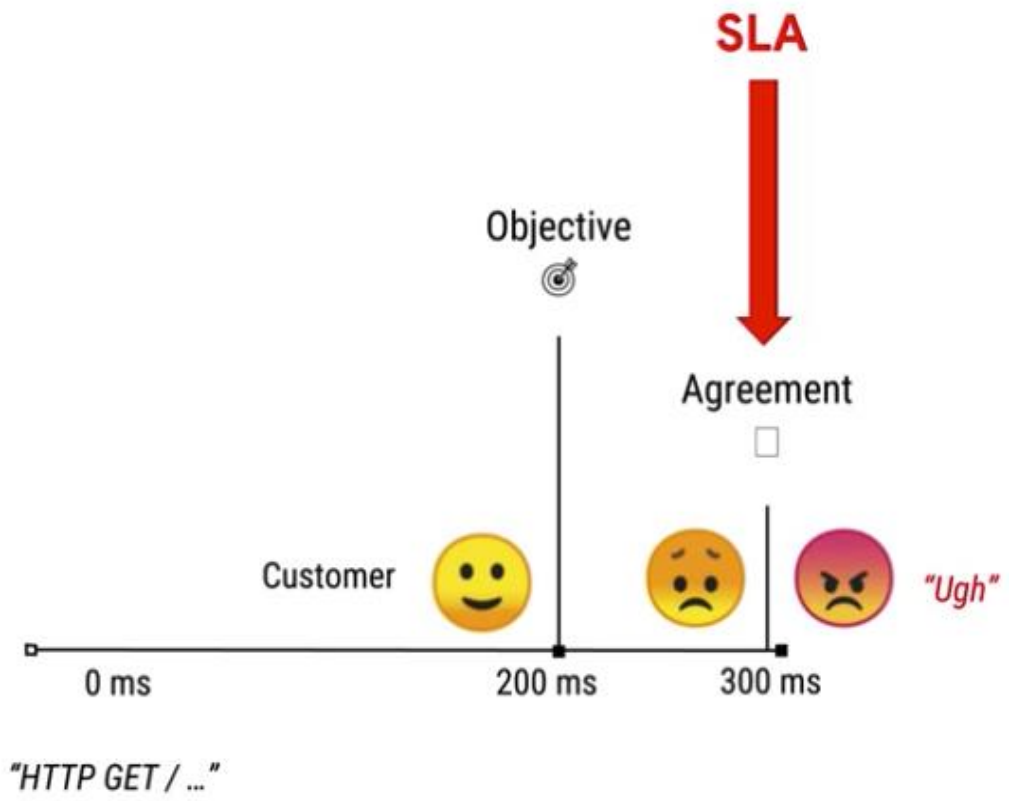
- Measure Everything (Observability)
- Automate themselves out of a Job…
- Never let a good Incident go to waste
- Fight Burnout

Common metrics SREs deal with…



SLA

- Customer Facing
- Established by contract
- Consequences if not met



Effective SLOs

- Have executive buy-in
- Not generally Customer Facing
- Are accurately measured.
- Indicator of Customer (un)Happiness
- Consequences if not met (error budgets)

SLI

Calculating the SLIs

Using the preceding metrics, we can calculate our current SLIs over the previous seven days, as shown in Table 2-2.

Table 2-2. Calculations for SLIs over the previous seven days

Availability	`sum(rate(http_requests_total{host="api", status!~"5.."}[7d]))`
	`/`
	`sum(rate(http_requests_total{host="api"}[7d])`

Latency	`histogram_quantile(0.9, rate(http_request_duration_seconds_bucket[7d]))` `histogram_quantile(0.99, rate(http_request_duration_seconds_bucket[7d]))`

Type of service	Type of SLI	Description
Request-driven	Availability	The proportion of requests that resulted in a successful response.
Request-driven	Latency	The proportion of requests that were faster than some threshold.
Request-driven	Quality	If the service degrades gracefully when overloaded or when backends are unavailable, you need to measure the proportion of responses that were served in an undegraded state. For example, if the User Data store is unavailable, the game is still playable but uses generic imagery.
Pipeline	Freshness	The proportion of the data that was updated more recently than some time threshold. Ideally this metric counts how many times a user accessed the data, so that it most accurately reflects the user experience.
Pipeline	Correctness	The proportion of records coming into the pipeline that resulted in the correct value coming out.

