

GOTO **AMSTERDAM 2023**

#GOTOams



GOTO
Guide

LET US HELP YOU

Ask questions
through **the app**



also remember to rate session



THANK YOU!

#GOTOams



Ten things we've learned from running production infrastructure at Google

GOTO Amsterdam 2023

Christof Leng

cleng@google.com

Twitter @moroquen

linkedin.com/in/christofleng/



Site Reliability Engineering

About the speaker

Site Reliability Engineer @ Google since 2014

Based in Google's Munich office

Worked on Ads, Cloud, and developer tooling

Currently working on central SRE programs and tools

Before Google: researching reliable distributed systems



“

SRE is what happens when
you ask a software engineer
to design and run operations. ”

Benjamin Treynor Sloss, Vice President of 24x7 Engineering, *Google*



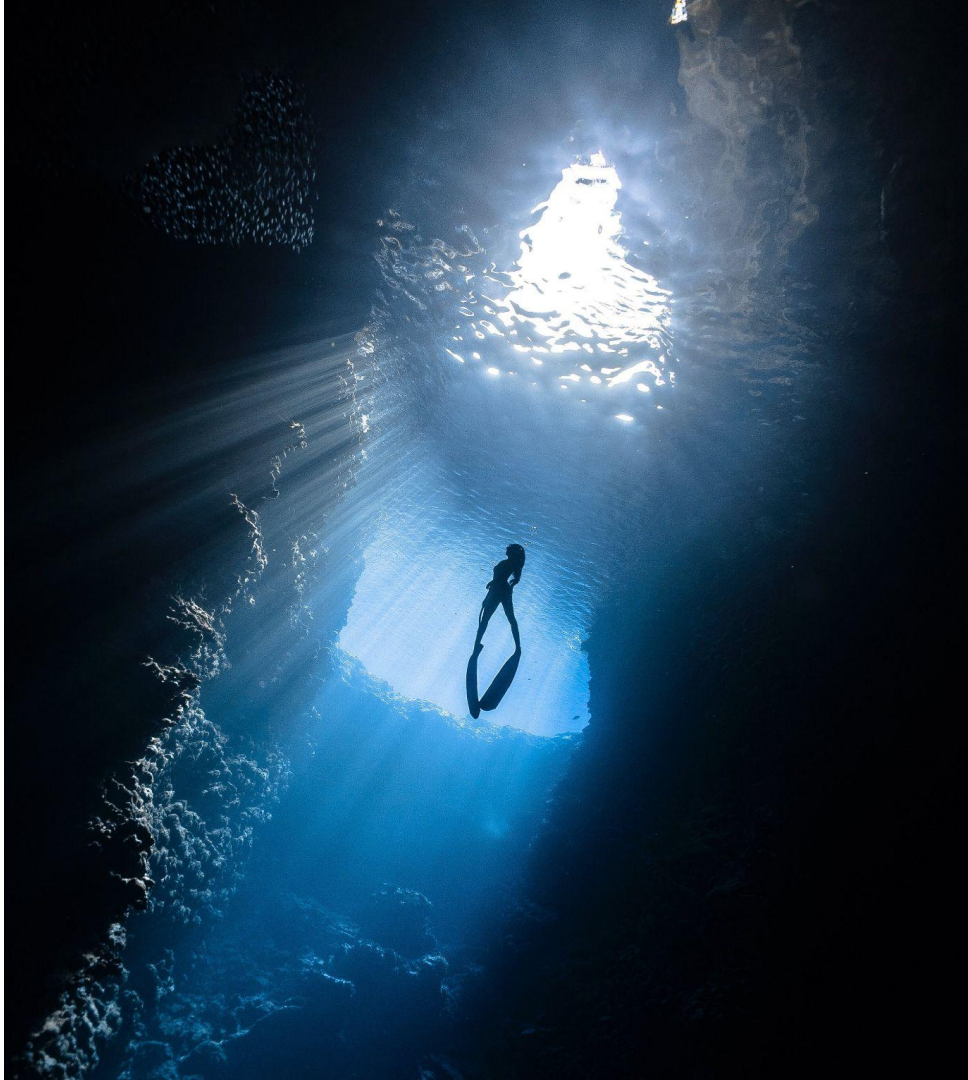
SRE works on many different Google products



"Culture eats strategy for breakfast"
(Commonly attributed to Peter Drucker)

#1 Reliability can't be taken for granted

- Easy to forget while there's plenty.
- Often too late to fix when it runs out.
- There always needs to be a voice for reliability.
- Hope is not a strategy.
(Google SRE motto)
- Planning for reliability needs to start early ("shift left").





#2 Cattle vs. Pets

- Pets have names, personalities, and are expensive to maintain.
- Cattle have numbers, are uniform, and are individually cheaper.
- Standardization (hardware, infrastructure, architecture) matters for scalability.
- Cognitive load is an important bottleneck.

