# A Post Incident Review²

Tom Partington - ANZx

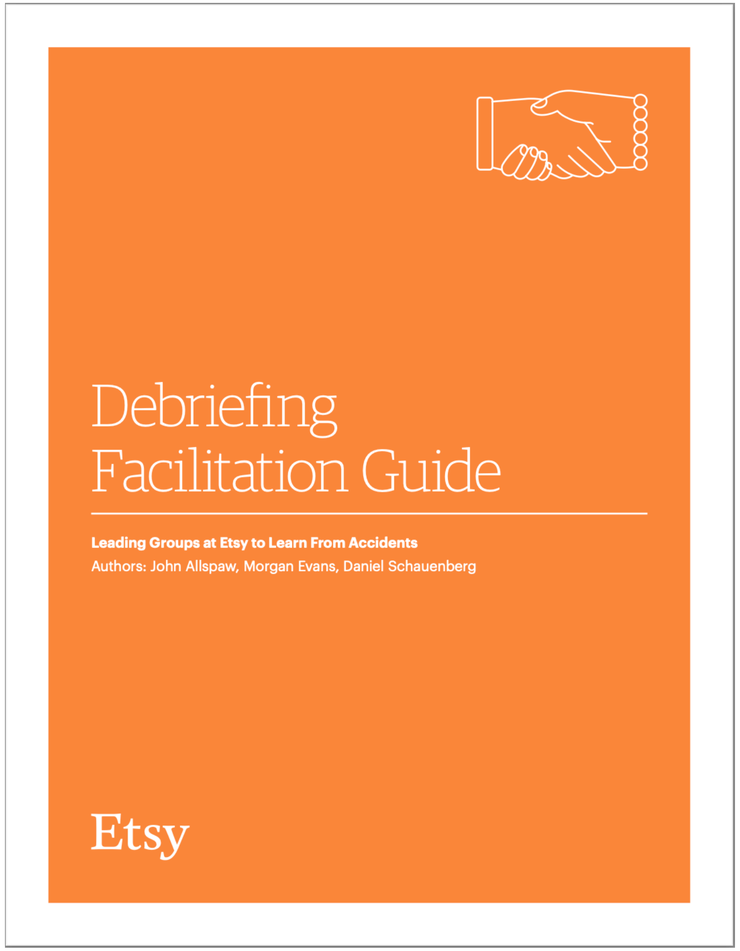

@parmigiana

Part 1: Common PIR Styles

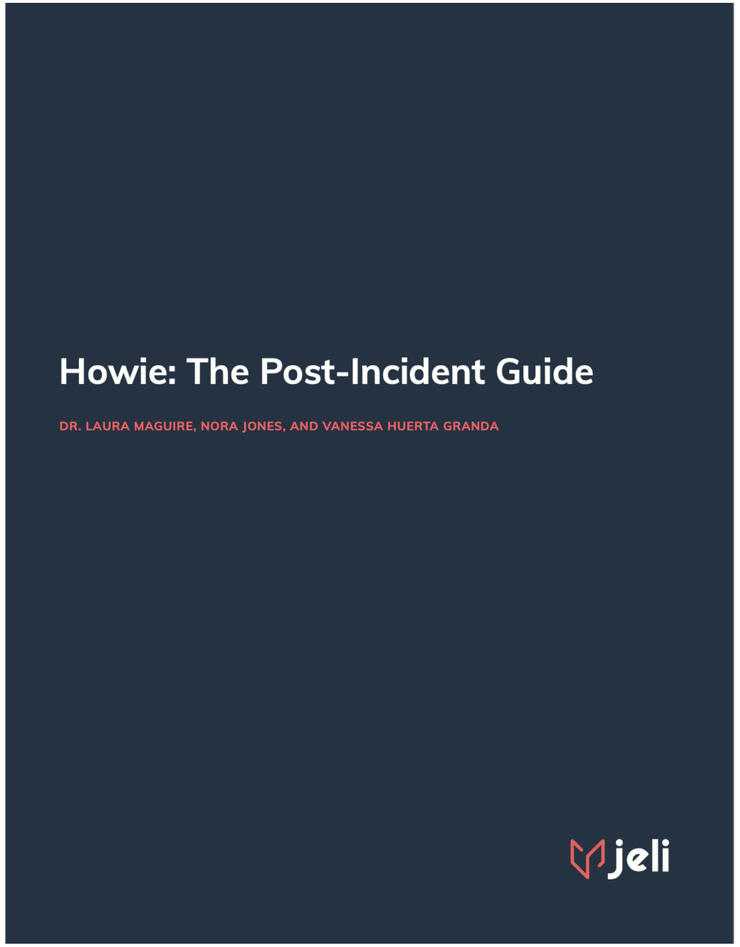
Part 2: Our Process

Part 3: What’s missing and improvements

# Hands on Guides




**Etsy**
**Debrief Facilitation Guide**
**2016**




**Jeli**
**Howie: The Post-Incident Guide**
**2021**

# Maps, Context, and Tribal Knowledge:

## On the Structure and Use of Post-Incident Analysis Artifacts in Software Development and Operations

J. Paul Reed | LUND UNIVERSITY

**As an industry, we are not getting better at this**; that is, we do not possess some inherent quality or skill that makes us 'auto*magically*' improve as we experience our own organizational incidents; and there is no evidence to suggest that we pay any attention to other software organization's incidents and outages in a complete enough way so as to be of use in reducing or eliminating our own organizational incidents and accidents, as we observe in, say, aviation accidents.

