Recap
SREcon24 Americas

Ingo Averdunk
Distinguished Engineer
IBM

averdunk@de.ibm.com
@ingoa
/in/ingoaverdunk

IBM

TL:DR;

- Theme of SREcon 2024 Americas: You have to make Tradeoff’s
- 600+ participants, 229 companies, 24 countries

- SRE seems like “one domain”, but it covers a lot of ground. SREcon provides a variety of topics. Lots of discussions on evolving and improving SRE practices
- You have to make Trade-offs: efficiency vs thoroughness; overlapping and/or conflicting objectives; during incidents; across accuracy, overhead, scalability; Costs and benefits of tradeoffs may be unevenly distributed.
- Some recurring key themes
  - Observability continues to evolve
  - Costs matter (cloud spend, cardinality, open source, etc. especially as consequence to end of interest rate)
  - Socio-technical systems
  - Systems Thinking

- This presentation is like “Speed-Dating” - a super-condensed summary of 24 hrs / 1.000+ slides into a 45min session.
  It is meant to be a teaser, to motivate people listening to the recordings 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. SREcon strives to challenge both those new to the profession as well as those who have been involved in it for decades. The conference has a culture of critical thought, deep technical insights, continuous improvement, and innovation.

- 10th Anniversary of SREcon

- 600+ participants / 229 companies / 24 countries

- Theme of SREcon 2024: **You have to make Trade-offs**: efficiency vs thoroughness; overlapping and/or conflicting objectives; during incidents; across accuracy, overhead, scalability
  - Tradeoff decisions are technical, organizational, and social (often implicit and unstated)
  - Tradeoff decisions are considered and managed differently across roles and levels
  - Tradeoff decisions cross boundaries
  - Costs and benefits of tradeoffs may be unevenly distributed
  - Tradeoff decisions evolve over time
  - Some goals and priorities get trashed along the way

Hard Choices, Tight Timelines:
A Closer Look at Skip-level Tradeoff Decisions during Incidents
Dr. Laura Maguire and Courtney Nash, The VOID