Error Budget

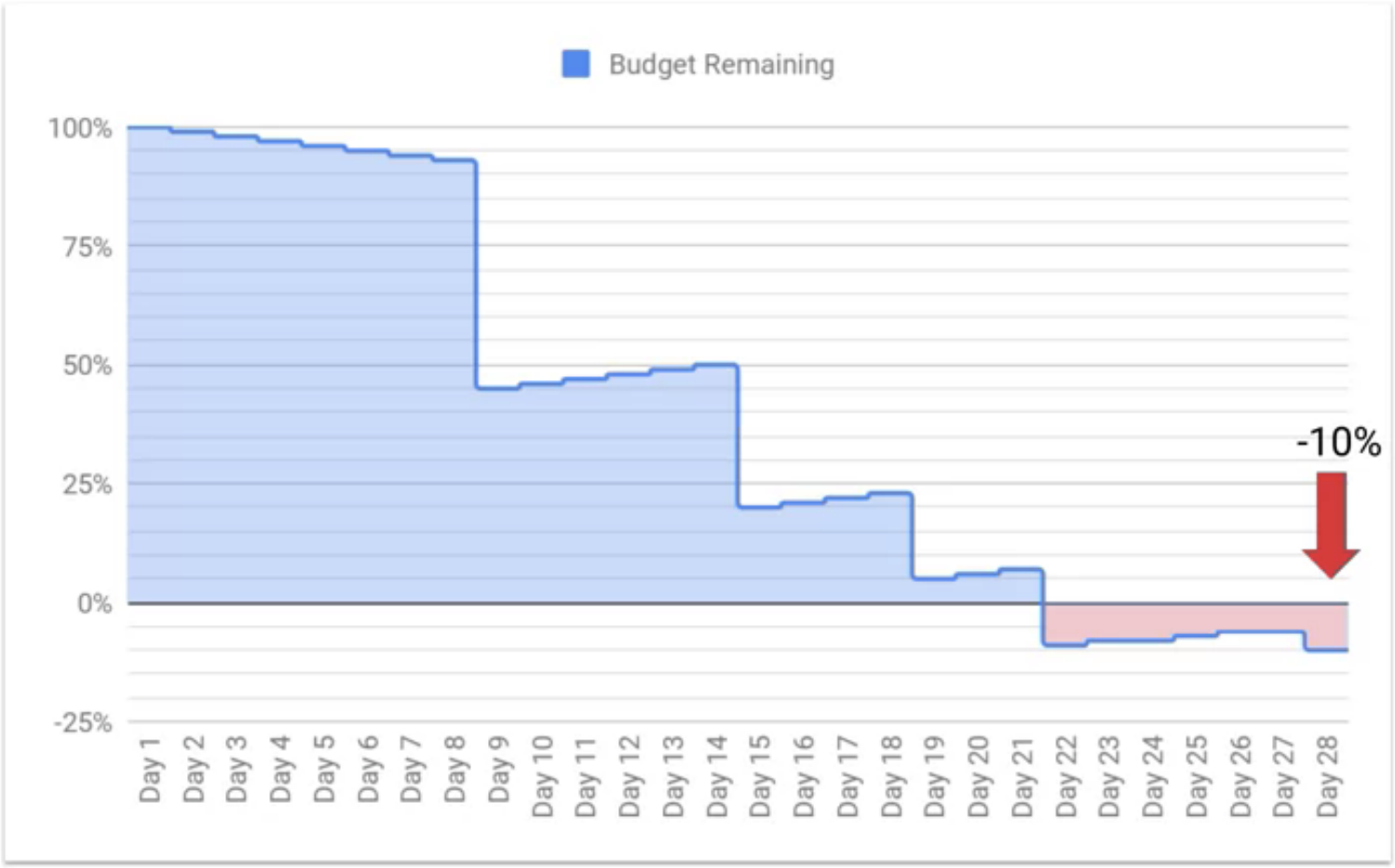
99.9% success = 0.1% failure

Tolerable errors accommodate

- Rolling out new software versions
- Releasing new features
- Planned downtime
- Inevitable hardware failures

0.1% unavailability
x 28 days
= 40.32 mins

Burning down your Error Budget



- Based on 28 day cycle
- Start with something attainable
- Define consequences

What do SREs Do?...

- Measure Everything (Observability)
- Automate themselves out of a Job…
- Never let a good Incident go to waste
- Fight Burnout

How to Automate yourself out of a Job

- Scratch the most annoying itch first
- Iterate, then automate. (reduce ”Toil”)
- Make it useful to others
- Write understandable code
- Make sure it’s robust at (appropriate) scale

What do SREs Do?...

- Measure Everything (Observability)
- Automate themselves out of a Job…
- Never let a good Incident go to waste
- Fight Burnout

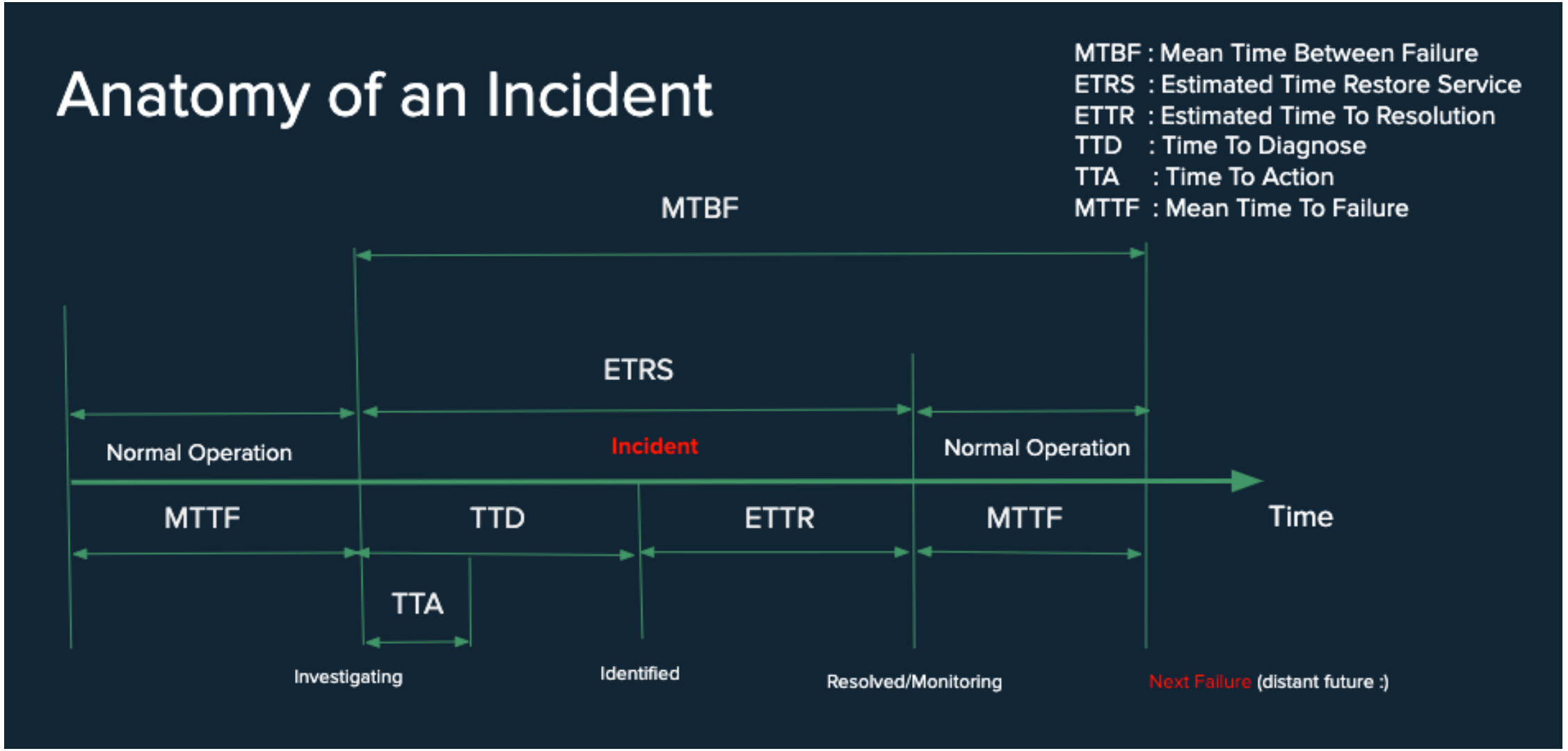
What is an Incident?