- J Paul Reed
Maps Context and
Tribal Knowledge

SRE Work is Cognitive Work

# Part 1: Common PIR Styles

# PIR Styles

1.

**Mechanistic Reasoning:**
The belief that our systems are like complicated machines, made up of components with no intrinsic relationships between them.

# How Complex Systems Fail

*(Being a Short Treatise on the Nature of Failure; How Failure is Evaluated; How Failure is Attributed to Proximate Cause; and the Resulting New Understanding of Patient Safety)*

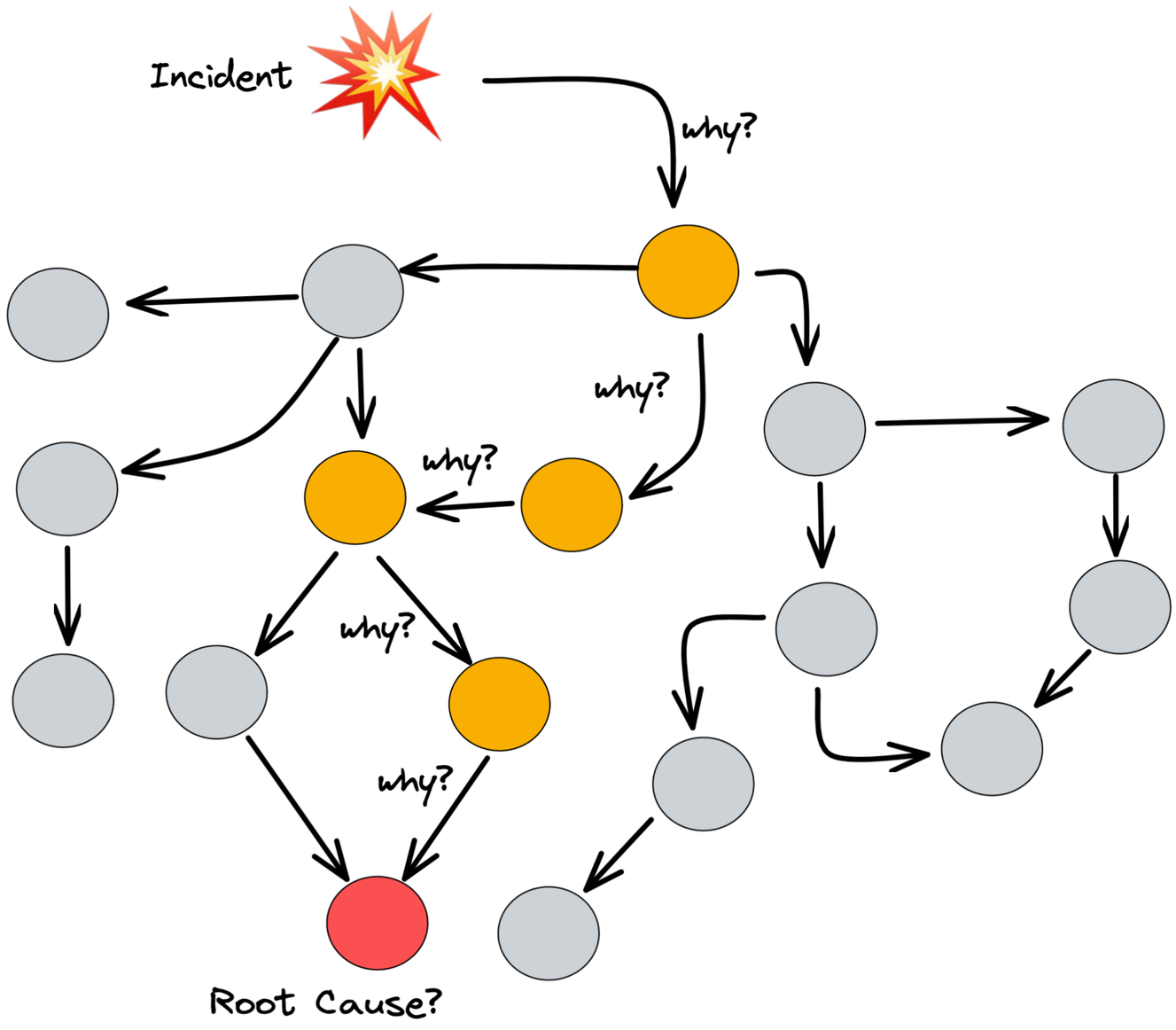
Richard I. Cook, MD
Cognitive technologies Laboratory
University of Chicago

