



Matt Brown, @xleem  
Customer Reliability Engineer  
November, 2018

# Know thy enemy

How to prioritize and communicate risk



# Matt Brown



# Matt Brown

I'm a kiwi! Live & Work in Cambridge!





# Matt Brown

I'm a kiwi! Live & Work in Cambridge!

1st time @ DevOpsDays ever



The slide has a blue header section containing the title and author information, followed by a black main content area featuring a video feed of a speaker at a podium.

**Reliability When Everything is a Platform**

Why we need to SRE our customers

Dave Rensin  
rensin@google.com

usenix  
THE ADVANCED SYSTEMS  
ASSOCIATION

SRE CON AMERICAS

A video feed shows a man speaking at a podium with a microphone. The podium has a sign that reads "SRE CON AMERICAS".



<https://goo.gl/T83gcf>

# Matt Brown

I'm a kiwi! Live & Work in Cambridge!

1st time @ DevOpsDays ever

Tech Lead for CRE @ Google

# Agenda

- What is risk?, some observations
- Approaches to risk, why prioritization is needed
- CRE's first attempt at prioritization
- What Risk Management can teach us about prioritization



# What is risk?

Google Cloud

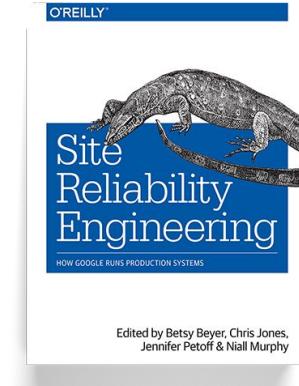


a situation involving exposure  
to danger.

define:risk  
[google.com](http://google.com)

# SLO is critical to SRE

SLI	SLO	SLA	Error Budget
<b>indicator</b>  A measurable quantity representing what's important to users	<b>objective</b>  The target you want your SLI to reach	<b>agreement</b>  Consequences when the SLO is not met.	1 - SLO  Our primary tool for prioritizing our work.
		 Not relevant to today's talk.	