An incident is defined as an unplanned interruption or reduction in quality of an IT service (a service interruption). – ITIL Glossary

At Google, incidents are issues that:

- Are escalated (because they're too big to handle alone)
- Require an immediate response
- Require an organized response

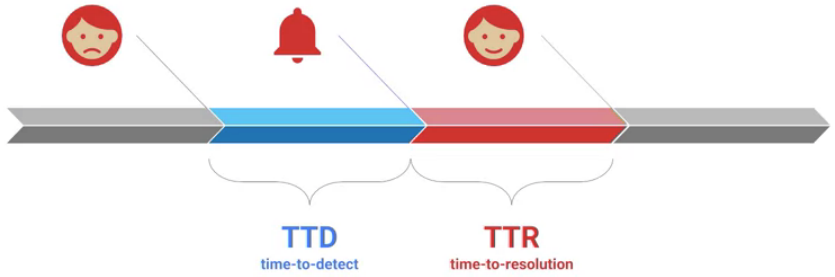
What is an Incident?



SLAs vs. Incidents

Availability %	Downtime per year	Downtime per month	Downtime per week	Downtime per day
90% ("one nine")	**36.5 days**	**72 hours**	**16.8 hours**	**2.4 hours**
95%	18.25 days	36 hours	8.4 hours	1.2 hours
97%	10.96 days	21.6 hours	5.04 hours	43.2 minutes
98%	7.30 days	14.4 hours	3.36 hours	28.8 minutes
99% ("two nines")	**3.65 days**	**7.20 hours**	**1.68 hours**	**14.4 minutes**
99.5%	1.83 days	3.60 hours	50.4 minutes	7.2 minutes
99.8%	17.52 hours	86.23 minutes	20.16 minutes	2.88 minutes
99.9% ("three nines")	**8.76 hours**	**43.8 minutes**	**10.1 minutes**	**1.44 minutes**
99.95%	4.38 hours	21.56 minutes	5.04 minutes	43.2 seconds
99.99% ("four nines")	**52.56 minutes**	**4.38 minutes**	**1.01 minutes**	**8.66 seconds**
99.995%	26.28 minutes	2.16 minutes	30.24 seconds	4.32 seconds
99.999% ("five nines")	**5.26 minutes**	**25.9 seconds**	**6.05 seconds**	**864.3 milliseconds**
99.9999% ("six nines")	**31.5 seconds**	**2.59 seconds**	**604.8 milliseconds**	**86.4 milliseconds**
99.99999% ("seven nines")	**3.15 seconds**	**262.97 milliseconds**	**60.48 milliseconds**	**8.64 milliseconds**
99.999999% ("eight nines")	**315.569 milliseconds**	**26.297 milliseconds**	**6.048 milliseconds**	**0.864 milliseconds**
99.9999999% ("nine nines")	**31.5569 milliseconds**	**2.6297 milliseconds**	**0.6048 milliseconds**	**0.0864 milliseconds**

- 99.9% uptime = max(43.8 mins/month)
- TRS = TTD + TTR
- sigma(TRS_1 + TRS_2 + … TRS_n) <= 43.8 mins/month
- You have the most control over TTD
- So if it takes you 20 mins to assemble a response team…
- you have 23.8 minutes to get everything back online in order to keep your 99.9 SLA…



What do SREs Do?...

- Measure Everything (Observability)
- Automate themselves out of a Job…
- Never let a good Incident go to waste
- Fight Burnout

SREs fight Burnout ...

Resilience Engineering

Black Swan: an unpredictable or unforeseen event, typically one with extreme consequences.

"Make no mistake—the coming N weeks are going to be personally and professionally stressful, and at times we will race to keep ahead of events as they unfold. But we have been preparing for crises for over a decade, and we're ready. At a time when people around the world need information, communication, and computation more than ever, we will ensure that Google is there to help them."

—Benjamin Treynor Sloss, Vice President, Engineering, Google's Site Reliability Engineering Team, March 3, 2020

Resilience Engineering

Black Swan: an unpredictable or unforeseen event, typically one with extreme consequences.

"I know history will be dominated by an improbable event. I just don't know what that event will be; therefore, I should practice Resilience"

- Geoff White, December 2020

Thank You

What do SREs Do?...

- Create procedures, processes and tools to improve the reliability and resiliency of a system.
- Automate themselves out of a Job…
- First Responders when Incidents occur.

What tools do SREs use?...

- SLIs, SLOs, SLAs
- Error Budgets
- Run Books
- Incident Management
- Follow-up Actions (ticketing system)
- Promote well being (anti-burnout)