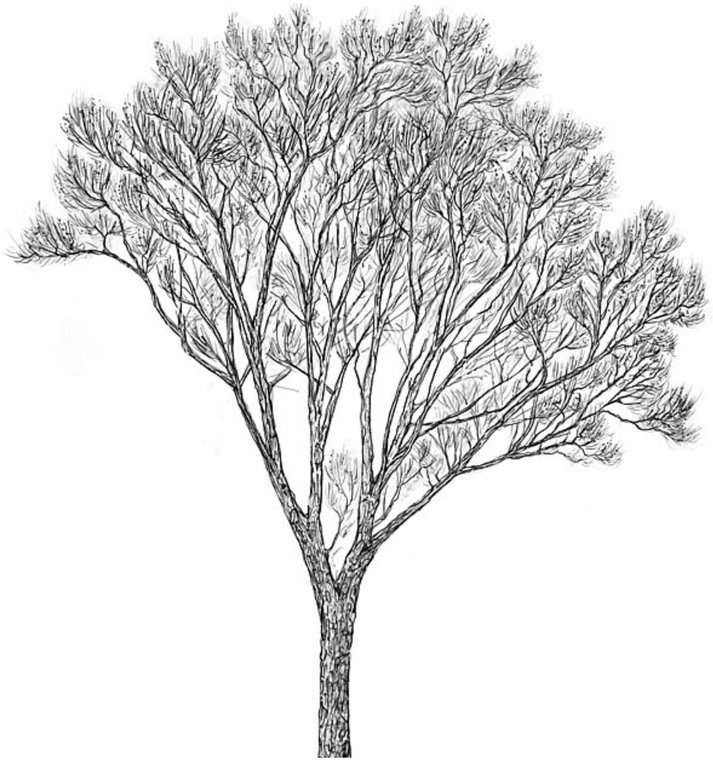
https://how.complexsystems.fail/

# PIR Styles

1.