#3 Blamelessness

- Assume that everyone is competent and well-intentioned.
- Don't try "fixing" people. Fix systems and processes.
- Everyone must feel comfortable coming forward without fear of consequences.
- Only when we have the complete information, will we be able to improve.
- When problems are swept under the rug, they accumulate.



#4 Measure what matters

- Agree on measurable goals (SLOs) to prevent conflicts.
- Focus on the user - measure what they care about.
- Anything you don't measure gets worse.
- Advise, don't block - people find ways around the gatekeeper.



A Word On Ops



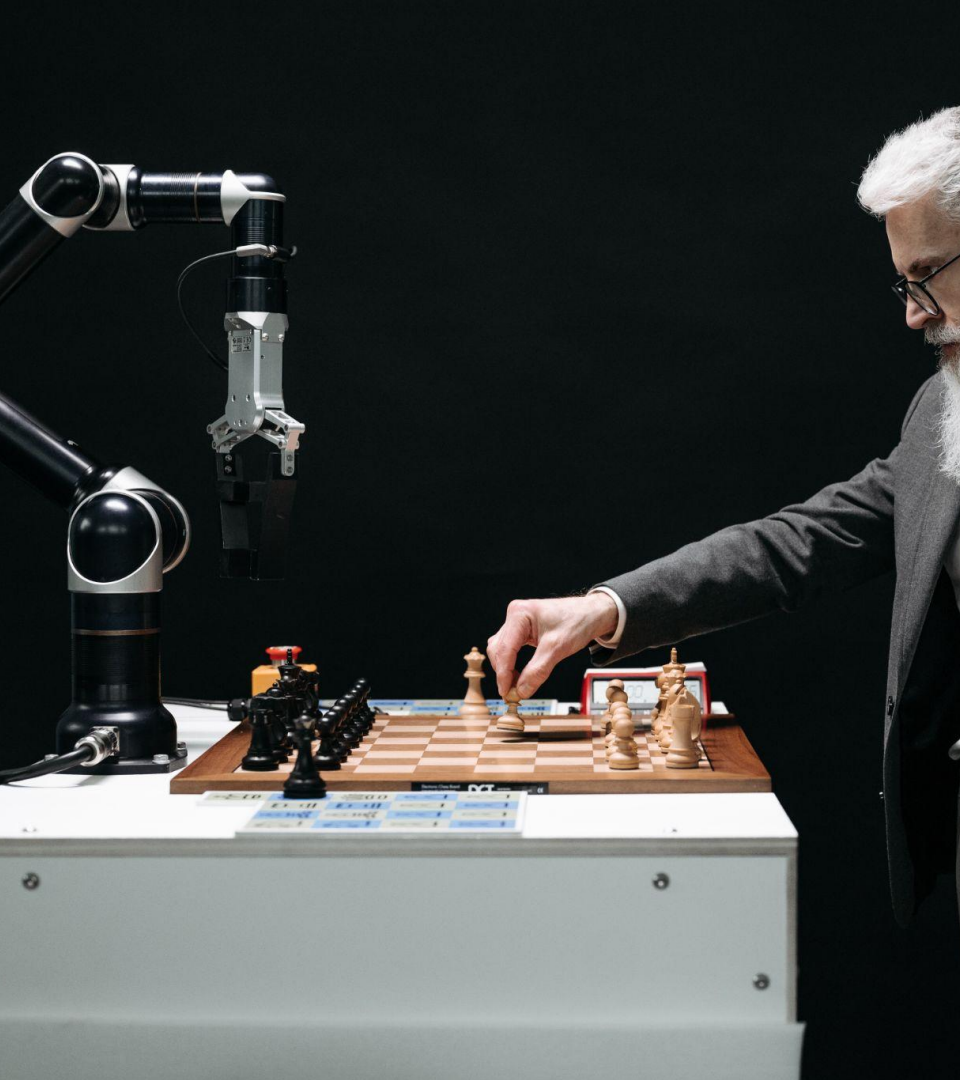
#5 The best way to learn how a system works is watching its failure modes

- Oncall is not a purpose, it's a means to an end.
- Failures are complex, easy to misinterpret from a distance.
- Seemingly unrelated failures may have a deeper connection.
- You need skin in the game to build up credibility.

#6 No heroes

- Heroism is bad - for the hero, the team, and the system.
- SRE > SLA
- The oncaller's job is to make sure the problem is fixed, not to fix the problem.
- The oncaller is never alone. Escalate.





#7 Automate yourself out of your (current) job

- Don't send a human to do a robot's job.
- Automation scales better than manual work.
- It's actually often primarily about consistency, not efficiency.
- Removing manual tasks will free up time for engineering work.



