# Recap SREcon19 Europe


Ingo Averdunk
Distinguished Engineer
IBM
@ingoa

Pavlos Ratis
Site Reliability Engineer
HolidayCheck
@dastergon

# TL:DR;

- Theme of SREcon 2019 Europe: Core principles, Unsolved / Open Problems in SRE
- 819 attendees; 278 companies; ~100 attendees from D/A/CH

- SRE is a maturing profession, we're about to enter the 3rd age of SRE
- Still, there are unanswered questions
  - Normalization of deviance. Willingness to accept friction
  - SRE Journey: Starting, How to train SREs, team structure, solo / remote SRE, culture change, etc.
  - Value, limits and risks of SLO, AIOps and Automation
  - Justification of investments & improvements; tying to the business objectives

- A lot of presentations centered on Observability, SLO and Distributed Tracing
- Increasing discussion on the Interdependency of services, especially in a micro-services world
- Due to the increasing complexity, different approaches are being considered
  - Systems and Control Theory to treat safety as a control problem, not a failure problem
  - Service Mesh
  - Statistics, AI / ML

- This presentation is like "Speed-Dating" - a super-condensed summary of 10 hrs / 300+ slides into a 30min session. It is meant to be a teaser, to motivate people listening to the replays of topics they find interesting.

# Some facts upfront

- SREcon is a gathering of engineers who care deeply about site reliability, systems engineering, and working with complex distributed systems at scale.

- Europe 2019: 819 attendees; 278 companies
  (Americas: 650 attendees, AP: 300 attendees)

- Theme of SREcon 2019: Unsolved / Open problems in SRE; Core SRE principles

  This year, SREcon EMEA will focus on unanswered questions in SRE. We want to discuss the problems no one is talking about, the problems everyone complains about with no real consensus on how to solve them. If you think there is an elephant in the room that we, the SRE community, have failed to talk about—come and tell us about it!

- Attendance

  Obviously the likes of: Google, Facebook, LinkedIn, GitLab, Cloudflare
  Amadeus, Blomberg, booking.com, criteo, Demonware, Disney, Elastic, Goldman Sachs, HolidayCheck, Hostinger, Huawei, Humio, IBM, ING, Intercom, karriere.at, Maersk, Microlise, Microsoft, Monzo Bank, Oracle, Outbrain, Paddy Power Betfair, SEMrush, Shopify, SIXT, Sparkpost, Squadcast, Squarespace, StackState, Tableau, Talentsoft, Twill, Udemy, Workday, Xanadu, Yandex, Zalando, Zendesk, etc.

  DE (65), CH (22), AT (7)

# SREcon 2019 Europe Theme

## Unsolved/Open Problems in SRE

- Should developers be oncall?
  Does being an SRE always mean being oncall?
- Does every company need SRE?
  What does the sole SRE at a company do? Are there organisations without a need for SRE?
- What do SLIs look like for things that aren't stateless webapps?
- What does the rise of cloud providers and technologies mean for SRE?
- How do you SRE services where you don't have access to the code or can't make changes to it?

## Core Principles

This year we are introducing a Core Principles track. Talks in this track will focus on providing a deep understanding of how technologies we use everyday function and why it's important to know these details when supporting and scaling your infrastructure.

For this track, we're looking for a number of topics, such as:

- Databases (e.g. how is data stored on disk in MySQL, PostgreSQL, etc.?)
- Observability (e.g. monitoring overview, events vs. metrics, whitebox vs. blackbox, visualizations)
- Data Infrastructure (e.g. how does Hadoop work? What is MapReduce?)
- Distributed Systems (e.g. consistency and consensus)
- Network (e.g. HTTP routing and load balancing)
- Languages and performance (e.g. debugging systems with GDB)

# Agenda

usenix SRECON EUROPE MIDDLE EAST AFRICA

**Wireless Network Information**
SSID: CCDGuest
No password required

www.usenix.org/srecon19emea
#SREcon

## Wednesday

**07:45–08:45** Morning Coffee and Tea — The Forum

**08:45–10:30**

**Opening Plenary Session**
The Liffey

**Opening Remarks**
Program Co-Chairs: Emil Stolarsky, Incident Labs, and Murali Suriar, Google

**The SRE I Aspire to Be**
Yaniv Aknin, Google Cloud

**Opening Plenary**
Nancy Leveson, MIT

**10:30–11:00** Break with Refreshments — The Forum

**11:00–12:30**

| Track 1 (Core Principles) The Liffey B | Track 2 The Liffey A | Track 3 Liffey Hall 2 |
|---|---|---|
| **SLOs for Data-Intensive Services** Yoann Fouquet, Booking.com | **A Tale of Two Rotations: Building a Humane & Effective On-Call** Nick Lee, Uber | **Automating HA Deployments with BGP, IPv6, and Anycast** John Studarus, JHL Consulting LLC |
| **Latency SLOs Done Right** Heinrich Hartmann, Circonus | **Support Operations Engineering: Scaling Developer Products to the Millions** Junade Ali, Cloudflare | |
| **Building a Scalable Monitoring System** Molly Struve, Kenna Security | **The Unmonitored Failure Domain: Mental Health** Jaime Woo, Incident Labs | |

**12:30–14:00** Luncheon — The Forum

**14:00–15:30**

| Track 1 (Core Principles) The Liffey B | Track 2 The Liffey A | Track 3 Liffey Hall 2 | Track 4 Liffey Meeting Room 2 |
|---|---|---|---|
| **Control Theory for SRE** Ted Hahn, TCB Technologies, and Mark Hahn, Ciber Global | **Being Reasonable about SRE** Vitek Urbanec, Unity Technologies | **SRE Classroom, Or, How to Design a Reliable Distributed System in 3 Hours** Alex Perry, Google LLC, and Andrew Suffield, Goldman Sachs | **Implementing Distributed Consensus** Dan Lüdtke and Kordian Bruck, Google |
| **Eventually Consistent Service Discovery** Suhail Patel, Monzo | **From Nothing to SRE: Practical Guidance on Implementing SRE in Smaller Organisations** Matthew Huxtable, Sparx | | |
| **Network Monitor: A Tale of ACKnowledging an Observability Gap** Jason Gedge, Shopify | **My Life as a Solo SRE** Brian Murphy, G-Research | | |