2. Why? Why? Why? Why? Why?

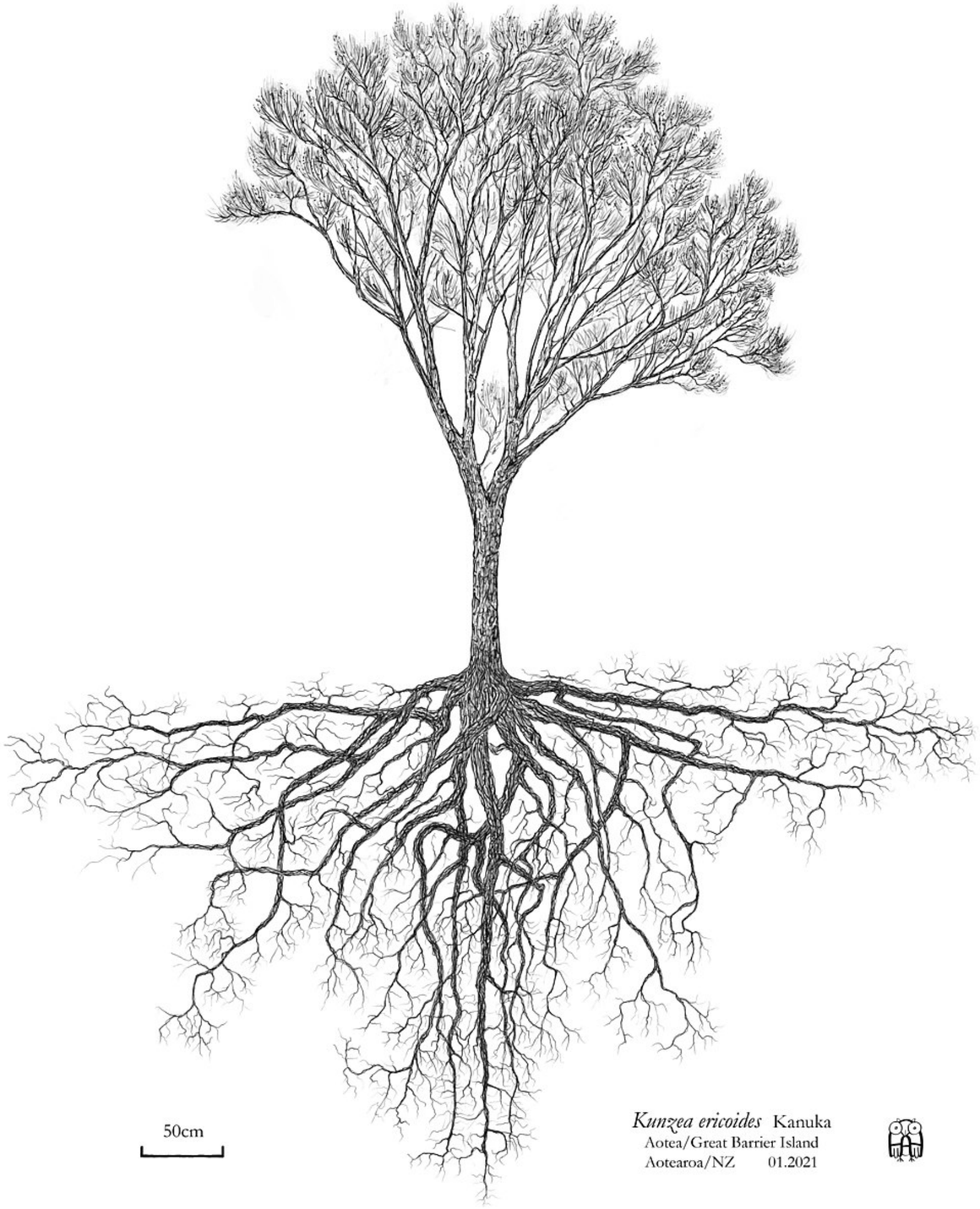

Incident
why?
why?
why?
why?
why?
Root Cause?



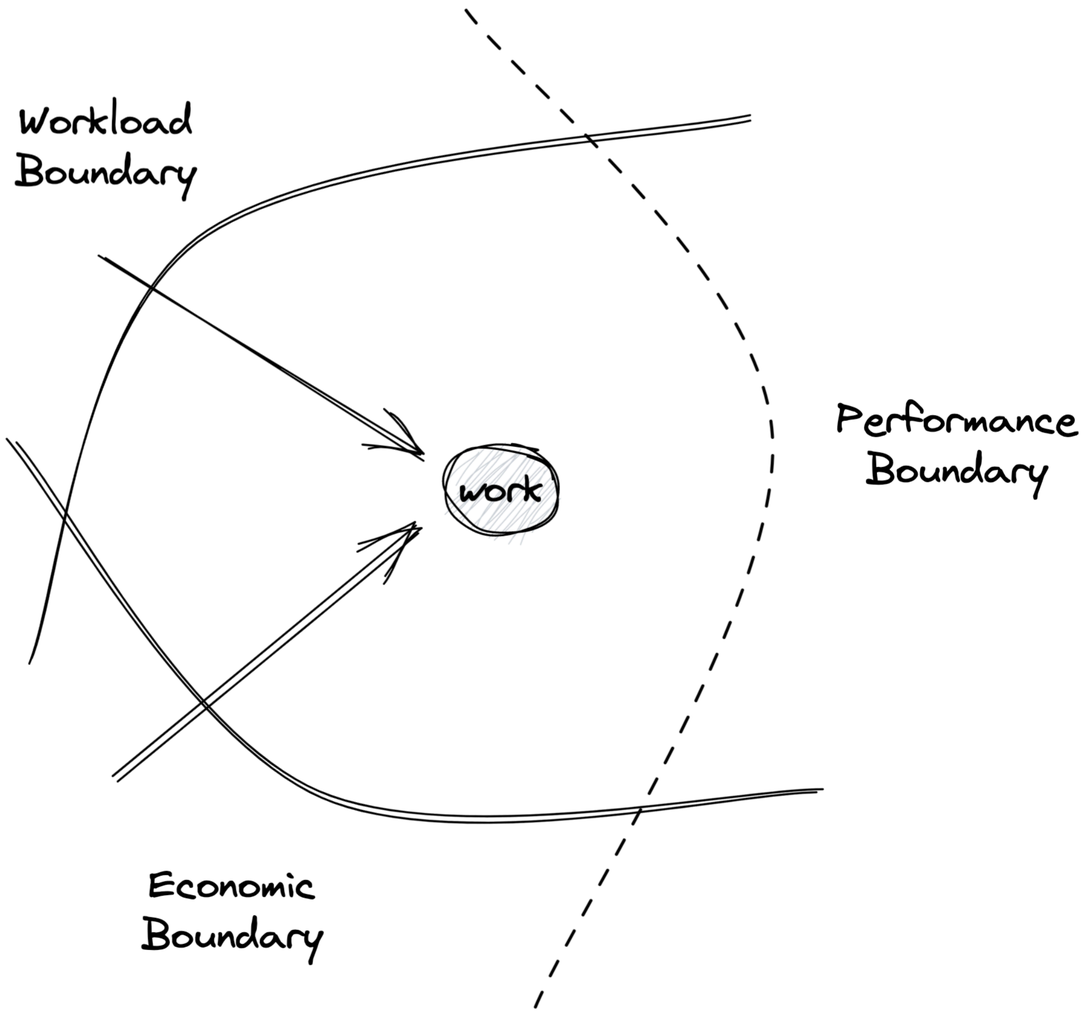
# Root Cause(s)?