Sponsors

PLATINUM SPONSOR

FireHydrant

GOLD SPONSORS

cortex

IBM

incident.io

mezmo

SENTRY

SILVER SPONSORS

APPTIO

Bloomberg

CITADEL | Securities

DISCOVER

EDGE DELTA

honeycomb.io

Jane Street

Meta

NOBL9

OBSERVE

splunk>

ManageEngine Site24x7

stanza

steadybit

Transposit

usenix
THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

BRONZE SPONSORS

antithesis

bytebridge

CHECK POINT

Checkly

dagknows

elastic

Firefly

Grafana Labs

Last9

logz.io

middleware

new relic

occams hub

rootly

SigLens

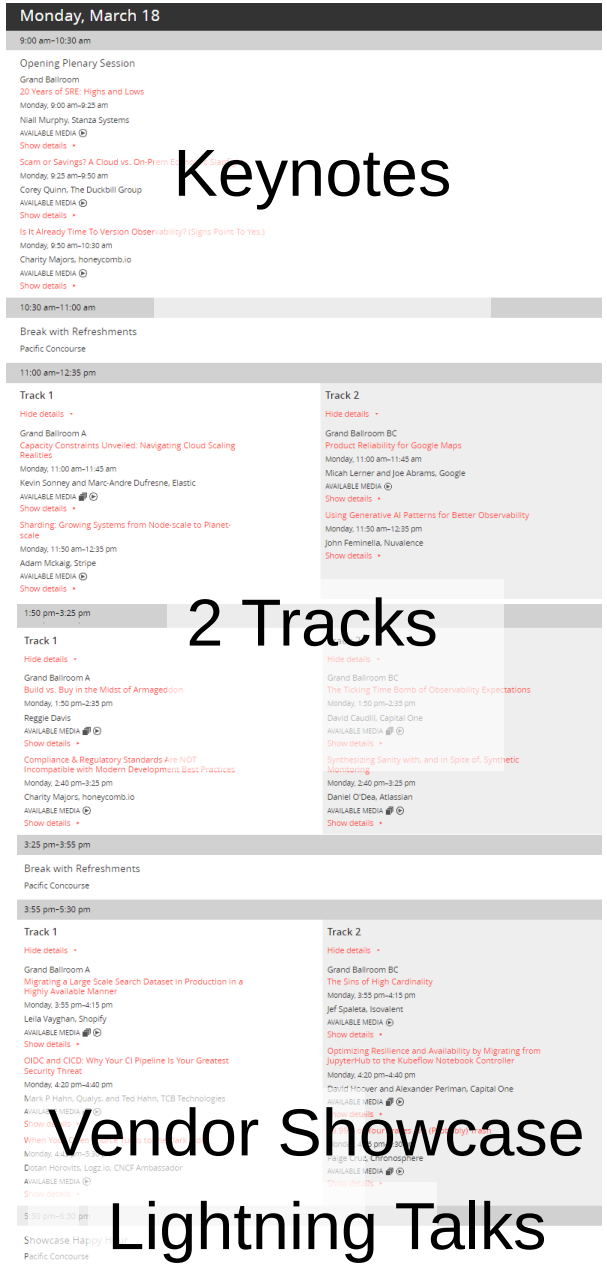
OPEN ACCESS SPONSOR

Google

MEDIA SPONSOR

no starch press

Agenda at a glance

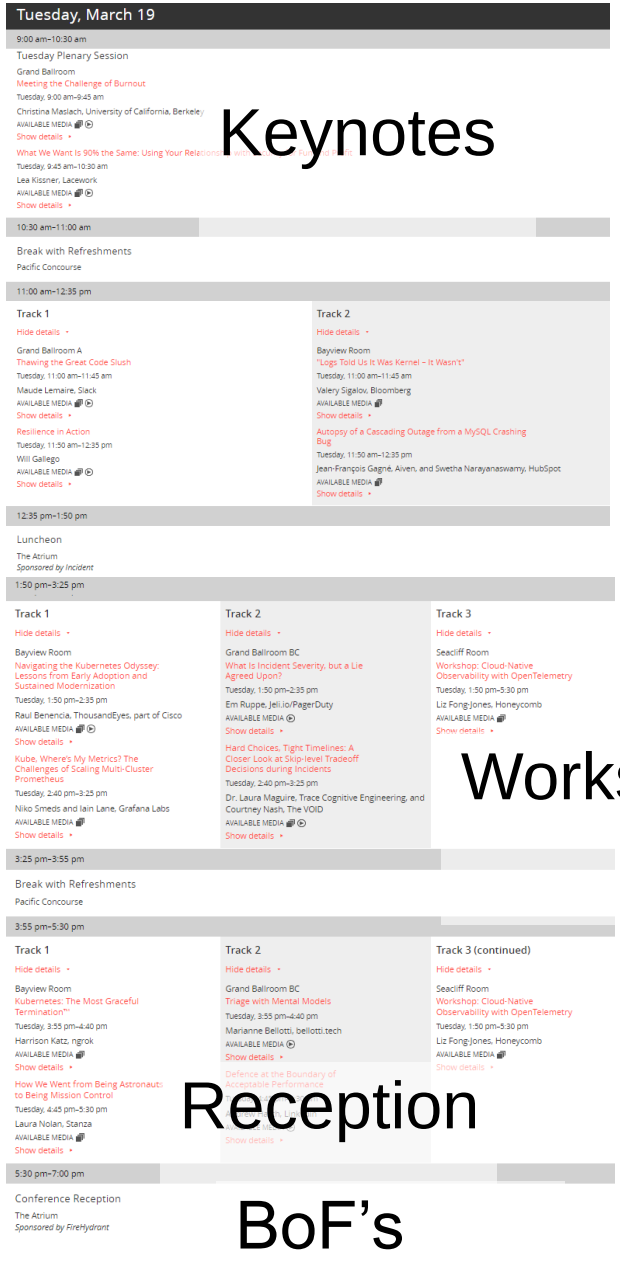


Keynotes

2 Tracks

Vendor Showcase

Lightning Talks

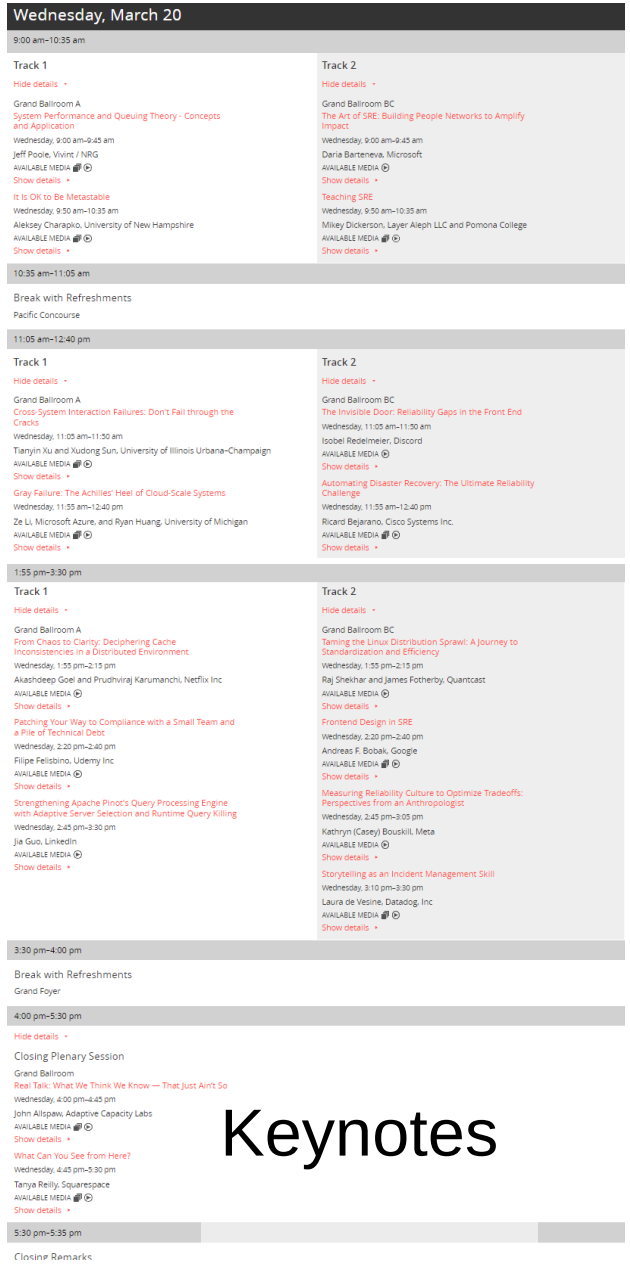


Keynotes

Workshop

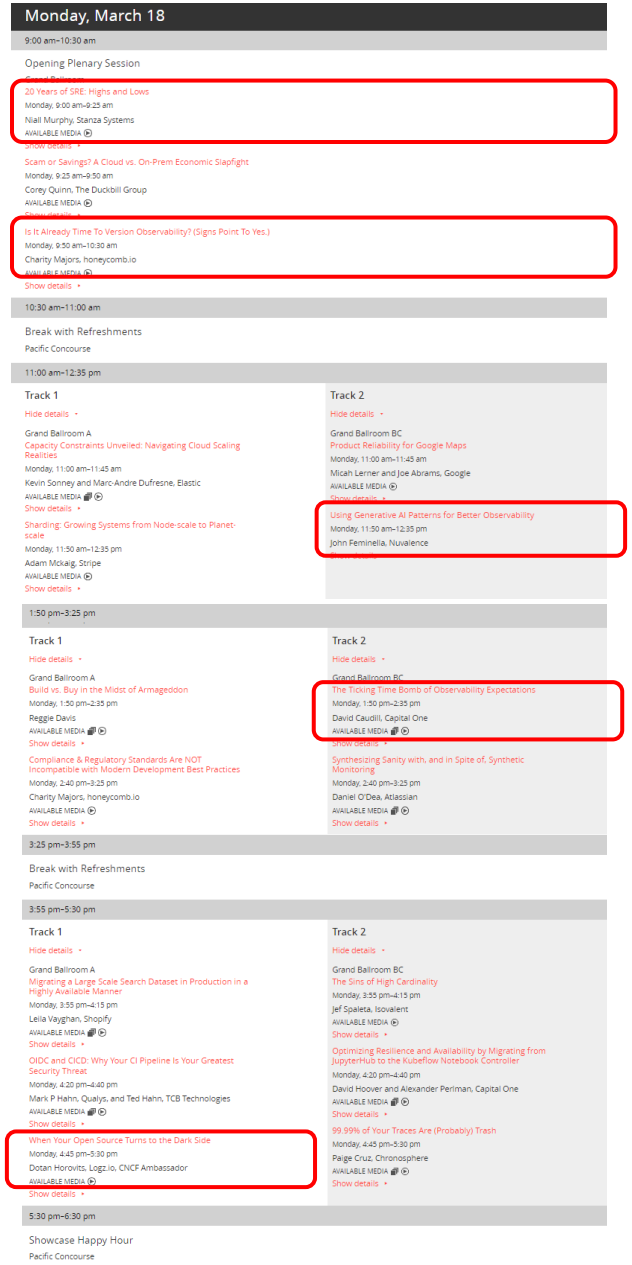
Reception

BoF's



Keynotes

Agenda at a glance



Monday, March 18

9:00 am–10:30 am

Opening Plenary Session

20 Years of SRE: Highs and Lows
Monday, 9:00 am–9:25 am
Niall Murphy, Stanza Systems
AVAILABLE MEDIA
Show details

Scam or Savings? A Cloud vs. On-Prem Economic Slapfight
Monday, 9:25 am–9:50 am
Corey Quinn, The Duckbill Group
AVAILABLE MEDIA
Show details

Is It Already Time To Version Observability? (Signs Point To Yes.)
Monday, 9:50 am–10:30 am
Charity Majors, honeycomb.io
AVAILABLE MEDIA
Show details

10:30 am–11:00 am

Break with Refreshments
Pacific Concourse

11:00 am–12:35 pm

Track 1
Hide details
Grand Ballroom A
Capacity Constraints Unveiled: Navigating Cloud Scaling Realities
Monday, 11:00 am–11:45 am
Kevin Sonney and Marc-Andre Dufresne, Elastic
AVAILABLE MEDIA
Show details

Sharding: Growing Systems from Node-scale to Planet-scale
Monday, 11:50 am–12:35 pm