**15:30–16:00** Break with Refreshments — The Forum

**16:00–17:30**

| Track 1 (Core Principles) The Liffey B | Track 2 The Liffey A | Track 3 Liffey Hall 2 |
|---|---|---|
| **Zero Touch Prod: Towards Safer and More Secure Production Environments** Michał Czapiński and Rainer Wolafka, Google Switzerland | **All of Our ML Ideas Are Bad (and We Should Feel Bad)** Todd Underwood, Google | **SRE Classroom, Or, How to Design a Reliable Distributed System in 3 Hours** Alex Perry, Google LLC, and Andrew Suffield, Goldman Sachs *(Continued from previous session)* |
| **Zero-Downtime Rebalancing and Data Migration of a Mature Multi-Shard Platform** Justin Li and Florian Weingarten, Shopify | **Fast, Available, Catastrophically Failing? Safely Avoiding Behavioral Incidents in Complex Production Systems** Ramin Keene, fuzzbox.io | |

**17:30–18:30** Social Hour, *Sponsored by Microsoft Azure* — The Forum

## Thursday

**08:00–09:00** Morning Coffee and Tea, *Sponsored by Bloomberg* — The Forum

**09:00–10:30**

| Track 1 (Core Principles) The Liffey B | Track 2 The Liffey A | Track 3 Liffey Hall 2 |
|---|---|---|
| **Advanced Napkin Math: Estimating System Performance from First Principles** Simon Eskildsen, Shopify | **Deploying SRE Training Best Practices to Production: How We SRE'ed Our SRE Education Program** Jennifer Petoff, Google Ireland, and JC van Winkel, Google Switzerland | **Effective Distributed Tracing Workshop** Pedro Alves, Serbay Arslanhan, and Luis Mineiro, Zalando SE |
| **The Map Is Not the Territory: How SLOs Lead Us Astray, and What We Can Do about It** Narayan Desai, Google | **SRE by Influence, Not Authority: How the New York Times Prepares for Large Scale Events** Vinessa Wan and Brett Haranin, The New York Times | |

**10:30–11:00** Break with Refreshments — The Forum

**11:00–12:30**

| Track 1 (Core Principles) The Liffey B | Track 2 The Liffey A | Track 3 Liffey Hall 2 |
|---|---|---|
| **Load Balancing Building Blocks** Kyle Lexmond, Facebook | **Are We All on the Same Page? Let's Fix That** Luis Mineiro, Zalando | **Effective Distributed Tracing Workshop** Pedro Alves, Serbay Arslanhan, and Luis Mineiro, Zalando SE *(Continued from previous session)* |
| **What Happens When You Type en.wikipedia.org?** Effie Mouzeli and Alexandros Kosiaris, Wikimedia Foundation | **Weathering the Storm: How Early Warnings Save the Farm** Brian Sherwin, LinkedIn Corporation | |

**12:30–14:00** Luncheon, *Sponsored by Blameless* — The Forum

**14:00–15:30**

| Track 1 (Core Principles) The Liffey B | Track 2 The Liffey A | Track 3 Liffey Hall 2 | Track 4 Liffey Meeting Room 2 |
|---|---|---|---|
| **Refining Systems Data without Losing Fidelity** Liz Fong-Jones, honeycomb.io | **How to Do SRE When You Have No SRE** Joan O'Callaghan, Udemy | **Statistics for Engineers** Heinrich Hartmann, Circonus | **Managing Microservices with Istio Service Mesh** Rafik Harabi, Innovsquare |
| **Tracing Real-Time Distributed Systems** Evgeny Yakimov, Bloomberg LP | **One on One SRE** Amy Tobey | | |

**15:30–16:00** Break with Refreshments — The Forum

**16:00–17:30**

| Track 1 The Liffey B | Track 2 The Liffey A | Track 3 Liffey Hall 2 | Track 4 Liffey Meeting Room 2 |
|---|---|---|---|
| **A Customer Service Approach to SRE** John Looney, Facebook | **Prioritizing Trust While Creating Applications** Jennifer Davis, Microsoft | **Statistics for Engineers** Heinrich Hartmann, Circonus *(Continued from previous session)* | **Managing Microservices with Istio Service Mesh** Rafik Harabi, Innovsquare *(Continued from previous session)* |
| **SRE & Product Management: How to Level up Your Team (and Career!) by Thinking like a Product Manager** Jen Wohlner, Livepeer | **Software Patching Needn't Be a Can of Worms** Philip Rowlands | | |

**17:30–18:30** Lightning Talks — The Liffey B

**18:30–20:00** Conference Reception — Level 3 Foyer

## Friday

**08:00–09:00** Morning Coffee and Tea — The Forum

**09:00–10:30**

| Track 1 The Liffey B | Track 2 The Liffey A | Track 3 Liffey Hall 2 |
|---|---|---|
| **Bigtable: A Journey from Binary to Service and the Lessons Learned along the Way** Brendan Gleason and Gaurav Prabhu Gaonkar, Google | **Building Resilience: How to Learn More from Incidents** Nick Stenning, Microsoft | **What I Wish I Knew before Going On-Call** Chie Shu and Dorothy Jung, Yelp |
| **SDKs Are Not Services and What This Means for SREs** Justin Coffey, Criteo | **How Stripe Invests in Technical Infrastructure** Will Larson, Stripe | |

**10:30–11:00** Break with Refreshments — The Forum

**11:00–12:30**

| Track 1 The Liffey B | Track 2 The Liffey A | Track 3 Liffey Hall 2 |
|---|---|---|
| **Why Automating Everything Adds to Your Toil** Colin Thorne and Cameron McCallister, IBM | **Pushing through Friction** Dan Na, Squarespace | **Unconference: Unsolved Problems in SRE** Kurt Andersen, LinkedIn |
| **Autopsy of a MySQL Automation Disaster** Jean-François Gagné, MessageBird | **Perks and Pitfalls of Building a Remote First Team** Ryan Neal, Netlify | |

**12:30–14:00** Luncheon — The Forum

**14:00–15:30**

| Track 1 The Liffey B | Track 2 The Liffey A |
|---|---|
| **Expect the Unexpected: Preparing SRE Teams for Responding to Novel Failures** John Arthorne, Shopify | **Evolution of Observability Tools at Pinterest** Naoman Abbas, Pinterest |
| **Hiring Great SREs** Brian Rutkin, Twitter, Inc. | **How to SRE When Everything's Already on Fire** Alex Hidalgo and Alex Lee, Squarespace |
| **SRE in the Third Age** Björn Rabenstein, Grafana Labs | |