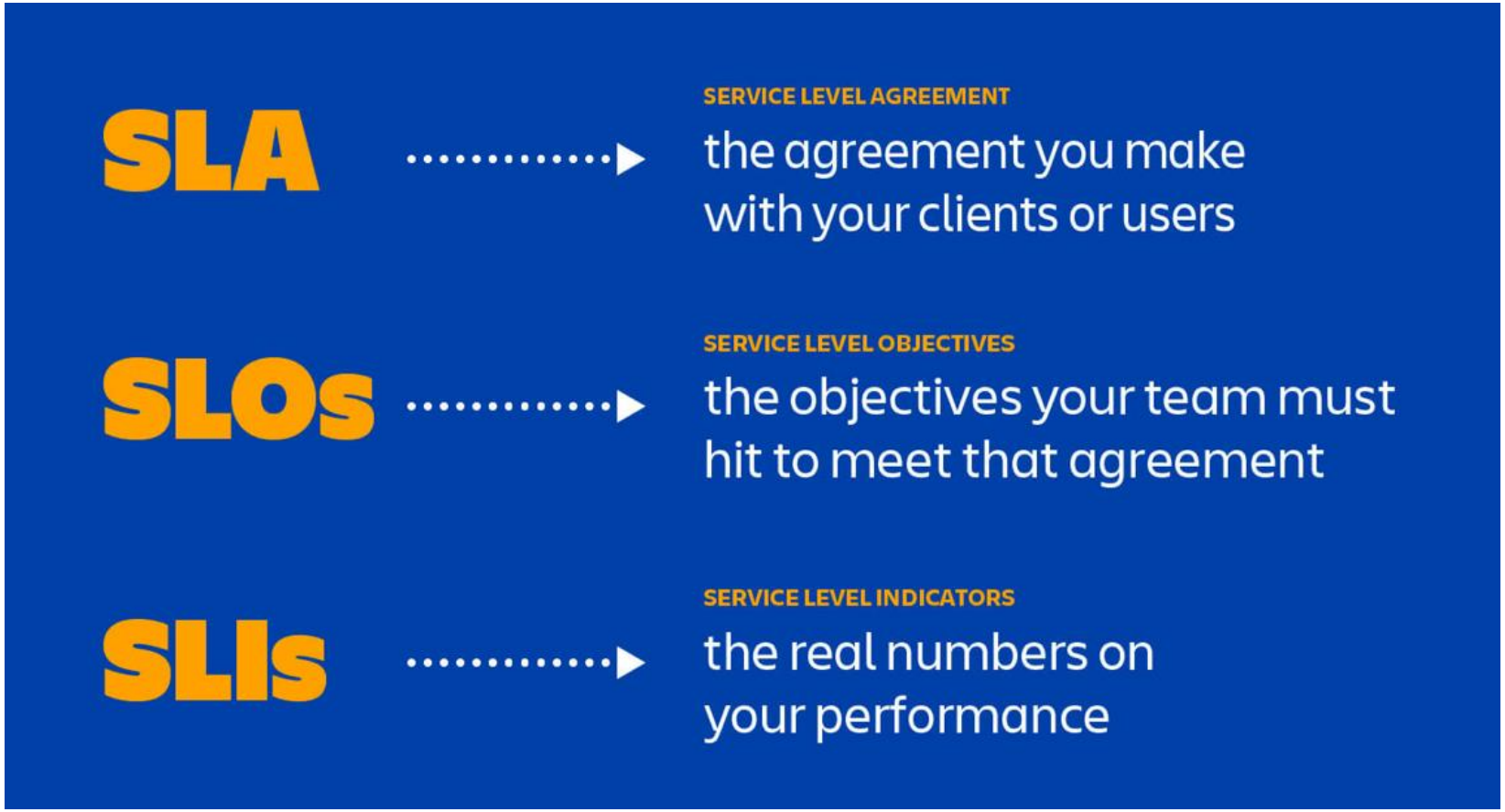
How do SREs use these tools?...

- Create , monitor and leverage SLIs and SLOs to maintain SLAs
- Participate in the establishment of Error Budgets
- Create Run books (check lists, procedures) for use by themselves and others
- Analysis of After Action Reviews (Post Mortems) and RCAs
- Create Follow-up Actions after an Incident and insure that these FUA are resolved.
- Proactive in eliminating stress from the Software Life Cycle

What is SRE series…

- SRE 101
- **All about SLIs, SLOs, Error Budgets and SLAs**
- Incidents and Incident Management
- Trends in SRE (CRE, Chaos Engineering, Resilience Engineering)

Common metrics SREs deal with…



SLI

Calculating the SLIs

Using the preceding metrics, we can calculate our current SLIs over the previous seven days, as shown in Table 2-2.

Table 2-2. Calculations for SLIs over the previous seven days

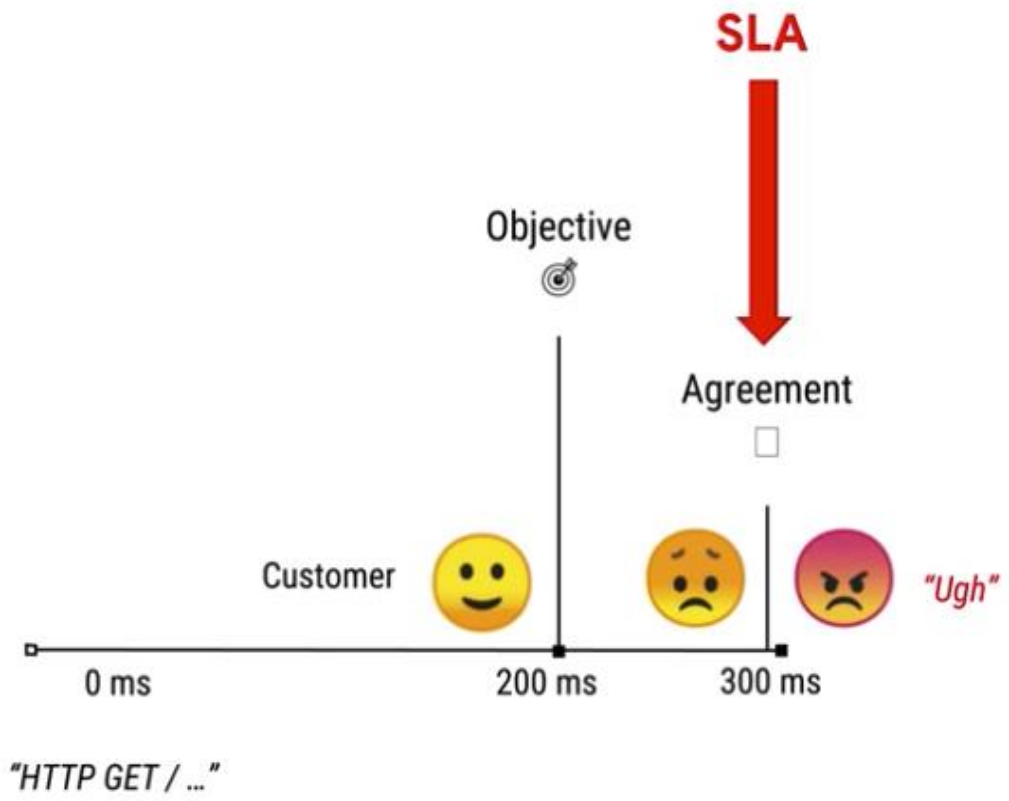
Availability	`sum(rate(http_requests_total{host="api", status!~"5.."}[7d]))`
	`/`
	`sum(rate(http_requests_total{host="api"}[7d])`