Adam Mckaig, Stripe
AVAILABLE MEDIA
Show details

Track 2
Hide details
Grand Ballroom BC
Product Reliability for Google Maps
Monday, 11:00 am–11:45 am
Micah Lerner and Joe Abrams, Google
AVAILABLE MEDIA
Show details

Using Generative AI Patterns for Better Observability
Monday, 11:50 am–12:35 pm
John Feminella, Nuvalence
Show details

1:50 pm–3:25 pm

Track 1
Hide details
Grand Ballroom A
Build vs. Buy in the Midst of Armageddon
Monday, 1:50 pm–2:35 pm
Reggie Davis
AVAILABLE MEDIA
Show details

Compliance & Regulatory Standards Are NOT Incompatible with Modern Development Best Practices
Monday, 2:40 pm–3:25 pm
Charity Majors, honeycomb.io
AVAILABLE MEDIA
Show details

Track 2
Hide details
Grand Ballroom BC
The Ticking Time Bomb of Observability Expectations
Monday, 1:50 pm–2:35 pm
David Caudill, Capital One
AVAILABLE MEDIA
Show details

Synthesizing Sanity with, and in Spite of, Synthetic Monitoring
Monday, 2:40 pm–3:25 pm
Daniel O'Dea, Atlassian
AVAILABLE MEDIA
Show details

3:25 pm–3:55 pm

Break with Refreshments
Pacific Concourse

3:55 pm–5:30 pm

Track 1
Hide details
Grand Ballroom A
Migrating a Large Scale Search Dataset in Production in a Highly Available Manner
Monday, 3:55 pm–4:15 pm
Leila Vaughan, Shopify
AVAILABLE MEDIA
Show details

OIDC and CICD: Why Your CI Pipeline Is Your Greatest Security Threat
Monday, 4:20 pm–4:40 pm
Mark P Hahn, Qualys, and Ted Hahn, TCB Technologies
AVAILABLE MEDIA
Show details

When Your Open Source Turns to the Dark Side
Monday, 4:45 pm–5:30 pm
Dotan Horovits, Logz.io, CNCF Ambassador
AVAILABLE MEDIA
Show details

Track 2
Hide details
Grand Ballroom BC
The Sins of High Cardinality
Monday, 3:55 pm–4:15 pm
Jef Spaleta, Isovalent
AVAILABLE MEDIA
Show details