**15:30–16:00** Break with Refreshments — Level 1 Foyer

**16:00–17:35**

**Closing Plenary Session**
The Liffey

**Fault Tree Analysis Applied to Apache Kafka**
Andrey Falko, Lyft

**Applicable and Achievable Formal Verification**
Heidy Khlaaf, Adelard LLP

**Closing Remarks**
Program Co-Chairs: Emil Stolarsky, Incident Labs, and Murali Suriar, Google

# Agenda

Pavlos

Ingo

usenix SRE CON EUROPE MIDDLE EAST AFRICA

**Wireless Network Information**
SSID: CCDGuest
No password required

www.usenix.org/srecon19emea
#SREcon

## Wednesday

**07:45–08:45** Morning Coffee and Tea — The Forum

**08:45–10:30**

**Opening Plenary Session** — The Liffey
**Opening Remarks** — Program Co-Chairs: Emil Stolarsky, Incident Labs, and Murali Suriar, Google
**The SRE I Aspire to Be** — Yaniv Aknin, Google Cloud
**Opening Plenary** — Nancy Leveson, MIT

**10:30–11:00** Break with Refreshments — The Forum

**11:00–12:30**

| Track 1 (Core Principles) — The Liffey B | Track 2 — The Liffey A | Track 3 — Liffey Hall 2 |
|---|---|---|
| **SLOs for Data-Intensive Services** — Yoann Fouquet, Booking.com | **A Tale of Two Rotations: Building a Humane & Effective On-Call** — Nick Lee, Uber | **Automating HA Deployments with BGP, IPv6, and Anycast** — John Studarus, JHL Consulting LLC |
| **Latency SLOs Done Right** — Heinrich Hartmann, Circonus | **Support Operations Engineering: Scaling Developer Products to the Millions** — Junade Ali, Cloudflare | |
| **Building a Scalable Monitoring System** — Molly Struve, Kenna Security | **The Unmonitored Failure Domain: Mental Health** — Jaime Woo, Incident Labs | |

**12:30–14:00** Luncheon — The Forum

**14:00–15:30**

| Track 1 (Core Principles) — The Liffey B | Track 2 — The Liffey A | Track 3 — Liffey Hall 2 | Track 4 — Liffey Meeting Room 2 |
|---|---|---|---|
| **Control Theory for SRE** — Ted Hahn, TCB Technologies, and Mark Hahn, Ciber Global | **Being Reasonable about SRE** — Vitek Urbanec, Unity Technologies | **SRE Classroom, Or, How to Design a Reliable Distributed System in 3 Hours** — Alex Perry, Google LLC, and Andrew Suffield, Goldman Sachs | **Implementing Distributed Consensus** — Dan Lüdtke and Kordian Bruck, Google |
| **Eventually Consistent Service Discovery** — Suhail Patel, Monzo | **From Nothing to SRE: Practical Guidance on Implementing SRE in Smaller Organisations** — Matthew Huxtable, Sparx | | |
| **Network Monitor: A Tale of ACKnowledging an Observability Gap** — Jason Gedge, Shopify | **My Life as a Solo SRE** — Brian Murphy, G-Research | | |

**15:30–16:00** Break with Refreshments — The Forum

**16:00–17:30**

| Track 1 (Core Principles) — The Liffey B | Track 2 — The Liffey A | Track 3 — Liffey Hall 2 |
|---|---|---|
| **Zero Touch Prod: Towards Safer and More Secure Production Environments** — Michał Czapiński and Rainer Wolafka, Google Switzerland | **All of Our ML Ideas Are Bad (and We Should Feel Bad)** — Todd Underwood, Google | **SRE Classroom, Or, How to Design a Reliable Distributed System in 3 Hours** — Alex Perry, Google LLC, and Andrew Suffield, Goldman Sachs |
| **Zero-Downtime Rebalancing and Data Migration of a Mature Multi-Shard Platform** — Justin Li and Florian Weingarten, Shopify | **Fast, Available, Catastrophically Failing? Safely Avoiding Behavioral Incidents in Complex Production Systems** — Ramin Keene, fuzzbox.io | *(Continued from previous session)* |

**17:30–18:30** Social Hour, *Sponsored by Microsoft Azure* — The Forum

## Thursday

**08:00–09:00** Morning Coffee and Tea, *Sponsored by Bloomberg* — The Forum

**09:00–10:30**

| Track 1 (Core Principles) — The Liffey B | Track 2 — The Liffey A | Track 3 — Liffey Hall 2 |
|---|---|---|
| **Advanced Napkin Math: Estimating System Performance from First Principles** — Simon Eskildsen, Shopify | **Deploying SRE Training Best Practices to Production: How We SRE'ed Our SRE Education Program** — Jennifer Petoff, Google Ireland, and JC van [illegible], Google Switzerland | **Effective Distributed Tracing Workshop** — Pedro Alves, Serbay Arslanhan, and Luis Mineiro, Zalando SE |
| **The Map Is Not the Territory: How SLOs Lead Us Astray, and What We Can Do about It** — Narayan Desai, Google | **SRE by Influence, Not Authority: How the New York Times Prepares for Large Scale Events** — Vinessa Wan and Brett Haranin, The New York Times | |

**10:30–11:00** Break with Refreshments — The Forum

**11:00–12:30**

| Track 1 (Core Principles) — The Liffey B | Track 2 — The Liffey A | Track 3 — Liffey Hall 2 |
|---|---|---|
| **Load Balancing Building Blocks** — Kyle Lexmond, Facebook | **Are We All on the Same Page? Let's Fix That** — Luis Mineiro, Zalando | **Effective Distributed Tracing Workshop** — Pedro Alves, Serbay Arslanhan, and Luis Mineiro, Zalando SE |
| **What Happens When You Type en.wikipedia.org?** — Effie Mouzeli and Alexandros Kosiaris, Wikimedia Foundation | **Weathering the Storm: How Early Warnings Save the Farm** — Brian Sherwin, LinkedIn Corporation | *(Continued from previous session)* |

**12:30–14:00** Luncheon, *Sponsored by Blameless* — The Forum

**14:00–15:30**

| Track 1 (Core Principles) — The Liffey B | Track 2 — The Liffey A | Track 3 — Liffey Hall 2 | Track 4 — Liffey Meeting Room 2 |
|---|---|---|---|
| **Refining Systems Data without Losing Fidelity** — Liz Fong-Jones, honeycomb.io | **How to Do SRE When You Have No SRE** — Joan O'Callaghan, Udemy | **Statistics for Engineers** — Heinrich Hartmann, Circonus | **Managing Microservices with Istio Service Mesh** — Rafik Harabi, Innovsquare |
| **Tracing Real-Time Distributed Systems** — Evgeny Yakimov, Bloomberg LP | **One on One SRE** — Amy Tobey | | |