Latency	`histogram_quantile(0.9, rate(http_request_duration_seconds_bucket[7d]))` `histogram_quantile(0.99, rate(http_request_duration_seconds_bucket[7d]))`

Type of service	Type of SLI	Description
Request-driven	Availability	The proportion of requests that resulted in a successful response.
Request-driven	Latency	The proportion of requests that were faster than some threshold.
Request-driven	Quality	If the service degrades gracefully when overloaded or when backends are unavailable, you need to measure the proportion of responses that were served in an undegraded state. For example, if the User Data store is unavailable, the game is still playable but uses generic imagery.
Pipeline	Freshness	The proportion of the data that was updated more recently than some time threshold. Ideally this metric counts how many times a user accessed the data, so that it most accurately reflects the user experience.
Pipeline	Correctness	The proportion of records coming into the pipeline that resulted in the correct value coming out.

SLA

- Customer Facing
- Established by contract
- Consequences if not met



Effective SLOs

- Have executive buy-in
- Not generally Customer Facing
- Are accurately measured.
- Indicator of Customer (un)Happiness
- Consequences if not met (error budgets)

Availability

Availability %	Downtime per year	Downtime per month	Downtime per week	Downtime per day
90% ("one nine")	**36.5 days**	**72 hours**	**16.8 hours**	**2.4 hours**
95%	18.25 days	36 hours	8.4 hours	1.2 hours
97%	10.96 days	21.6 hours	5.04 hours	43.2 minutes
98%	7.30 days	14.4 hours	3.36 hours	28.8 minutes
99% ("two nines")	**3.65 days**	**7.20 hours**	**1.68 hours**	**14.4 minutes**
99.5%	1.83 days	3.60 hours	50.4 minutes	7.2 minutes
99.8%	17.52 hours	86.23 minutes	20.16 minutes	2.88 minutes
99.9% ("three nines")	**8.76 hours**	**43.8 minutes**	**10.1 minutes**	**1.44 minutes**
99.95%	4.38 hours	21.56 minutes	5.04 minutes	43.2 seconds
99.99% ("four nines")	**52.56 minutes**	**4.38 minutes**	**1.01 minutes**	**8.66 seconds**
99.995%	26.28 minutes	2.16 minutes	30.24 seconds	4.32 seconds
99.999% ("five nines")	**5.26 minutes**	**25.9 seconds**	**6.05 seconds**	**864.3 milliseconds**
99.9999% ("six nines")	**31.5 seconds**	**2.59 seconds**	**604.8 milliseconds**	**86.4 milliseconds**
99.99999% ("seven nines")	**3.15 seconds**	**262.97 milliseconds**	**60.48 milliseconds**	**8.64 milliseconds**
99.999999% ("eight nines")	**315.569 milliseconds**	**26.297 milliseconds**	**6.048 milliseconds**	**0.864 milliseconds**
99.9999999% ("nine nines")	**31.5569 milliseconds**	**2.6297 milliseconds**	**0.6048 milliseconds**	**0.0864 milliseconds**