Optimizing Resilience and Availability by Migrating from JupyterHub to the Kubeflow Notebook Controller
Monday, 4:20 pm–4:40 pm
David Hoover and Alexander Periman, Capital One
AVAILABLE MEDIA
Show details

99.99% of Your Traces Are (Probably) Trash
Monday, 4:45 pm–5:30 pm
Paige Cruz, Chronosphere
AVAILABLE MEDIA
Show details

5:30 pm–6:30 pm

Showcase Happy Hour
Pacific Concourse

Tuesday, March 19

9:00 am–10:30 am

Tuesday Plenary Session
Grand Ballroom

Meeting the Challenge of Burnout
Tuesday, 9:00 am–9:45 am
Christina Maslach, University of California, Berkeley
AVAILABLE MEDIA
Show details

What We Want Is 90% the Same: Using Your Relationship with Security for Fun and Profit
Tuesday, 9:45 am–10:30 am
Lea Kissner, Lacework
AVAILABLE MEDIA
Show details

10:30 am–11:00 am

Break with Refreshments
Pacific Concourse

11:00 am–12:35 pm

Track 1
Hide details
Thawing the Great Code Slush
Tuesday, 11:00 am–11:45 am
Maude Lemaire, Slack
AVAILABLE MEDIA
Show details

Resilience in Action
Tuesday, 11:50 am–12:35 pm
Will Gallego
AVAILABLE MEDIA
Show details

Track 2
Hide details
Bayview Room
"Logs Told Us It Was Kernel – It Wasn't"
Tuesday, 11:00 am–11:45 am
Valery Sigalov, Bloomberg
AVAILABLE MEDIA
Show details

Autopsy of a Cascading Outage from a MySQL Crashing Bug
Tuesday, 11:50 am–12:35 pm
Jean-François Gagné, Aiven, and Swetha Narayanaswamy, HubSpot
AVAILABLE MEDIA
Show details

12:35 pm–1:50 pm

Luncheon
The Atrium
Sponsored by Incident

1:50 pm–3:25 pm

Track 1
Hide details
Bayview Room
Navigating the Kubernetes Odyssey: Lessons from Early Adoption and Sustained Modernization
Tuesday, 1:50 pm–2:35 pm
Raul Benencia, ThousandEyes, part of Cisco
AVAILABLE MEDIA
Show details

Kube, Where's My Metrics? The Challenges of Scaling Multi-Cluster Prometheus
Tuesday, 2:40 pm–3:25 pm
Niko Smeds and Iain Lane, Grafana Labs
AVAILABLE MEDIA
Show details

Track 2
Hide details
Grand Ballroom BC
What Is Incident Severity, but a Lie Agreed Upon?
Tuesday, 1:50 pm–2:35 pm
Em Ruppe, jeli.io/PagerDuty
AVAILABLE MEDIA
Show details

Hard Choices, Tight Timelines: A Closer Look at Skip-level Tradeoff Decisions during Incidents
Tuesday, 2:40 pm–3:25 pm
Dr. Laura Maguire, Trace Cognitive Engineering, and Courtney Nash, The VOID
AVAILABLE MEDIA
Show details

Track 3
Hide details
Seacliff Room
Workshop: Cloud-Native Observability with OpenTelemetry
Tuesday, 1:50 pm–5:30 pm
Liz Fong-Jones, Honeycomb
AVAILABLE MEDIA
Show details

3:25 pm–3:55 pm

Break with Refreshments
Pacific Concourse

3:55 pm–5:30 pm

Track 1
Hide details
Bayview Room
Kubernetes: The Most Graceful Termination?
Tuesday, 3:55 pm–4:40 pm
Harrison Katz, ngrok
AVAILABLE MEDIA
Show details

How We Went from Being Astronauts to Being Mission Control
Tuesday, 4:45 pm–5:30 pm
Laura Nolan, Stanza
AVAILABLE MEDIA
Show details

Track 2
Hide details
Grand Ballroom BC
Triage with Mental Models
Tuesday, 3:55 pm–4:40 pm
Marianne Bellotti, bellotti.tech
AVAILABLE MEDIA
Show details

Defence at the Boundary of Acceptable Performance
Tuesday, 4:45 pm–5:30 pm
Andrew Hatch, LinkedIn
AVAILABLE MEDIA
Show details

Track 3 (continued)
Hide details
Seacliff Room
Workshop: Cloud-Native Observability with OpenTelemetry
Tuesday, 1:50 pm–5:30 pm
Liz Fong-Jones, Honeycomb
AVAILABLE MEDIA
Show details

5:30 pm–7:00 pm

Conference Reception
The Atrium
Sponsored by FireHydrant

Wednesday, March 20

9:00 am–10:35 am

Track 1
Hide details
Grand Ballroom A
System Performance and Queuing Theory - Concepts and Application
Wednesday, 9:00 am–9:45 am
Jeff Poole, Vivint / NRG
AVAILABLE MEDIA
Show details

It Is OK to Be Metastable
Wednesday, 9:50 am–10:35 am
Aleksey Charapko, University of New Hampshire
AVAILABLE MEDIA
Show details

Track 2
Hide details
Grand Ballroom BC
The Art of SRE: Building People Networks to Amplify Impact
Wednesday, 9:00 am–9:45 am
Daria Barteneva, Microsoft
AVAILABLE MEDIA
Show details

Teaching SRE
Wednesday, 9:50 am–10:35 am
Mikey Dickerson, Layer Aleph LLC and Pomona College
AVAILABLE MEDIA
Show details

10:35 am–11:05 am

Break with Refreshments
Pacific Concourse

11:05 am–12:40 pm

Track 1
Hide details
Grand Ballroom A
Cross-System Interaction Failures: Don't Fall through the Cracks
Wednesday, 11:05 am–11:50 am
Tianyin Xu and Xudong Sun, University of Illinois Urbana-Champaign
AVAILABLE MEDIA
Show details