**15:30–16:00** Break with Refreshments — The Forum

**16:00–17:30**

| Track 1 — The Liffey B | Track 2 — The Liffey A | Track 3 — Liffey Hall 2 | Track 4 — Liffey Meeting Room 2 |
|---|---|---|---|
| **A Customer Service Approach to SRE** — John Looney, Facebook | **Prioritizing Trust While Creating Applications** — Jennifer Davis, Microsoft | **Statistics for Engineers** — Heinrich Hartmann, Circonus *(Continued from previous session)* | **Managing Microservices with Istio Service Mesh** — Rafik Harabi, Innovsquare *(Continued from previous session)* |
| **SRE & Product Management: How to Level up Your Team (and Career!) by Thinking like a Product Manager** — Jen Wohlner, Livepeer | **Software Patching Needn't Be a Can of Worms** — Philip Rowlands | | |

**17:30–18:30** Lightning Talks — The Liffey B

**18:30–20:00** Conference Reception — Level 3 Foyer

## Friday

**08:00–09:00** Morning Coffee and Tea — The Forum

**09:00–10:30**

| Track 1 — The Liffey B | Track 2 — The Liffey A | Track 3 — Liffey Hall 2 |
|---|---|---|
| **Bigtable: A Journey from Binary to Service and the Lessons Learned along the Way** — Brendan Gleason and Gaurav Prabhu Gaonkar, Google | **Building Resilience: How to Learn More from Incidents** — Nick Stenning, Microsoft | **What I Wish I Knew before Going On-Call** — Chie Shu and Dorothy Jung, Yelp |
| **SDKs Are Not Services and What This Means for SREs** — Justin Coffey, Criteo | **How Stripe Invests in Technical Infrastructure** — Will Larson, Stripe | |

**10:30–11:00** Break with Refreshments — The Forum

**11:00–12:30**

| Track 1 — The Liffey B | Track 2 — The Liffey A | Track 3 — Liffey Hall 2 |
|---|---|---|
| **Why Automating Everything Adds to Your Toil** — Colin Thorne and Cameron McCallister, IBM | **Pushing through Friction** — Dan Na, Squarespace | **Unconference: Unsolved Problems in SRE** — Kurt Andersen, LinkedIn |
| **Autopsy of a MySQL Automation Disaster** — Jean-François Gagné, MessageBird | **Perks and Pitfalls of Building a Remote First Team** — Ryan Neal, Netlify | |

**12:30–14:00** Luncheon — The Forum

**14:00–15:30**

| Track 1 — The Liffey B | Track 2 — The Liffey A |
|---|---|
| **Expect the Unexpected: Preparing SRE Teams for Responding to Novel Failures** — John Arthorne, Shopify | **Evolution of Observability Tools at Pinterest** — Naoman Abbas, Pinterest |
| **Hiring Great SREs** — Brian Rutkin, Twitter, Inc. | **How to SRE When Everything's Already on Fire** — Alex Hidalgo and Alex Lee, Squarespace |
| **SRE in the Third Age** — Björn Rabenstein, Grafana Labs | |

**15:30–16:00** Break with Refreshments — Level 1 Foyer

**16:00–17:35**

**Closing Plenary Session** — The Liffey
**Fault Tree Analysis Applied to Apache Kafka** — Andrey Falko, Lyft
**Applicable and Achievable Formal Verification** — Heidy Khlaaf, Adelard LLP
**Closing Remarks** — Program Co-Chairs: Emil Stolarsky, Incident Labs, and Murali Suriar, Google

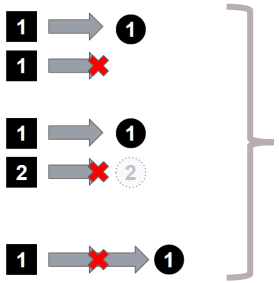
# The SRE I aspire to be

Yaniv Aknin @aknin, Google

*Apply Engineering principles to improve reliability, balance with innovation. Tie measurement to business / project priorities.*

Engineer: “using scientific principles to design and build $things” For SRE: $things = reliability
Measure=operationalize, but what is the right measure, the right measurement ?
**Measurably optimize reliability vs. cost**

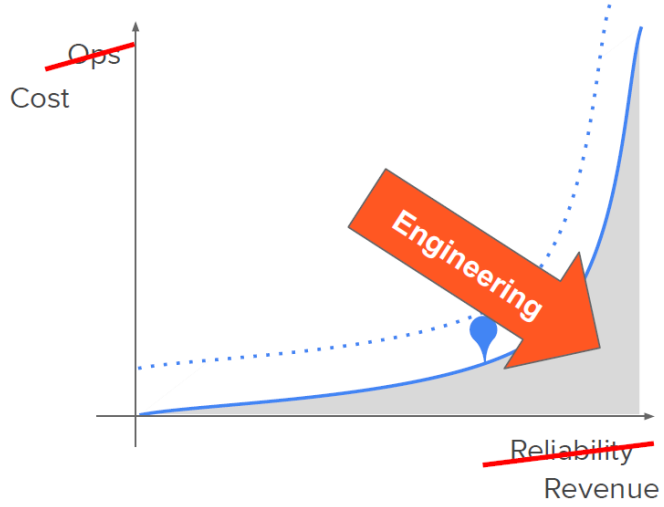
The modest SRE Toolbox
- Trade cost - redundant resource
- Trade quality - degraded results
- Trade latency - retry transient failures