50cm

*Kunzea ericoides* Kanuka
Aotea/Great Barrier Island
Aotearoa/NZ 01.2021



# Rasmussen's Safety Model

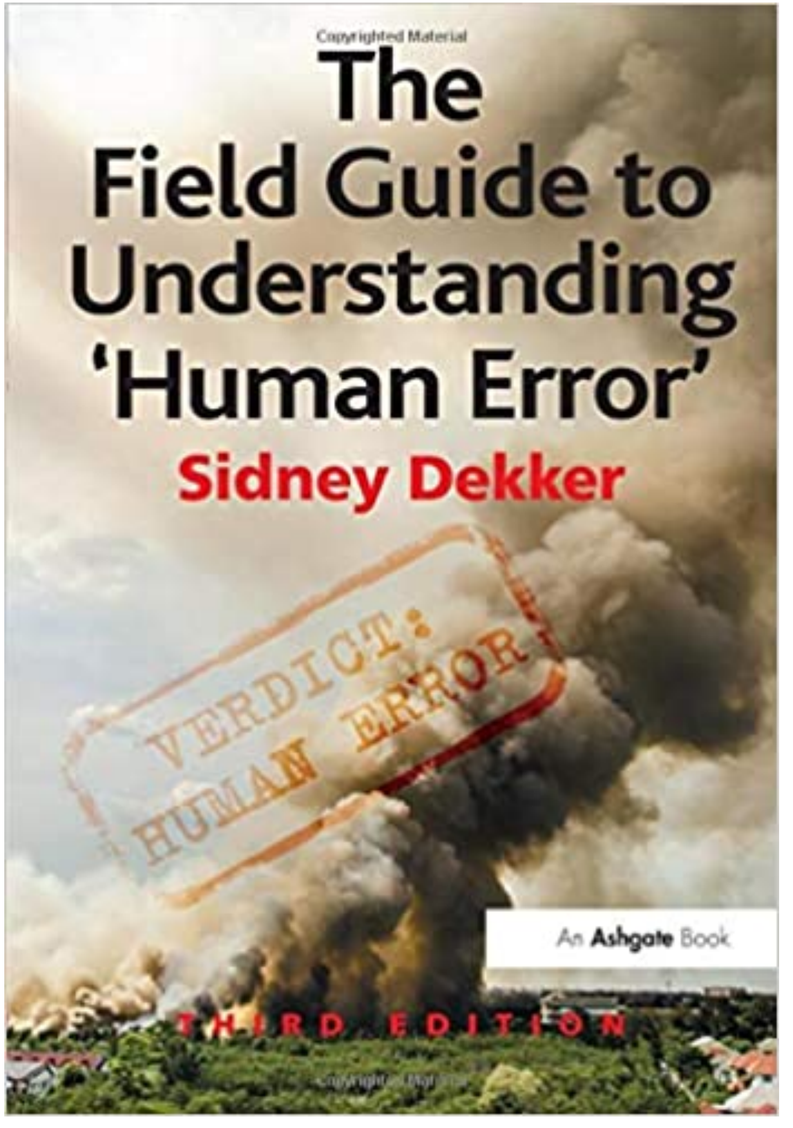