Gray Failure: The Achilles' Heel of Cloud-Scale Systems
Wednesday, 11:55 am–12:40 pm
Ze Li, Microsoft Azure, and Ryan Huang, University of Michigan
AVAILABLE MEDIA
Show details

Track 2
Hide details
The Invisible Door: Reliability Gaps in the Front End
Wednesday, 11:05 am–11:50 am
Isobel Redelmeier, Discord
AVAILABLE MEDIA
Show details

Automating Disaster Recovery: The Ultimate Reliability Challenge
Wednesday, 11:55 am–12:40 pm
Ricard Bejarano, Cisco Systems Inc.
AVAILABLE MEDIA
Show details

1:55 pm–3:30 pm

Track 1
Hide details
Grand Ballroom A
From Chaos to Clarity: Deciphering Cache Inconsistencies in a Distributed Environment
Wednesday, 1:55 pm–2:15 pm
Akashdeep Goel and Prudhviraj Karumanchi, Netflix Inc
AVAILABLE MEDIA
Show details

Patching Your Way to Compliance with a Small Team and a Pile of Technical Debt
Wednesday, 2:20 pm–2:40 pm
Filipe Felisbino, Udemy Inc
AVAILABLE MEDIA
Show details

Strengthening Apache Pinot's Query Processing Engine with Adaptive Server Selection and Runtime Query Killing
Wednesday, 2:45 pm–3:30 pm
Jia Guo, LinkedIn
AVAILABLE MEDIA
Show details

Track 2
Hide details
Grand Ballroom BC
Taming the Linux Distribution Sprawl: A Journey to Standardization and Efficiency
Wednesday, 1:55 pm–2:15 pm
Raj Shekhar and James Fotherby, Quantcast
AVAILABLE MEDIA
Show details

Frontend Design in SRE
Wednesday, 2:20 pm–2:40 pm
Andreas F. Bobak, Google
AVAILABLE MEDIA
Show details

Measuring Reliability Culture to Optimize Tradeoffs: Perspectives from an Anthropologist
Wednesday, 2:45 pm–3:05 pm
Kathryn (Casey) Bouskill, Meta
AVAILABLE MEDIA
Show details

Storytelling as an Incident Management Skill
Wednesday, 3:10 pm–3:30 pm
Laura de Vesine, Datadog, Inc
AVAILABLE MEDIA
Show details

3:30 pm–4:00 pm

Break with Refreshments
Grand Foyer

4:00 pm–5:30 pm

Hide details
Closing Plenary Session
Grand Ballroom

Real Talk: What We Think We Know — That Just Ain't So
Wednesday, 4:00 pm–4:45 pm
John Allspaw, Adaptive Capacity Labs
AVAILABLE MEDIA
Show details

What Can You See from Here?
Wednesday, 4:45 pm–5:30 pm
Tanya Reilly, Squarespace
AVAILABLE MEDIA
Show details

5:30 pm–5:35 pm

Closing Remarks

20 Years of SRE: Highs and Lows

Niall Murphy, Stanza Systems

SRE Thinking – people thinking of production; engineering problems away

Put a slightly less terrible thing in place today, before the next terrible thing comes up. This buys us time.

Highlights

- SRE arrived across many industry, and in multi-national companies. This is a good sign for a profession.
- Knowledge is permutated: Taxonomy & Terms.
- DevOps, ITIL, Platform Engineering – SRE model does has a say how systems be managed, how organizations should be structured.
- People continue want to learn about it.
- Forrester: 46% of Google SRE principles apply directly to your enterprises – what about the rest ?
- SRE ideas have penetrated general business conscientiousness
- Gartner: by 2027, 75% of enterprises will use SRE practices across their organizations

Lowlights

- Career Pipeline needs attention.
- A number of failed SRE implementations in companies whose names you've heard of – *Doing SRE is hard*
- Widespread mishandling, dilution, problem with what the term & role means
- Neither generating nor applying (mathematical) models often and widely enough
- Zero interest rate no more – reliably no longer matters as growth is not available
- Doctrine overtaking reality-oriented pragmatism
- "Operations is low status" battle is no yet won

Using Generative AI Patterns for Better Observability

John Feminella, Nuvalence

LLM Patterns

- **Helpful assistant pattern**: Turn natural language questions into discovery queries
- **Artifact inspector**: Answer questions about a large set of documents you haven't read
- Application on LLM on metric data? Don't reach for models here first (too complex & expensive) - there are other things that fit better)

Use cases:

- Give suggestions for what to try next when I am stuck (i.e. identify potential solutions for a queue backup based on the provided logs)
- Summarize something complicated (i.e. what are the 5 most important points in this incident report)
- Quickly gain footing in unfamiliar territory (i.e. list common reasons for memory leak in a python application and suggest remedies)

Skepticism

- GenAI returns non-deterministic responses
- Answers are *plausible*, but not necessarily *correct* (sometimes this is correlated, but many times its is not)
- Cheating vs. AIOps vs. LLM

LLM are real - LLM are commercially available for small amount of money - You should use LLM (try them out and see if patterns provide help)

The Ticking Time Bomb of Observability Expectations

David Caudill, Capital One

Cognitive load: “how much stuff are you holding in your head at once”

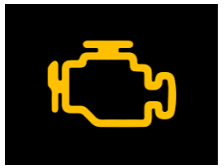
Do:

- Design dashboard for a new developer *with migraine*
- Take advantage of *note widget* in dashboard (what does this metric mean)
- Group logically similar items together, use *visual container*
- Decide on a *few very* important dashboard you will share
- Put someone in charge of *making the data meaningful*