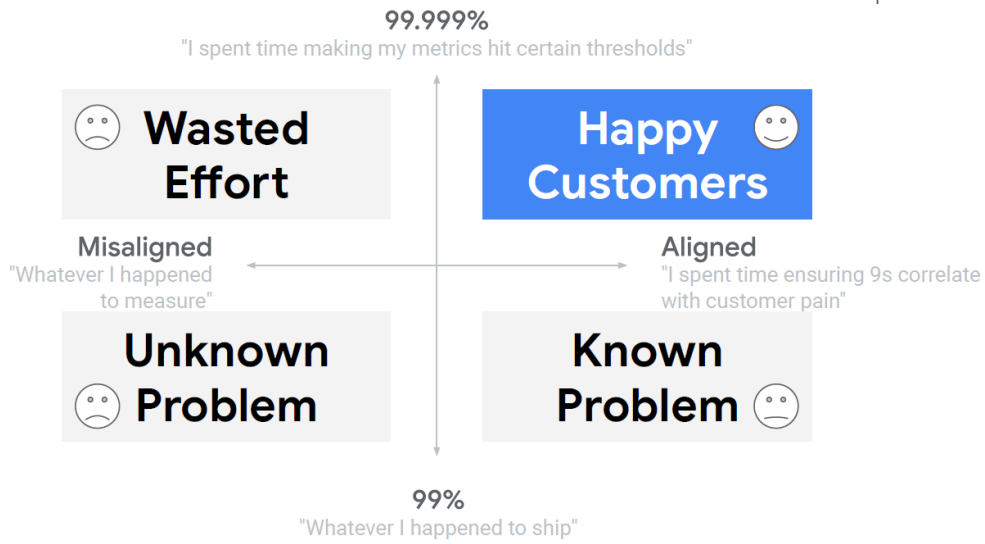



Compound/Advanced Patterns

| | | | |
|---|---|---|---|
| Waterfall | Jitter | Breaker | Infra as code |
| Partitioning | Sidecar | Fail static | Self-healing |

**Tension: Innovation vs. Reliability**
”Error Budget”




The SRE I aspire to be
- Have a measurement of reliability
- Measurement is tied to project priorities
- Ops work is tied to the measurement

# Being reasonable about SRE

Vitek Urbanec, Unity

*SRE adoption can be challenging when done out of context. Reliability is about motivation.*

Adopting SRE: check-in-the-box and buzzword driven adoption
But
- out of context
- does it fit the culture ?

Risk: same team, skills, culture, cooler name, higher expectations
**Shifting from ops to SRE needs time and effort**
There is nothing wrong with ops - if it is working for you

What makes it tough:
- SREs need to level-up soft skills
- SREs need to understand app development
- SRE thrives a “special” culture

**Want to be reasonable about SRE?**
**- Learn and get educated**
**- Build inclusive attitude**
**- Treat tooling as a product**
**- Look for value to provide, not a box to fit into**

# SRE in the Third Age

Björn Rabenstein, Grafana Labs

*A look into the future of SRE.*


**SRE Ages**

| 1st age (2003-2014) | 2nd age (2014-Now) | 3rd age |
|---|---|---|
| SRE was proprietary to Google | SRE became a well-known discipline in the tech community, including books and conferences | Hasn't begun yet |

*In the 3rd age…*
You won't need SREs.
You will need SRE.

*Recruiting in the 3rd age…*
Don't look for SREs.
Look for SRE mindsets.

*In the 3rd age…*
The whole SRE layer is even thinner, so it will be easy to make this part of every engineer's curriculum.

SRE will naturally spread until it's everywhere.
You'll always act in an SRE-spirit, even after transitioning into a different role.

# Deploying SRE Training Best Practices to Production

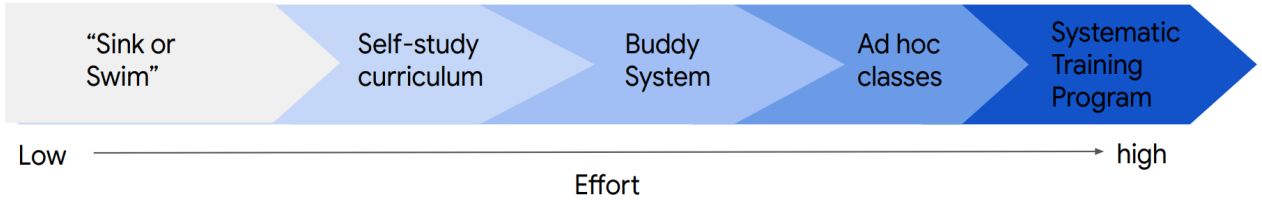
Jennifer Petoff @jennski & JC van Winkel, Google


*Behind the scenes of the SRE EDU Orientation curriculum at Google. SRE training best practices.*

## SRE trainings

- build confidence and reduce imposter syndrome
- are not about a fire hose of information
- offer hands on exercises

## Continuum of Training Options




## Tips

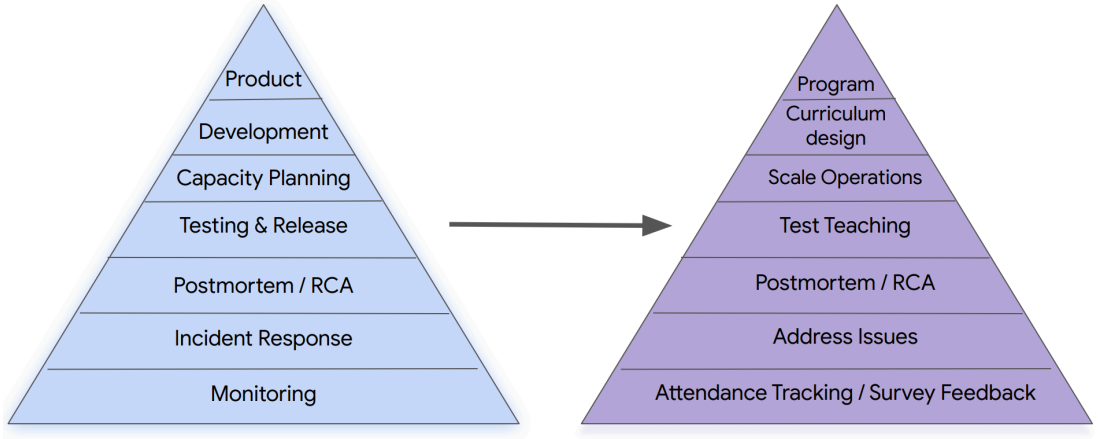
- Avoid "Sink or Swim": breeds stress and frustration
- Move away from passive listening
- Instill confidence
- Troubleshoot a real system, built for this purpose

## Adapting for Small Companies

- Probably no classes, but self directed and hands on exercises
- Hands on in an environment that looks like a production environment
- Have a script that breaks things
- Plausible story for breakage

The Service Reliability Hierarchy provides a useful framework for building and running an SRE training program