# A situation involving consumption of the error budget





# My observations on risk

Google Cloud

# What's the biggest risk to your app / service





Image: <https://unsplash.com/photos/wS4-XYTyG5k>

# Many flavours

# Personal



Image: <https://pixabay.com/en/german-wasp-insect-animal-3216970/>, CC0



Image: <https://unsplash.com/photos/wS4-XYTyG5k>

**Some risk is  
good!!**



# Approaches to risk

Google Cloud

# Ignorance

Is not bliss



Image: <https://www.pexels.com/photo/beach-wave-948331/>, CC0



# Paranoia

Is just as bad

Eliminate  
Reduce  
Avoid

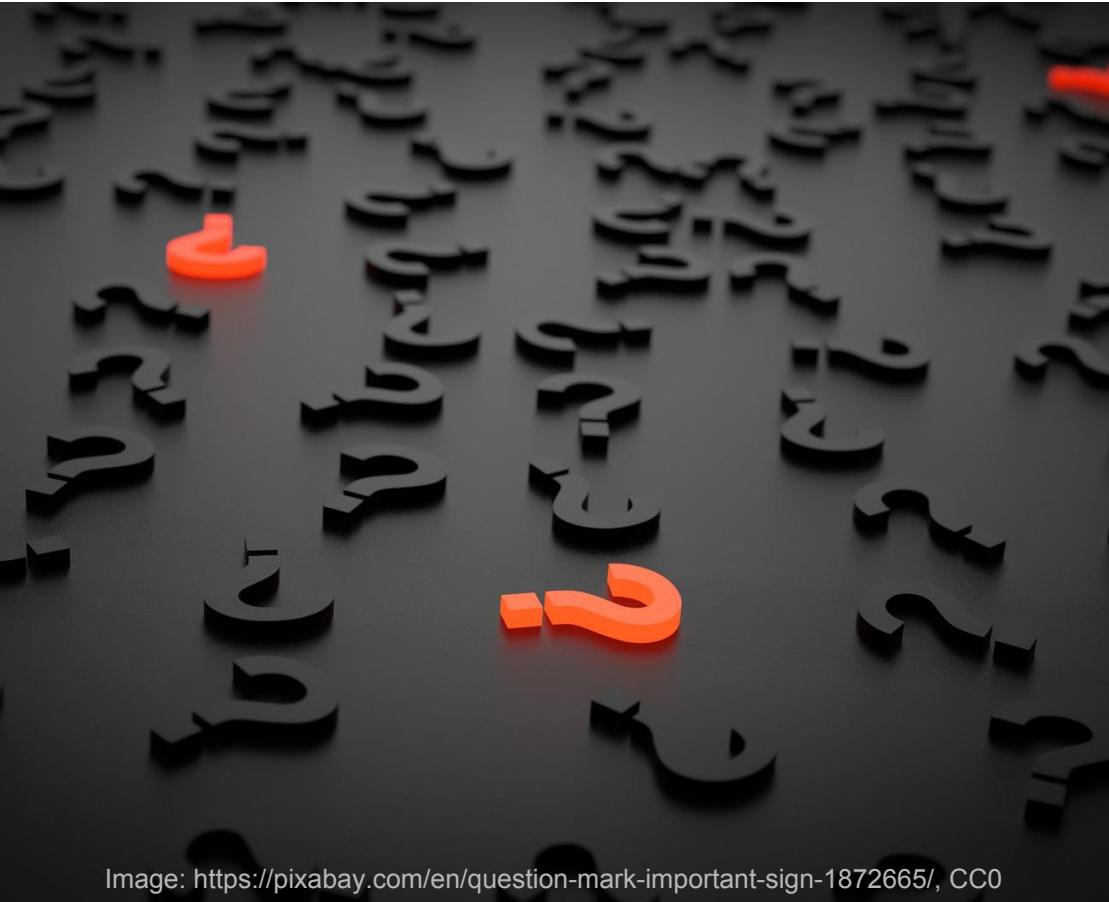


Image: [https://unsplash.com/photos/efc\\_wvILRs4](https://unsplash.com/photos/efc_wvILRs4)



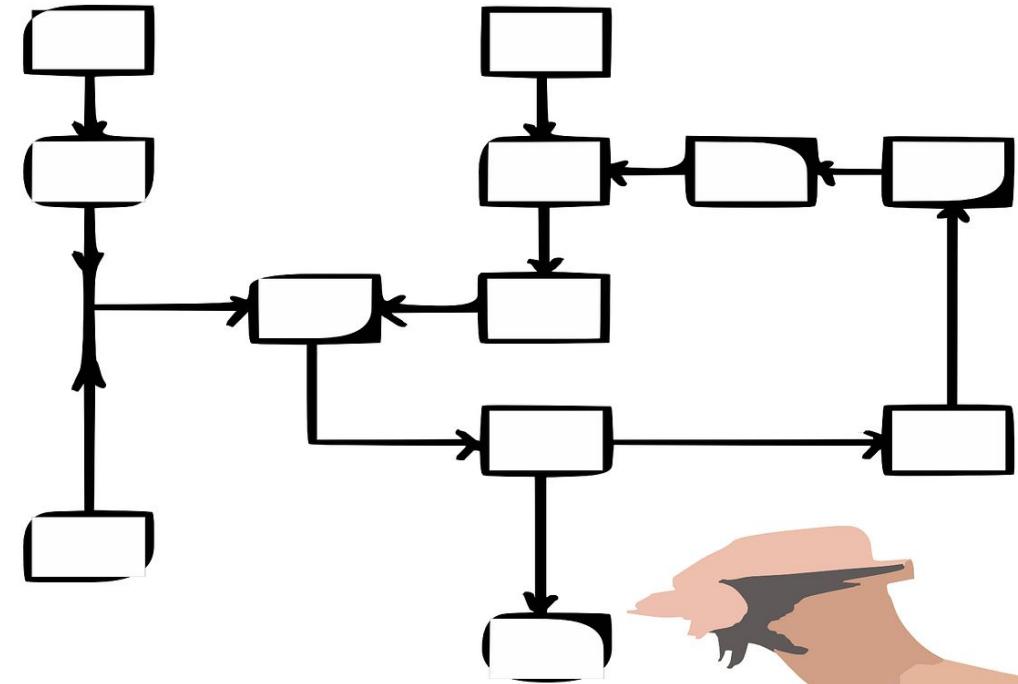
# Prioritizing risk

Google Cloud



# Intuition

# System / Process





# The Risk Matrix

Google Cloud

# Likelihood



# Impact



# The Matrix

Great display, easy to understand

Terrible for prioritization

	Catastrophic	Damaging	Minimal
Frequent	Overload results in slow or dropped requests during the peak hour each day.	The wrong server is turned off and requests are dropped.	Restarts for weekly upgrades drop in-progress requests (i.e., no lame ducking).
Common	A bad release takes the entire service down. Rollback is not tested.	Users report an outage before monitoring and alerting notifies the operator.	A daylight savings bug drops requests.
Rare	There is a physical failure in the hosting location that requires complete restoration from a backup or disaster recovery plan.	Overload results in a cascading failure. Manual intervention is required to halt or fix the issue.	A leap year bug causes all servers to restart and drop requests.