# PIR Styles

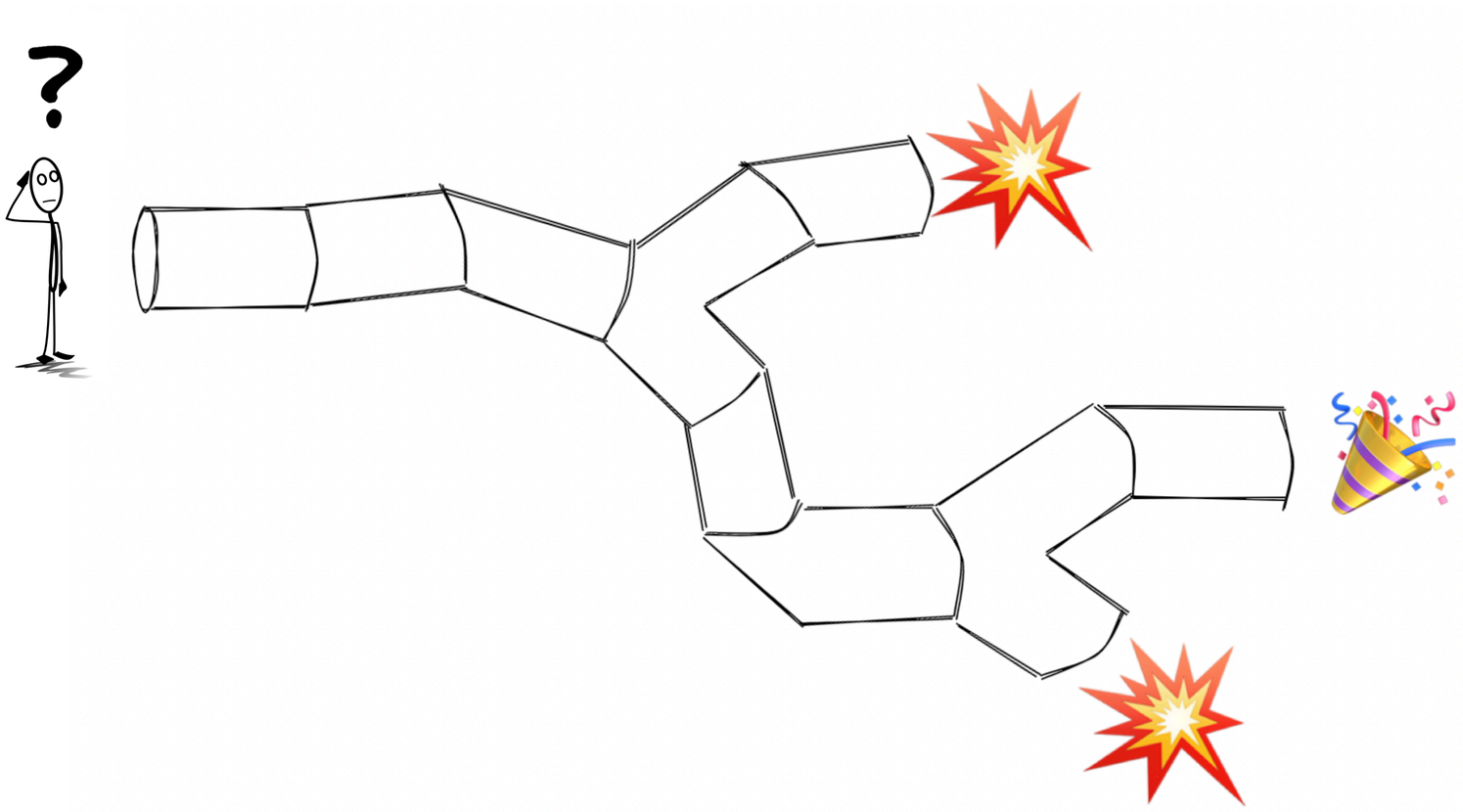
1.

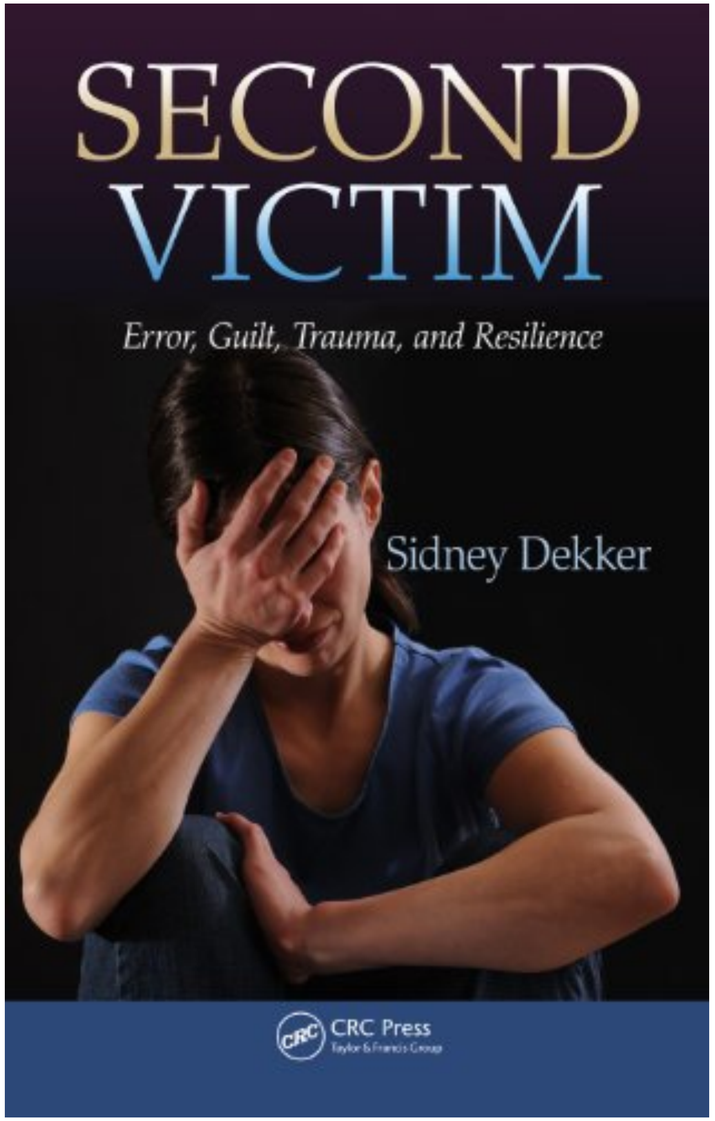
2. Why? Why? Why? Why? Why?