How to Apply SRE Principles to a Training Program




Service Reliability Hierarchy*

*SRE Training* Reliability Hierarchy

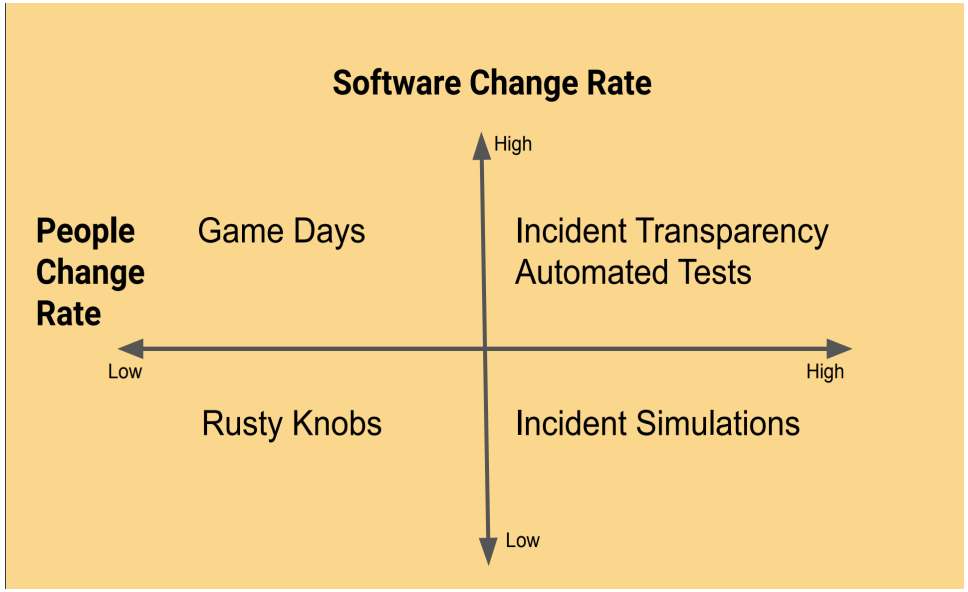
* https://landing.google.com/sre/sre-book/chapters/part3/

# Expect the Unexpected: Preparing SRE Teams for Responding to Novel Failures

John Arthorne @jarthorne, Shopify

*Preparing for truly unexpected failures.*

Deliberate practice makes incidents more comfortable; how do we practice unpredictable?




**Transparent Response**
- Shadowing
- Transparent decision making
- Senior staff leading by example

**Turn Rusty Knobs**
- Exercise failure recovery practices
- Builds confidence

**Incident Simulation**
- Wheel of Misfortune
- Only as good as existing human understanding of the system

**Automated Failure Testing**
- Focus on most routine failures (Timeouts, connection failures)
- Can't validate full system behavior

**Game Days**
- Create a hypothesis of system behavior
- Include real production failure
- Observe, Recover, Adapt


https://github.com/jarthorn/lego-incident-response

# Pushing through friction
## Dan Na @dxna, Squarespace

*Willingness to accept friction. Take the correct path, even if it's is hard, it ultimately leads to better outcome.*

Friction? **Gap between how things are, and how things should be**
- Code base with no owner
- No answer (for a question on Slack)
- Siloed team, no on-boarding, no diversity
- No convenient answer to move forward

Friction is never intentional
- Company growth (mostly midsize companies)
- Scale the product, scale the company
- Organization and processes incur friction slowly

The **normalization of deviance** is when deviant behavior becomes the norm.

**To anyone outside of your organization it's obvious that what you're doing doesn't make sense, but to those inside the organization it's normal and standard procedure.**

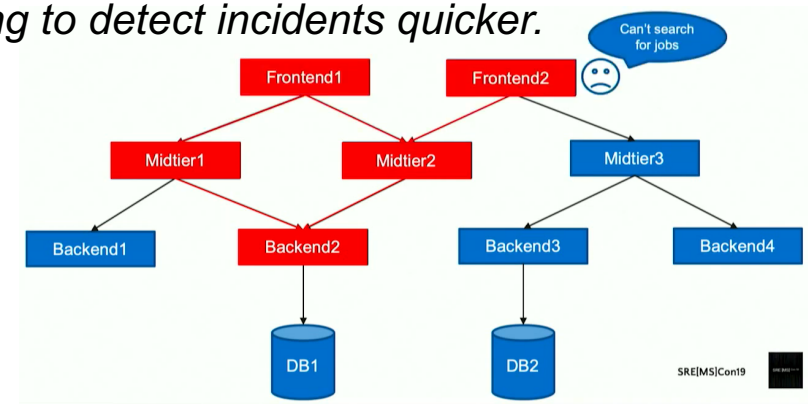
**Organization**
- ✓ Document single sources of truth and keep updates
- ✓ Adopt processes to vet technology decisions
- ✓ Long-term cultural behaviors
- ✓ Address hard truths, kindly
- ✓ Make glue-work mandatory for promotion
- ✓ Make psychological safety paramount

**Individuals**
- ✓ Develop you own sense of agency
- ✓ Intrinsic motivation: Autonomy, mastery, purpose
- ✓ Being a hero, or an asshole, doesn't scale
- ✓ Have important discussions face-to-face
- ✓ Get to know other people on other teams and in other orgs
- ✓ New idea? Try it once.

Normalization of deviance https://danluu.com/wat/
Being Glue: Noidea.dog/glue

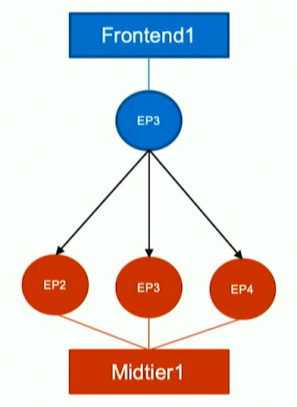
# How early warnings save the farm
## Brian Sherwin, LinkedIn

*Alert correlation platform, based on relationship model & near-time latency monitoring to detect incidents quicker.*




Monitoring in a microservice world
- Traverse relationship - between endpoints, to provide context
- Auto threshold for latency (mitigating false-positive through statistics)

Alert correlation platform
- Proactive escalations
- Near time monitoring (fast detection)
- Reactive identification
- Corroborating evidence
- Experience (confidence)




Design considerations
- Accuracy (no false negatives)
- Speed (time to give recommendation)
- Scalable (endpoints come and go all the time)
- Simplicity (no extra data required, or provided)
- Reusable

Results
- 90% incident detection (which dependency is broken)
- Catching hidden issues (not everything was monitored before)

Lessons learned
- Speed matters (pre-calculating tree)
- Scale of ingestion
- Hierarchy helps (call tree, traces and metrics)
- Validation rules; Accuracy shines; consider Deployment activity
- Evidence speaks
- Adoption reflects (promote, find out why not using)
- History repeats (store the history)

My philosophy on alerting
http://files.catwell.info/misc/mirror/rob-ewaschuk-google-sre-philosophy-alerting.pdf