Don’t

- Don’t pack status dashboards with more info than you need
- Don’t build a new dashboard every time you want to know something – aka. dashboard fatigue
- Don’t build dashboards in response to every single incident
- Don’t break Conveys Law with your org structure (responsible for only half of the service)

Start with what’s important, not what’s easy



Status Information (Check Engine)

←

Diagnostic Information (OBD2 code)
(On-board diagnostics - II)

→

When Your Open Source Turns to the Dark Side

Dotan Horovits, Logz.io

Recent spike of license changes -> OpenSource monetarization is problematic

- Elasticsearch & Kibana: moving away from Apache 2 to dual-licensed under SSPL -> OpenSearch
- HashiCorp: moving to BSL -> OpenTufu, OpenBao
- Grafana, Loki, Tempo relicensed to AGPLv3 (OSI compliance license)
- Going rogue: colors.js and faker.js; Offending commit
- …

Open Source is more than a license

What prevents the license to change ?

Who can make the change ?

Who governs the OSS: Individuals, Vendors, Foundations

Who is the “our” in “open source”?

Building OSS widely

- OSS is not a business model
- Build a sustainable business model, create differentiation on your cloud service
- Don’t expect material compensation

Using OSS wisely

- Manage your third licensing exposure (just like a security exposure)
- Take care with automation (automatic updates during CI/CD need to check license changes)
- Code smells can signal upcoming relicensing
- Extending OSS functionality: prefer plugins over downstream modifications

Selecting OSS wisely

- Which OSS license ?
- Who is behind the OSS ?
- What is the governance policy ?
- Vendor distros (packaging of the upstream project) of OSS can shield.

Meeting the Challenge of Burnout

Christina Maslach University of California, Berkeley

Maslach Burnout Inventory https://en.wikipedia.org/wiki/Maslach_Burnout_Inventory

Who is burning out - Not just a person-only focus -> Need to better ask: *Why* are people burning out

Typically, chronic workplace stressors that have not been successfully managed

- Feeling of energy depletion or exhaustion
- Increased mental distance from one's job, or feeling of negativism or cynicism related to one's job
- Reduced professional efficacy

Burnout is an *occupational* phenomenon. It is *not* classified as a medical condition.

Six Paths to a healthier workplace

- Demand overload – Sustainable workload
- Lack of control - Choice and control
- Insufficient Reward – Recognition and Reward
- Breakdown of Community – Supportive work community
- Absence of fairness – Fairness, respect, and social justice
- Value Conflicts – Clear values and meaningful work

Matching the job to people

- Modify the work conditions
- Environmental Psychology and the model of Ergonomics
- Apply design model to the social and psychological environment: Autonomy, Belongingness, Competence, Psychological safety, Fairness, Meaning, Positive emotions

Thawing the Great Code Slush

Maude Lemaire, Slack

Refactoring Chef Repo (terraform, chef scripts, etc)

- 335 distinct contributors
- 2473 Total PRs

3 practices established:

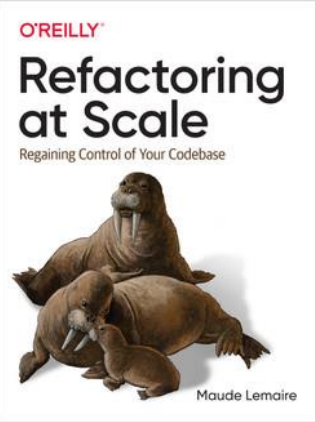
Reduce Risk associated with component: observability, resiliency, readiness, ability to rescue, Blast Radius of deploy
Calculate the aggregated score across these topics

Automated testing

Gradual rollout for a wider breadth of changes

Isn’t CAB just a good code review ?

Beginners Mind: refers to having an attitude of openness, eagerness, and lack of preconceptions when studying.



What makes a good code review (**Conference Slack**)

- that the code is actually read and understood vs just click approved
- sharing expertise about potentially risky choices being made
- Alignment, Empathy, and Trust
- ensuring that it's clear to everyone what the intent of the code change is
- Asking questions if something isn’t clear or impact isn’t specified.
- asking "what is the risk of this and have we mitigated it"
- the review makes you completely rethink your implementation
- spreading knowledge so that the reviewer also is equally aware of the workings of what was added
- documenting not just what is changing, but why
- Being empathetic/kind but asking thoughtful/deep/tough questions if needed about potential issues and risks. Keeping the review focused on the code itself, not the person who wrote the code
- tests are confirmed to test behavior instead of implementation
- Explaining what, why a change is required for. How does it impact customers? Put evidences of test (specially integration test reports).
- Also what is the rollback strategy if sth goes wrong

What is Incident Severity, but a Lie Agreed Upon?

Em Ruppe, Jeli.io

There is no correct way to do incident severity

Incident severity is a tapestry of trade-offs

What is severity measuring?

- Impact (can you measure it ?)
- Priority (can you measure it ?)

What is severity a mechanism for

- Communication
- After response (PIR, SLA to provide an RCA, Action items for RCA)

What organizational problem are you blaming the organization for ?

- Organizational / process / cultural issues *masked* through severity

If your are questioning whether its an incident, declare an incident (Fischer's Rule).

Addendum to Fischer's Rule: if multiple parties are debating the severity level of an incident during an incident, then it defaults to the higher level.