Error Budget

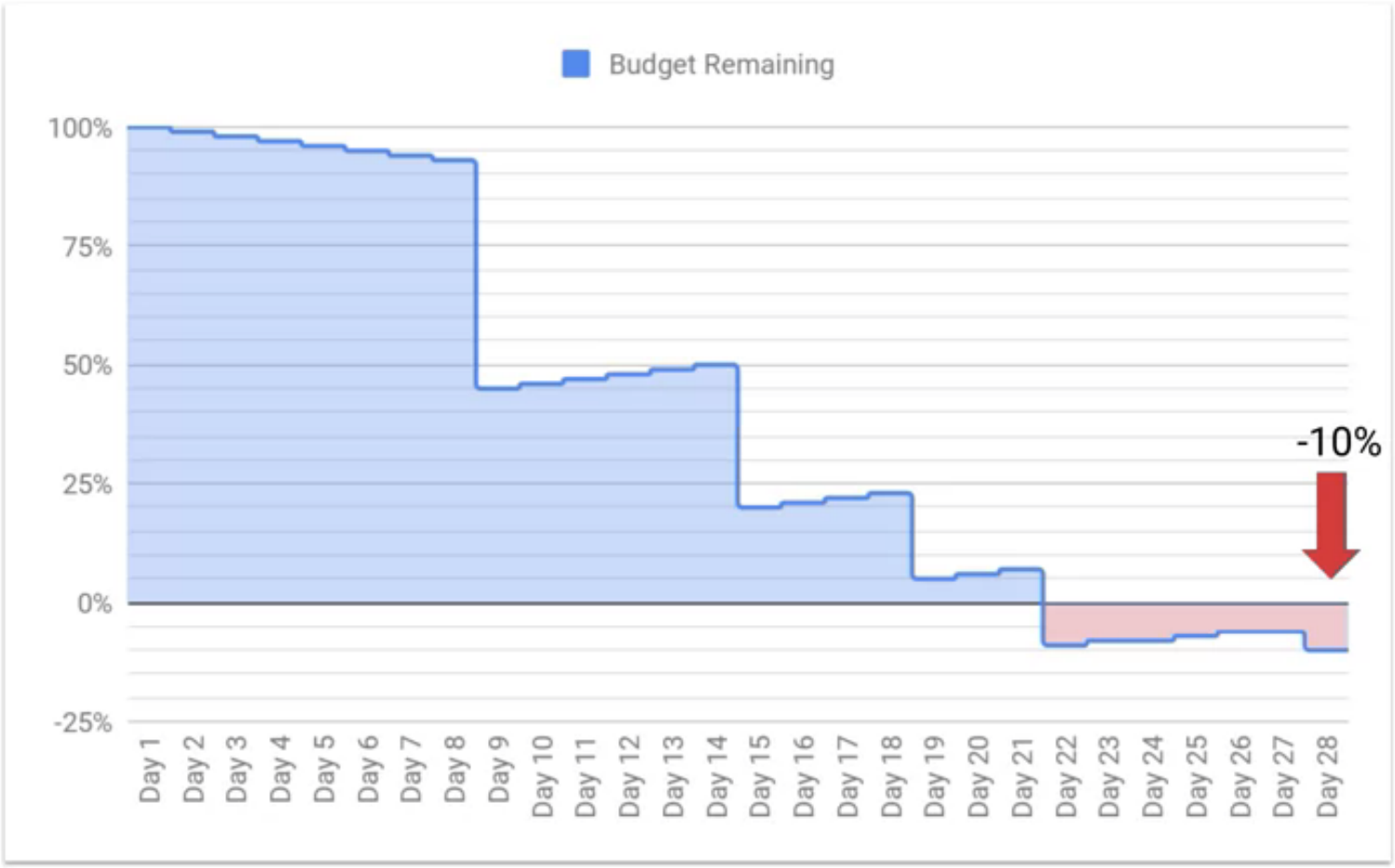
99.9% success = 0.1% failure

Tolerable errors accommodate

- Rolling out new software versions
- Releasing new features
- Planned downtime
- Inevitable hardware failures

0.1% unavailability
x 28 days
= 40.32 mins

Burning down your Error Budget



- Based on 28 day cycle
- Start with something attainable
- Define consequences

SLIs vs SLOs vs SLAs

Benefits

- Common incentives for Devs and SREs
- Dev team can self-manage risk
- Unrealistic goals become unattractive

Antipattern: SLO = SLA

Always set your SLOs tighter than your SLAs (e.g., SLO: 99.95%, SLA: 99.9%)

Antipattern: SLI = OKR (objectives and key results)/KPI (key performance indicator)

Goodhart's law applies here: when a measure becomes a target, it ceases to be a good measure.

What is SRE series…

- SRE 101
- All about SLIs, SLOs, Error Budgets and SLAs
- **Incidents and Incident Management**
- Trends in SRE (CRE, Chaos Engineering, Resilience Engineering)

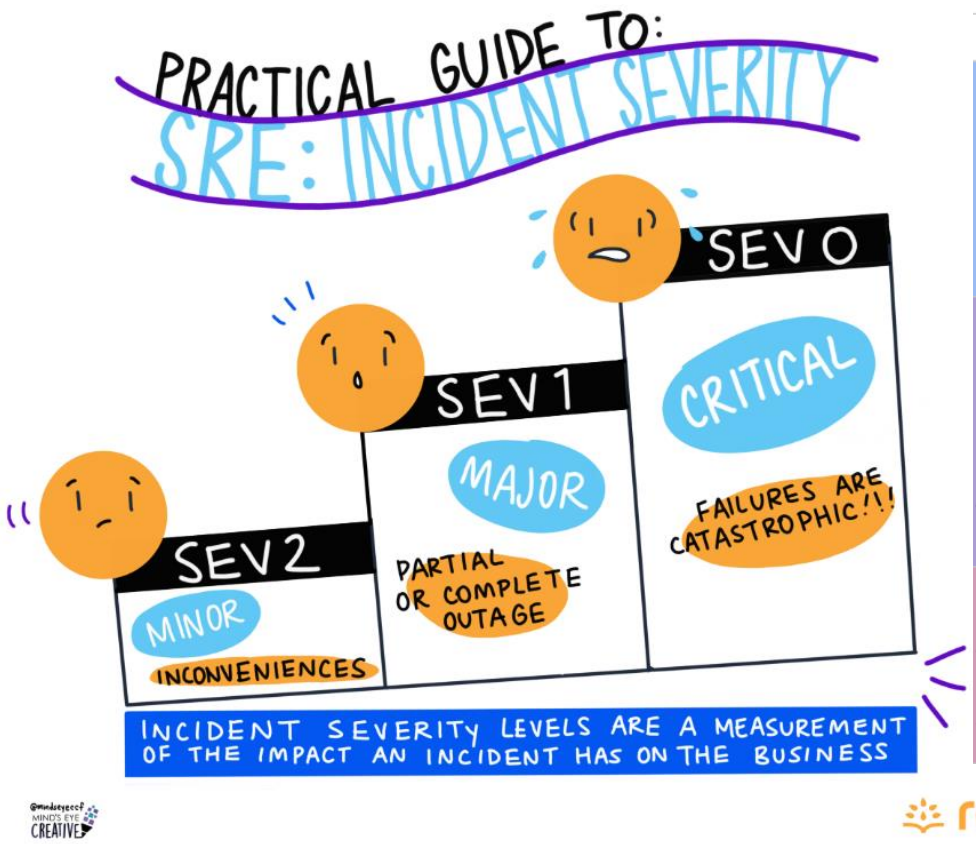
What is an Incident?

An incident is defined as an unplanned interruption or reduction in quality of an IT service (a service interruption). – ITIL Glossary

At Google, incidents are issues that:

- Are escalated (because they're too big to handle alone)
- Require an immediate response
- Require an organized response