# Zero Touch Prod: Towards Safer and More Secure Production Environments

Michał Czapiński and Rainer Wolafka, Google

*An approach towards making production safer and prevent outages.*

- Humans make mistakes repeatedly
- Follow a set of principles to enforce production safety practices
- Provide a framework to assess and track compliance

**Zero Touch Prod (ZTP)**

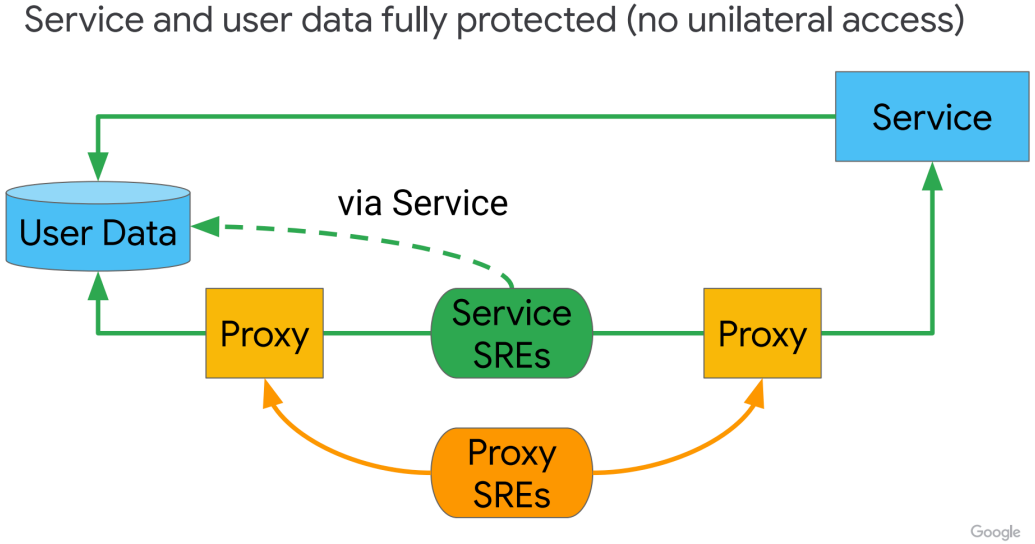
- Every change in prod must either be:
  - Made by automation (no humans)
  - Prevalidated by software
  - Made via audited break-glass mechanism

**Reliable Automation**

- Limiting Privilege: Authority Delegation
- Enforce safety policies: Safety Checks
- Controlling the rate of change: Rate Limiting

**Safe Proxies**

- Full audit log (who, when, what, why)
- Fine-grained authorisation
- Rate-limiting
- Removes unilateral privileged access
  - accidental production change
  - unauthorized access to user data



# Why automating everything adds to your toil

Colin Thorne @ColinJThorne & Cam McAllister, IBM

*Automation is Good! Toil is Bad. Reduce the toil caused by automation.*

**Toil:** Gets in the way of making progress. Repetitive manual tasks (Incidents, tickets, watching dashboards)

The key is to reduce the amount of toil.

**Automation:** Avoid manual tasks by getting computers to do it for us (chatbots, self-healing, deploying, self service)

Automation rots over time just like any code, automation needs constant care and feeding:

- Dependencies change
- Requirements change
- SREs change
- Production systems change
- Languages change

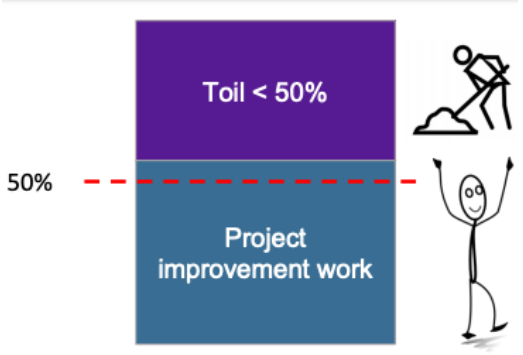
**"Ironically, although intended to relieve SREs of work, automation adds to systems' complexity and can easily make that work even more difficult"** [Seeking SRE, John Allspaw and Richard Cook]

Challenges

- **Unused automation:** Automation written once, but no one uses it
- **Duplicate automation:** Not invented here leads to duplicate automation
- **Too many tools**: The more tools you have, the more you have to maintain, the less they are used

Reduce toil produced by automation

- Build as a developer
- Maximise use of your automation
- Treat your automation as evolutionary steps



# How stripe invests in technical infrastructure

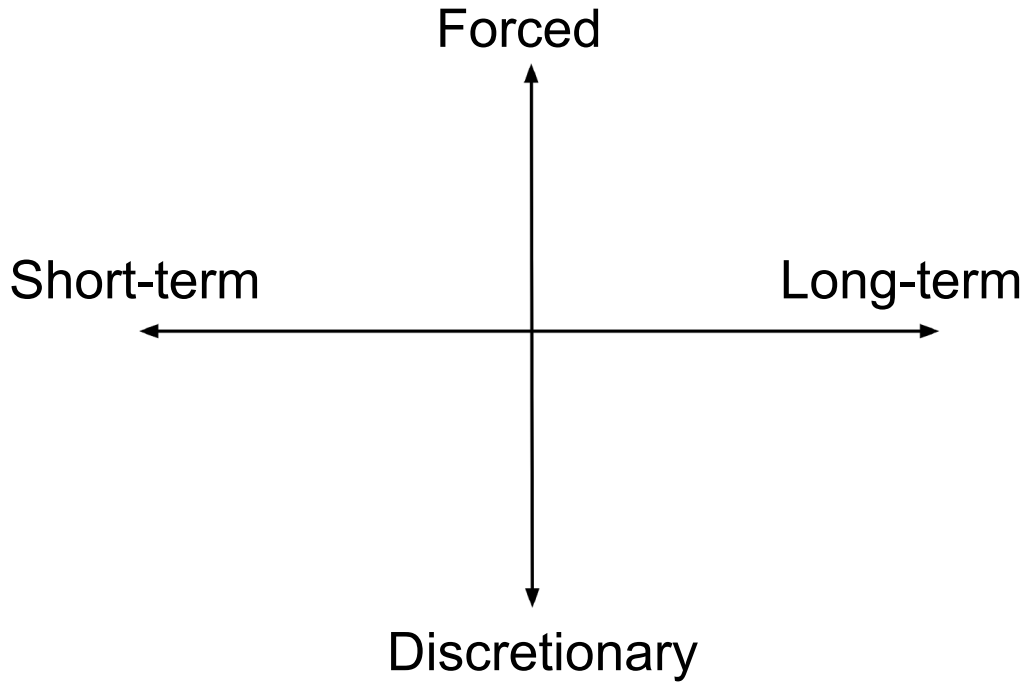
Will Larson @lethain, Stripe

*Prioritizing infrastructure investment … in a high autonomous environment … within a rapidly scaling business.*