Change is constant

#8 Change is the #1 reason for outages

- Find the right reliability/velocity balance for your product.
- Minimize unnecessary risk from changes.
- Don't test in production.
- Use GitOps.
- Don't deploy on "Fridays" (for varying values of \$Friday).
- Production freezes don't solve the underlying problem.





#9 Outages are inevitable

- You need change. Change has risks.
- The goal is not to prevent outages entirely, but to limit their overall cost.
- Try to mitigate first, root cause later.
- Be able to roll back your change quickly.
- Use written communication for incident management.
- Organizational transparency helps with root causing. Read code, not docs.

#10 No Haunted Graveyards

- A system can become so fragile and complex that no one dares to touch it.
- Complexity is a booby trap for change - complex systems need constant fixing.
- Don't accept neglect ("broken windows"). It's a slippery slope.
- Anyone can build complex systems - try building simple systems!



So what did we actually learn?

Common Themes



Relationships

Divided we fall, united we stand - we're all on the same team: Dev, Ops, SRE, Product, etc..



Change

You can't keep doing what you're doing. If production doesn't improve, it will get worse and may reach a tipping point.



Engineering

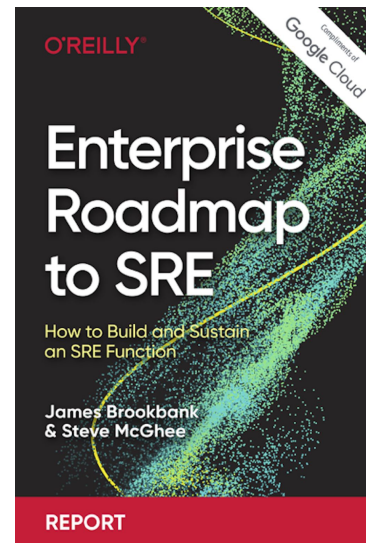
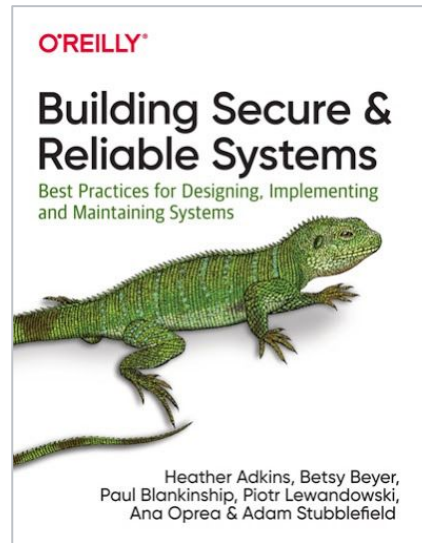
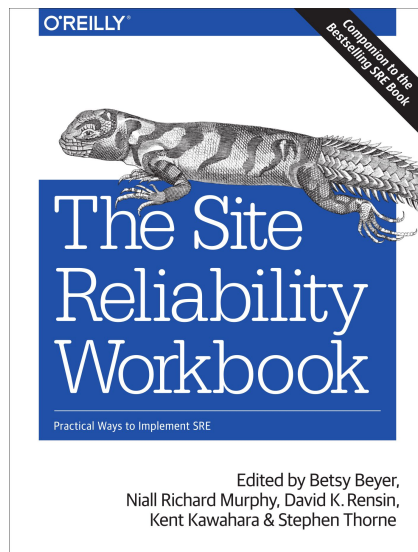
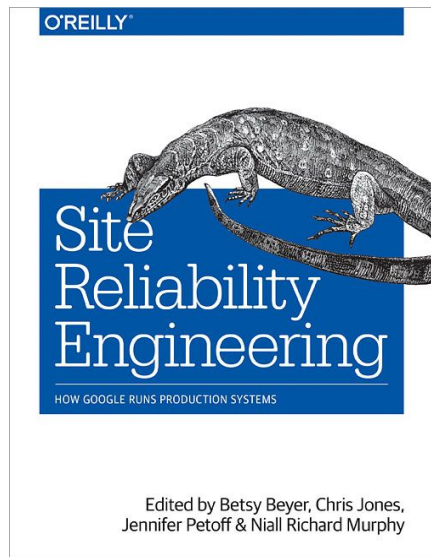
Discipline and hard work will only get you so far. The machine cannot be sustained by feeding it human sweat.



Simplicity

Boring is beautiful. Don't outsmart yourself. Your future self is going to regret it.

Follow us on Twitter: [@googlesre](https://twitter.com/googlesre). Find Google SRE publications—including the SRE Books, articles, trainings, and more—for free at sre.google/resources.



Book covers copyright O'Reilly Media. Used with permission.



GOTO
Guide



Remember to
rate this session

THANK YOU!



Download on the
App Store



GET IT ON
Google Play

#GOTOams