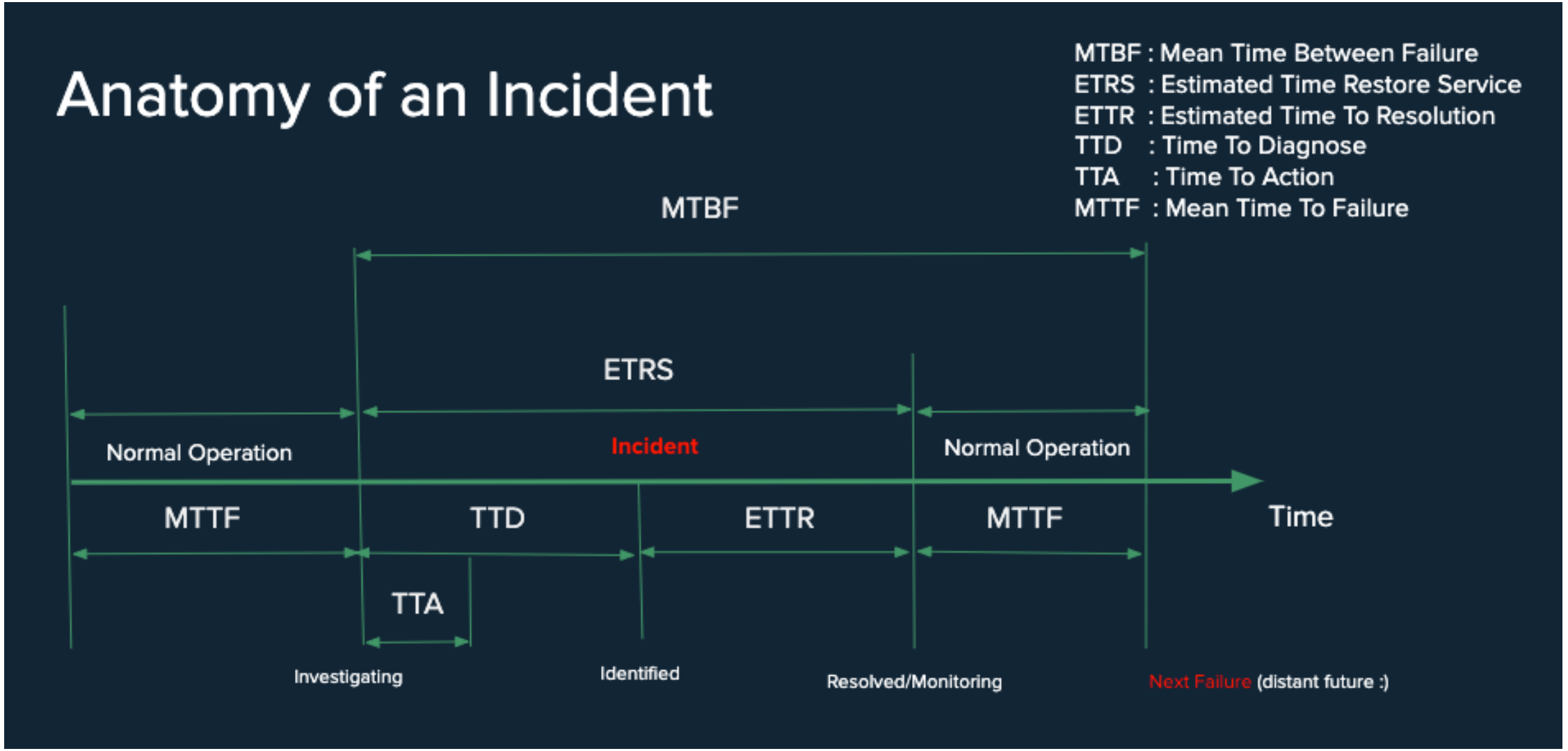
Severity Levels



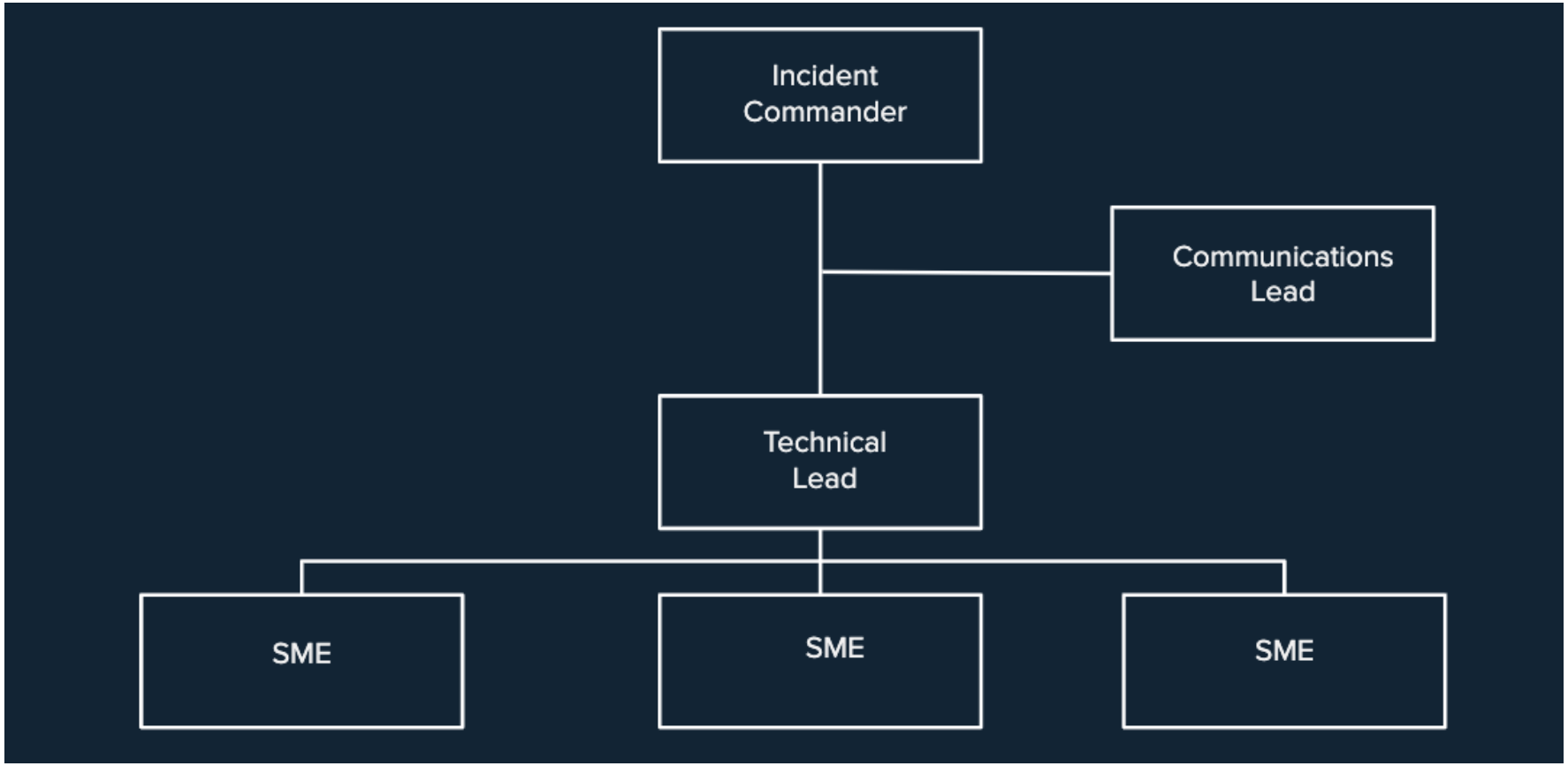
	Critical/P0	High/P1	Medium/P2
	Sev - 0	**Sev - 1**	**Sev - 2**
Utility Apps - Instacart, Amazon	1. App Crashing 2. Customer PII Leak	1. Financial Loss due to Tampering - Manipulation of Price/Offers	1. Partial/Incorrect Data for some products - e.g. missing product images or wrong product description 2. Invalid Promotions - e.g. Expired/Invalid offers
Entertainment Apps - Youtube, Netflix, TikTok	1. App Crashing 2. Customer PII Leak	1. Financial Loss due to Tampering - Manipulation of Price/Offers 2. DRM bypass - Digital Rights Management	1. Partial/Incorrect Data for some items. 2. Invalid Promotions - e.g. Expired/Invalid offers 3. Content rating issues - e.g. adult content not rated as 'A', Parental Lock
Financial Apps - Banking Apps, Trading Apps, Bitcoin Wallets/Apps	1. App Crashing 2. Customer PII Leak 3. Tamper Financial Transactions	1. Partial/Incorrect Data for some products/users - e.g. Incorrect user info, bank details 2. Financial Loss due to Tampering - Manipulation of Price/Offers	1. Invalid Promotions - e.g. Expired/Invalid offers