3. On Friday [redacted] forgot to disable the auto-scaling, so everything scaled down to 0 resulting in an outage for the whole weekend


Copyrighted Material
The Field Guide to Understanding 'Human Error'
Sidney Dekker
VERDICT: HUMAN ERROR
An Ashgate Book
THIRD EDITION

# Dekker’s Tunnel




SECOND
VICTIM
Error, Guilt, Trauma, and Resilience
Sidney Dekker
CRC Press

# PIR Styles

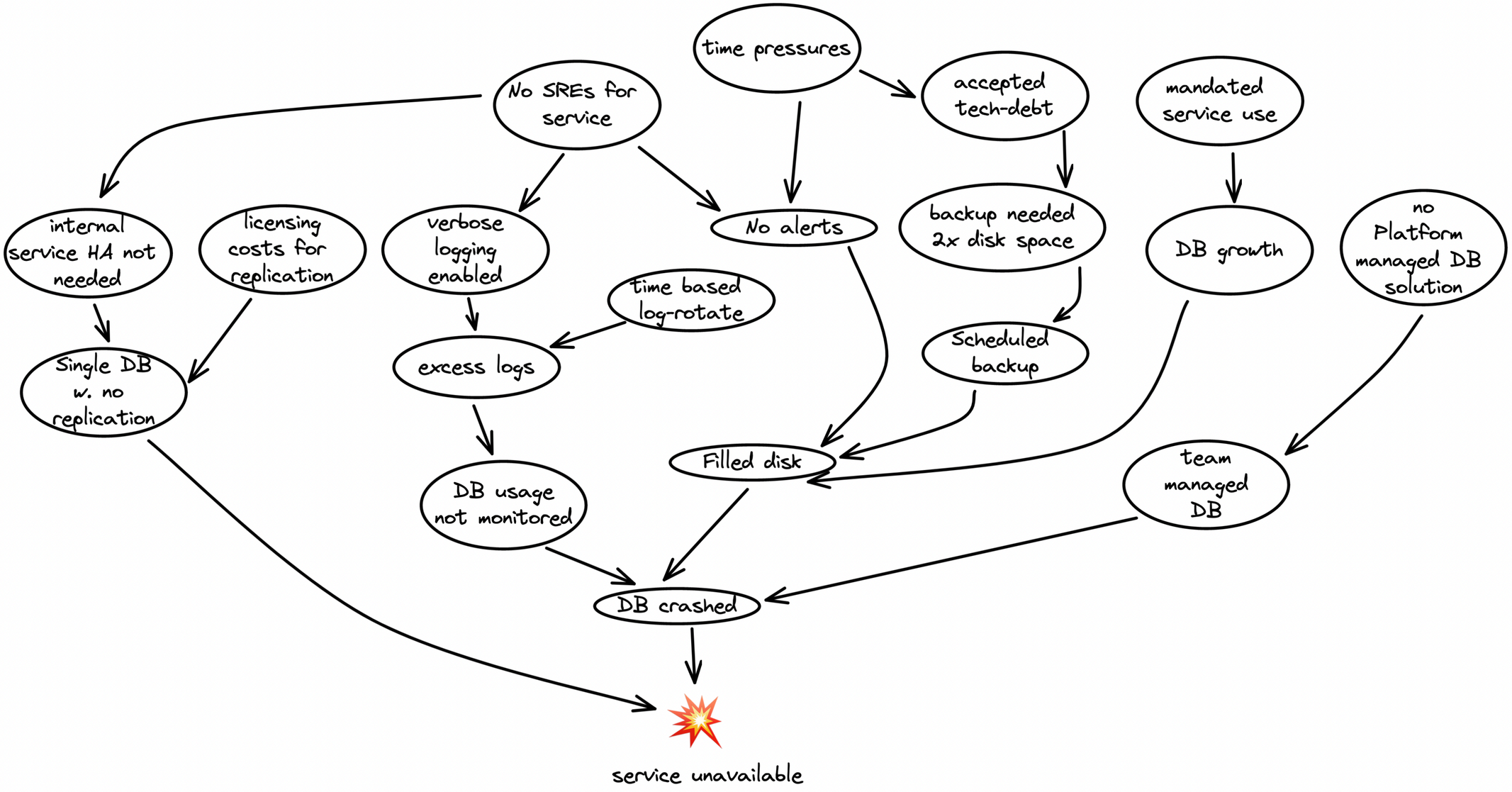
1.