Severity …

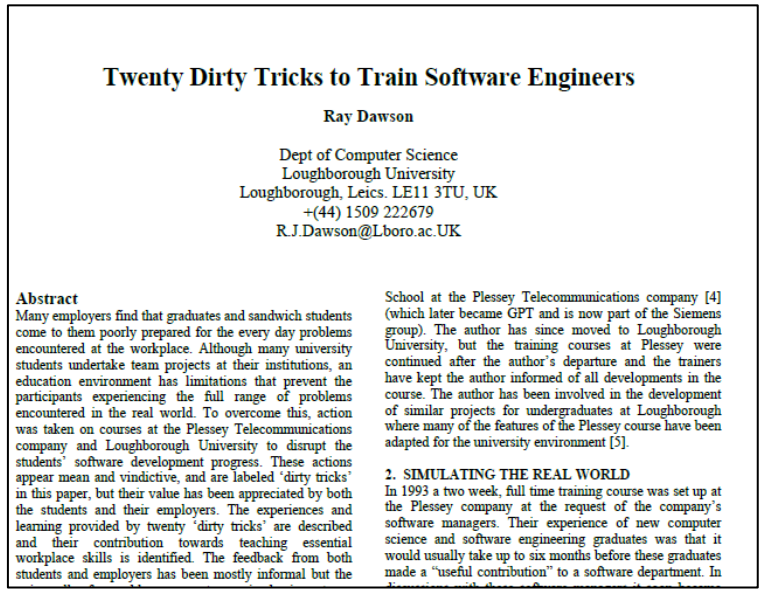
- Is measurable
- Is actionable
- Sets expectations
- Is not one size fits all
- Is a canary
- Is not forever (it *will* get stale)

Talk to the people at the sharp end.

Teaching SRE

Mikey Dickerson, Layer Aleph LLC and Pomona College

https://dl.acm.org/doi/abs/10.1145/337180.337204



Twenty Dirty Tricks to Train Software Engineers

Ray Dawson

Dept of Computer Science
Loughborough University
Loughborough, Leics. LE11 3TU, UK
+(44) 1509 222679
R.J.Dawson@Lboro.ac.UK

Abstract

Many employers find that graduates and sandwich students come to them poorly prepared for the every day problems encountered at the workplace. Although many university students undertake team projects at their institutions, an education environment has limitations that prevent the participants experiencing the full range of problems encountered in the real world. To overcome this, action was taken on courses at the Plessey Telecommunications company and Loughborough University to disrupt the students' software development progress. These actions appear mean and vindictive, and are labeled 'dirty tricks' in this paper, but their value has been appreciated by both the students and their employers. The experiences and learning provided by twenty 'dirty tricks' are described and their contribution towards teaching essential workplace skills is identified. The feedback from both students and employers has been mostly informal but the

School at the Plessey Telecommunications company [4] (which later became GPT and is now part of the Siemens group). The author has since moved to Loughborough University, but the training courses at Plessey were continued after the author's departure and the trainers have kept the author informed of all developments in the course. The author has been involved in the development of similar projects for undergraduates at Loughborough where many of the features of the Plessey course have been adapted for the university environment [5].

2. SIMULATING THE REAL WORLD

In 1993 a two week, full time training course was set up at the Plessey company at the request of the company's software managers. Their experience of new computer science and software engineering graduates was that it would usually take up to six months before these graduates made a "useful contribution" to a software department. In

Teaching SRE

- **Intention is to address the mindset**
- **Not step-by-step, more outcome-based directions**

Reading

- Well studied system disasters: Challenger, Three Mile Island, Air France 447
- Sensemaking (Karl Weick); Systems safety (Nancy Leveson)
- Disasters of a complex-systems nature (Mann Gulch wildfire, Chernobyl, Therac-25)

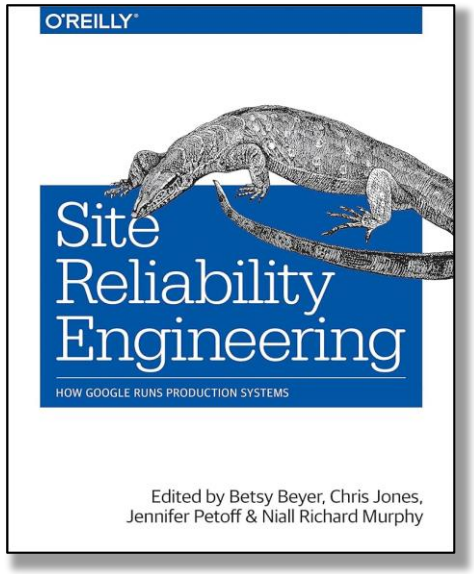
Practicum sequence (team of 4 to 5 ; weekly assignments, build on one the next)

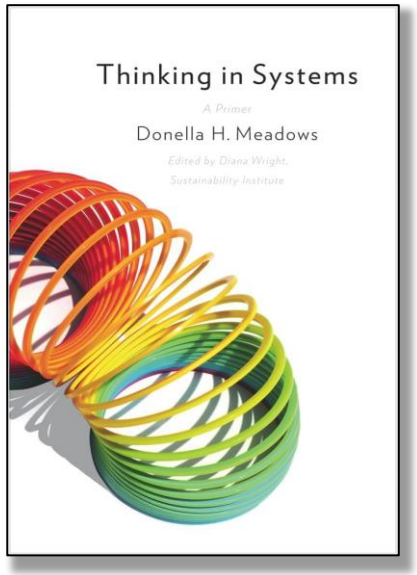
- Get AWS account, setup VM, ssh
- Install postgres, import data, write SQL
- Install tomcat, python, webapp
- Make the above actually work together *(difficulty spike)*
- Set APM-style instrumentation (investigate 4-5 options)
- Set up an oncall rotation with pagerduty
- Show improvement over week 4 availability under synthetic load of about 100 rpm *(confidence improves)*
- Show 90% availability (+ side activity) – disk fills up, memory leak
- Solve a synthetic outage and write a post mortem
- Rearchitect and scale up to withstand a 15 minute load test of 10,000 rpm – RDS, duplicate instance
- Deploy changes with CI/CD, sometimes containers