# Expected Cost

Google Cloud

# Expected cost

- Risk Management is a well studied field
- Expected Cost = Probability (Likelihood) \* Cost (Impact)
- Costs are easily comparable, solving our matrix problems.
- Can we rephrase our risk characteristics to be able to use this?
- \$\$ Cost is not always easy for SRE to estimate
- But we already have a budget. A cost is something you spend. We must be able to merge these concepts!

# Expected cost for SRE

Likelihood	Impact	Cost
Quantified as MTBF (days)	Quantified as MTTR (typically minutes).	Annual error budget minutes we expect this risk to consume.
Ideally from historical data.	How much of your error budget will this risk consume?	
Pragmatically we estimate. (ETBF)	ETTD  ETTR  % Users	



# Risk Input

Risk Name				
Operator accidentally deletes database; restore from backup required				
Bug in new release breaks uncommon request type				
Physical failure of hosting; implement back-up/DR plan				
Overload causes 15% slow requests at peak each day				
No lame-ducking/health-checks; restarts drop in-flight requests				

# Risk Input

Risk Name	ETTD (mins)	ETTR (mins)	% Users	ETBF
Operator accidentally deletes database; restore from backup required	5	480	100	1460
Bug in new release breaks uncommon request type	1440	30	2	90
Physical failure of hosting; implement back-up/DR plan	5	720	100	1095
Overload causes 15% slow requests at peak each day	0	60	15	1
No lame-ducking/health-checks; restarts drop in-flight requests	0	1	100	7

# Calculated Expected Cost

Risk Name	ETTD (mins)	ETTR (mins)	% Users	ETBF	Bad mins/year
Operator accidentally deletes database	5	480	100	1460	<b>121</b>
Bug in new release breaks uncommon request type	1440	30	2	90	<b>119</b>
Physical failure of hosting; implement back-up/DR plan	5	720	100	1095	<b>242</b>
Overload causes 15% slow requests at peak each day	0	60	15	1	<b>3287</b>
No lame-ducking/health-checks; restarts drop requests	0	1	100	7	<b>52</b>

# Stack Rank

How does this compare to your first guess?

Risk	Bad mins/year
Overload causes 15% slow requests at peak each day	3287
Physical failure of hosting; implement back-up/DR plan	242
Operator accidentally deletes database	121
Bug in new release breaks uncommon request type	119
No lame-ducking/health-checks; restarts drop requests	52

Risk	Bad mins/year	99.99%
Overload causes 15% slow requests at peak each day	3287	
Physical failure of hosting; implement back-up/DR plan	242	
Operator accidentally deletes database	121	
Bug in new release breaks uncommon request type	119	
No lame-ducking/health-checks; restarts drop requests	52	

# Error budget analysis

99.99% SLO

52.596 mins/year budget

25% threshold (13.1 mins)

Risk	Bad mins/year	99.9%
Overload causes 15% slow requests at peak each day	3287	
Physical failure of hosting; implement back-up/DR plan	242	
Operator accidentally deletes database	121	
Bug in new release breaks uncommon request type	119	
No lame-ducking/health-checks; restarts drop requests	52	

# Error budget analysis

99.9% SLO

525.96 mins/year budget

25% threshold (131 mins)

Risk	Bad mins/year	99.9%
Overload causes 15% slow requests at peak each day	3287	
Physical failure of hosting; implement back-up/DR plan	242	
Operator accidentally deletes database	121	
Bug in new release breaks uncommon request type	119	
...	407	

# Error budget analysis

99.9% SLO

525.96 mins/year budget

25% threshold (131 mins)

# Takeaways

SLO	Risks abound	Prioritization	Estimated Cost	Try it today!
You need an SLO, and an error budget.	The world is constantly trying to threaten our SLO.	We can't engage with every risk, we need to prioritize.	A well established technique for comparing risks.	It's easy to apply this technique.
Foundation for all SRE work and prioritization.	Our job as SREs is to manage that risk.	Humans are terrible at prioritizing risk.	Breaking a risk into characteristics gives opportunity to reduce bias.	Here's a template spreadsheet you can use: <a href="https://goo.gl/bnsPjZ">https://goo.gl/bnsPjZ</a>



# Thank you!

# Feedback Welcome

These slides

<https://goo.gl/bwT7eC>



Me

[mattbrown@google.com](mailto:mattbrown@google.com)

@xleem