rootly

What is an Incident?



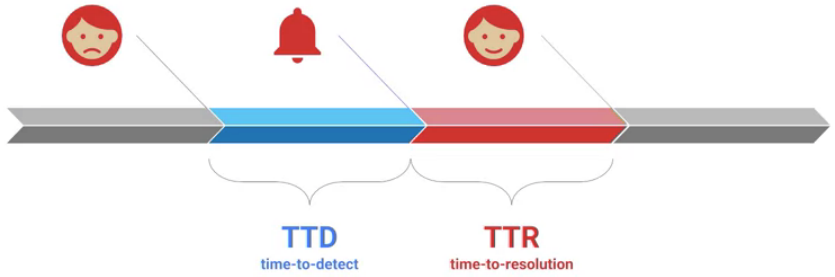
Incident Response Team



SLAs vs. Incidents

Availability %	Downtime per year	Downtime per month	Downtime per week	Downtime per day
90% ("one nine")	**36.5 days**	**72 hours**	**16.8 hours**	**2.4 hours**
95%	18.25 days	36 hours	8.4 hours	1.2 hours
97%	10.96 days	21.6 hours	5.04 hours	43.2 minutes
98%	7.30 days	14.4 hours	3.36 hours	28.8 minutes
99% ("two nines")	**3.65 days**	**7.20 hours**	**1.68 hours**	**14.4 minutes**
99.5%	1.83 days	3.60 hours	50.4 minutes	7.2 minutes
99.8%	17.52 hours	86.23 minutes	20.16 minutes	2.88 minutes
99.9% ("three nines")	**8.76 hours**	**43.8 minutes**	**10.1 minutes**	**1.44 minutes**
99.95%	4.38 hours	21.56 minutes	5.04 minutes	43.2 seconds
99.99% ("four nines")	**52.56 minutes**	**4.38 minutes**	**1.01 minutes**	**8.66 seconds**
99.995%	26.28 minutes	2.16 minutes	30.24 seconds	4.32 seconds
99.999% ("five nines")	**5.26 minutes**	**25.9 seconds**	**6.05 seconds**	**864.3 milliseconds**
99.9999% ("six nines")	**31.5 seconds**	**2.59 seconds**	**604.8 milliseconds**	**86.4 milliseconds**
99.99999% ("seven nines")	**3.15 seconds**	**262.97 milliseconds**	**60.48 milliseconds**	**8.64 milliseconds**
99.999999% ("eight nines")	**315.569 milliseconds**	**26.297 milliseconds**	**6.048 milliseconds**	**0.864 milliseconds**
99.9999999% ("nine nines")	**31.5569 milliseconds**	**2.6297 milliseconds**	**0.6048 milliseconds**	**0.0864 milliseconds**

- 99.9% uptime = max(43.8 mins/month)
- TRS = TTD + TTR
- sigma(TRS_1 + TRS_2 + … TRS_n) <= 43.8 mins/month
- You have the most control over TTD
- So if it takes you 20 mins to assemble a response team…
- you have 23.8 minutes to get everything back online in order to keep your 99.9 SLA…



What is SRE series…

- SRE 101
- All about SLIs, SLOs, Error Budgets and SLAs
- Incidents and Incident Management
- **Trends in SRE (CRE, Chaos Engineering, Resilience Engineering)**

What do CREs Do?...

- CRE is what you get when you take the principles and lessons of SRE and apply them towards customers.
- Analyze metrics to drive efficient adoption of the product
- Escalation point for Incidents, reducing Customer Anxiety
- Goal: *Drive Customer Anxiety -> 0 where; Anxiety = 1/Reliability*

Chaos Engineering

- Incident Replay
- Drills, drills and drills.
- Go fast and break things
- What to do if you have plenty of error budget
- Looking for trouble… you don’t need to introduce chaos, it’s already in the system... Waiting for you.

Resilience Engineering

Black Swan: an unpredictable or unforeseen event, typically one with extreme consequences.

"Make no mistake—the coming N weeks are going to be personally and professionally stressful, and at times we will race to keep ahead of events as they unfold. But we have been preparing for crises for over a decade, and we're ready. At a time when people around the world need information, communication, and computation more than ever, we will ensure that Google is there to help them."

—Benjamin Treynor Sloss, Vice President, Engineering, Google's Site Reliability Engineering Team, March 3, 2020

Resilience Engineering

Black Swan: an unpredictable or unforeseen event, typically one with extreme consequences.

"I know history will be dominated by an improbable event. I just don't know what that event will be; therefore, I should practice Resilience"

- Geoff White, December 2020

SREs fight Burnout ...