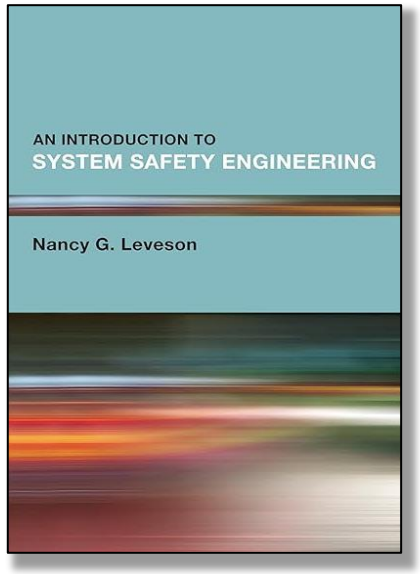
Planned surprises
→ intention is to build resilience and adaptability

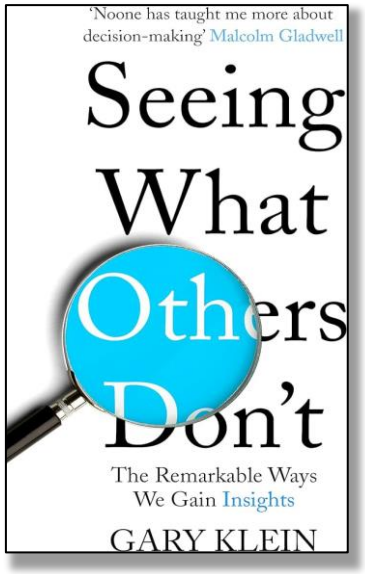
Unplanned surprises (Student does a rm -rf /)
→ trigger backup and version control

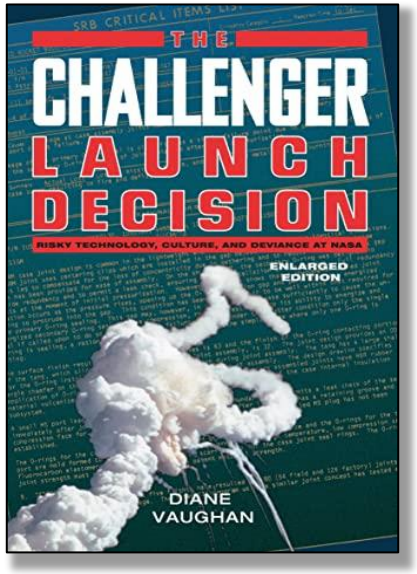
Books mentioned

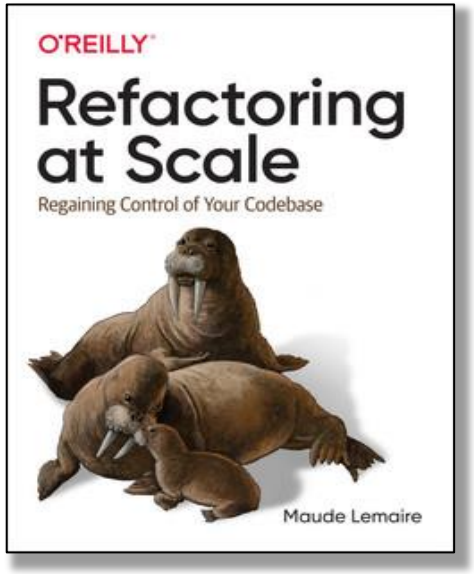


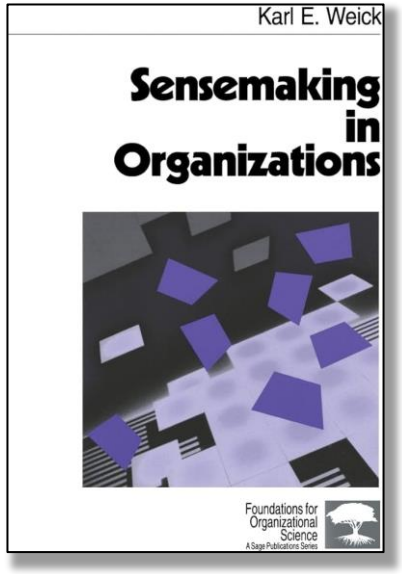












… any many more

Closing Thoughts

Many sessions were not at the beginner level, despite the fact that a larger portion of the attendees were attending the conference for the first time.

While the topics presented were all to the point, I had conversations during lunch and dinner with people that were confused:

- severities are bad
- the incident response is not a proper sequence
- observability (just starting to understand the difference to traditional monitoring and logging) needs to evolve once more
- even Google had significant outages that were not detected via automated alerts
- OpenSource is risky, even if it is as mature as Terraform
- Etc.

I am not saying that these topics were wrong (in fact, they match what I am seeing), but without the context and the proper maturity it can leave people baffled.

References and Links

Presentations/video/voice available at
https://www.usenix.org/conference/srecon24americas/program

Conference summary report:

- https://etherpad.wikimedia.org/p/Amer24-conference-report

Blog-Posts (English)

- https://willgallego.com/2024/03/24/srecon24-americas-recap/
- https://www.linkedin.com/posts/vshynkar_srecon-incedentmanagemet-productivityalchemy-activity-7178042973903806465-juTH/
- https://www.linkedin.com/feed/update/urn:li:activity:7175725519697567744
- https://www.linkedin.com/feed/update/urn:li:activity:7176463047098540032

Blog-Posts (Portuguese)

- https://www.linkedin.com/posts/luizcgs13_aqui-esta-o-resumo-do-primeiro-dia-de-srecon-activity-7175896157506707456-yt9T

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