**Escaping the firefight**
Forced: scale mongodb, lower AWS costs, GDPR
Discretionary: server to service, deep learning
Short-term: critical remediation, hit budget, support launch
Long-term: QoS strategy, “bend the cost curve”, rewrite a monolith

**Approach**
Reduce concurrent work, finish something useful
Eliminate categories of problems
Seeing signs of progress? If not: extend the size of the team
Once there is progress, stay the course

**Problems**:
- Making the most obvious solution
- Fixation on the local maxima
- Benchmarking with peer companies
- Infinite problems – what to pick: Prioritizing order by ROI, together with users
- Right opportunity – wrong solution: validate the approach
  (cheaply disprove the approach; try hardest cases early)




**Unifying approach:**
40% user asks
30% platform quality
30% key initiatives

# Latency SLOs Done Right

Heinrich Hartmann @heinrichhartmann, Circonus

*Percentile Metrics can't be used for SLOs*

For SLOs we need to compute percentiles over ...
- multiple weeks of data
- multiple nodes (potentially).

But: **Percentiles can't be aggregated.**

HDR Histogram Metrics allow you to easily calculate arbitrary Latency SLOs.

**Task**

*Count all requests over $period served faster than $threshold.*

**Three valid methods:**
- Log data
- Counter Metrics
- Histogram Metrics

***Log Data***
- Correct, clean, easy,
- BUT you need to keep all your log data for months ($$)
- *ssh+awk, ELK, Splunk, Honeycomb*

***Counter Metrics***
- Easy, correct, cost-effective, flexible in choosing intervals
- BUT you need to choose thresholds upfront
- *Prometheus ("Histograms"), Graphite, DataDog, VividCortext*

***Histogram Metrics***
- *Full flexibility in choosing thresholds and aggregation intervals, cost-effective*
- *BUT needs HDR histogram instrumentation*
- *Circonus, IronDB + Graphite / Grafana, Google internal tooling*

# Tracing Real-Time Distributed Systems

Evgeny Yakimov, Bloomberg

*Insights (and tradeoffs) when deploying distributed tracing at scale.*

100 billion market data “ticks” processed daily
Tracing: Custom library implementation based on OpenTracing, own agents and distribution; Jaeger to visualize

Challenges

- **Data size** (1k per span -> 500M spans per day; 30day storage -> 15B spans (@ $20k)
- **Message Fan-Out** (broadcast)
  Late stage filtering (up to 80% discard)
  Redundancy /hot / warm replicas)
  Result in noisy traces
  Solution: Cancel the Span collection
- **Splitting Messages**
  Multi-part messages can take different paths
  Solution: create new spans, ”dispatch” spans
- **Message conflation**
  Multiple upstream sources, high rate of messages
  Often only last value relevant
  Solution: Use “conflation” spans
- **Increasing Granularity**
  Spans are expensive
  Solution: Span.like tag semantics: TimeSpans, CheckPoints
- **Sampling**
  Head-based (trace creation time), Unitary (specific components)
  Solution: Tail-based approach

# A systems approach to Safety and Cybersecurity
## Nancy Leveson, MIT

*Use Systems Theory to treat safety as a control problem, not a failure problem. Build Safety.*

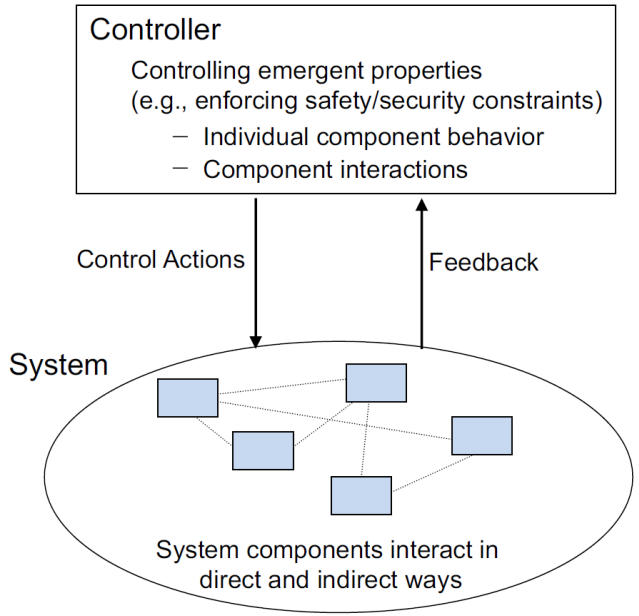
Accident = Loss of life, property damage, environmental pollution, mission
Human error is a symptom of a system that needs to be redesigned.
Traditional approach: Divide into separate parts, Analyze pieces separately and combine results

**Systems theory – a Systems Theoretic View of Safety and Security**
Too complex for complete analysis
Too organized for statistics

Focuses on **systems taken as a whole**, not on parts taken separately
**Emergent properties** (arise from complex interactions): Safety and security
Controller controls emergent properties through **actions and feedback**

STAMP: system-theoretic accident model and process

Building safety, not just measuring; Focus on preventing hazardous state
Safety prevent losses due to unintentional actions by benevolant actors
Security prevent losses due to intentional actions by malevolant actors




Info: http://psas.scripts.mit.edu (papers, presentations from conferences,
Engineer a Safe World: http://mitpress.mit.edu/books/engineering-safer-world
STPA Handbook http://psas.scripts.mit.edu
CAST Handbook http://sunnyday.mit.edu/CAST-Handbook.pdf

# References and Links

All presentations/video/voice available at
https://www.usenix.org/conference/srecon19emea/program

Other interesting talks

- All of Our ML Ideas Are Bad (and We Should Feel Bad)
- Load Balancing Building Blocks
- A Customer Service Approach to SRE

Some summary blogs:

- https://making.pusher.com/hot-sre-trends-in-2019/
- https://www.linkedin.com/pulse/look-back-srecon-emea-2019-bastian-spanneberg/

Misc:

- https://github.com/jarthorn/lego-incident-response
- https://github.com/dastergon/awesome-sre
- https://dastergon.gr/wheel-of-misfortune/

Twitter: #srecon https://twitter.com/hashtag/srecon