2. Why? Why? Why? Why? Why?

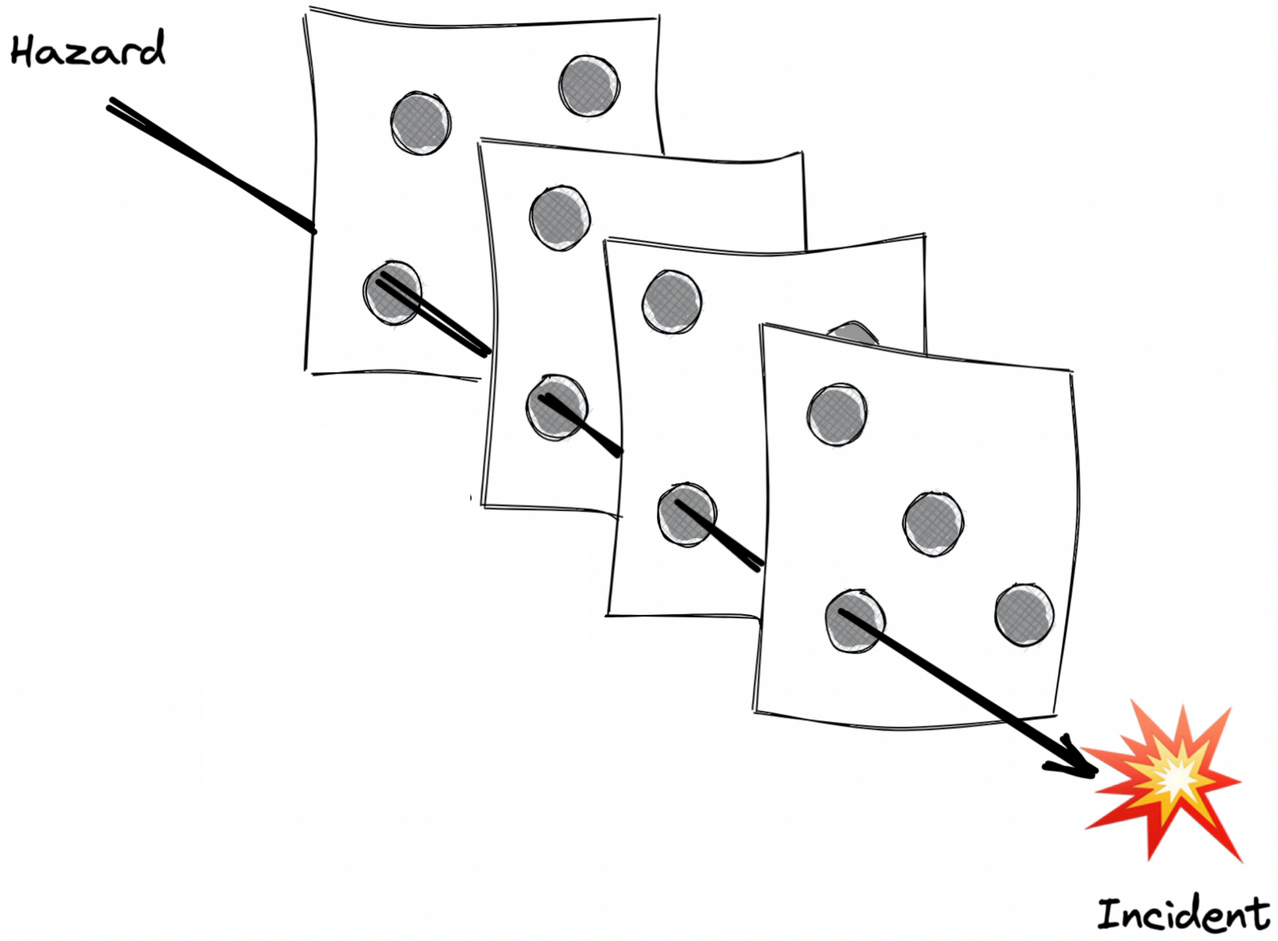
3. On Friday [redacted] forgot to disable the auto-scaling, so everything scaled down to 0 resulting in an outage for the whole weekend

4. Causal Map

# Causal Map


time pressures
No SREs for service
accepted tech-debt
mandated service use
internal service HA not needed
licensing costs for replication
verbose logging enabled
No alerts
backup needed 2x disk space
DB growth
no Platform managed DB solution
time based log-rotate
Single DB w. no replication
excess logs
Scheduled backup
Filled disk
team managed DB
DB usage not monitored
DB crashed
service unavailable

# James Reason's 'Swiss Cheese Model' of Accident Causation



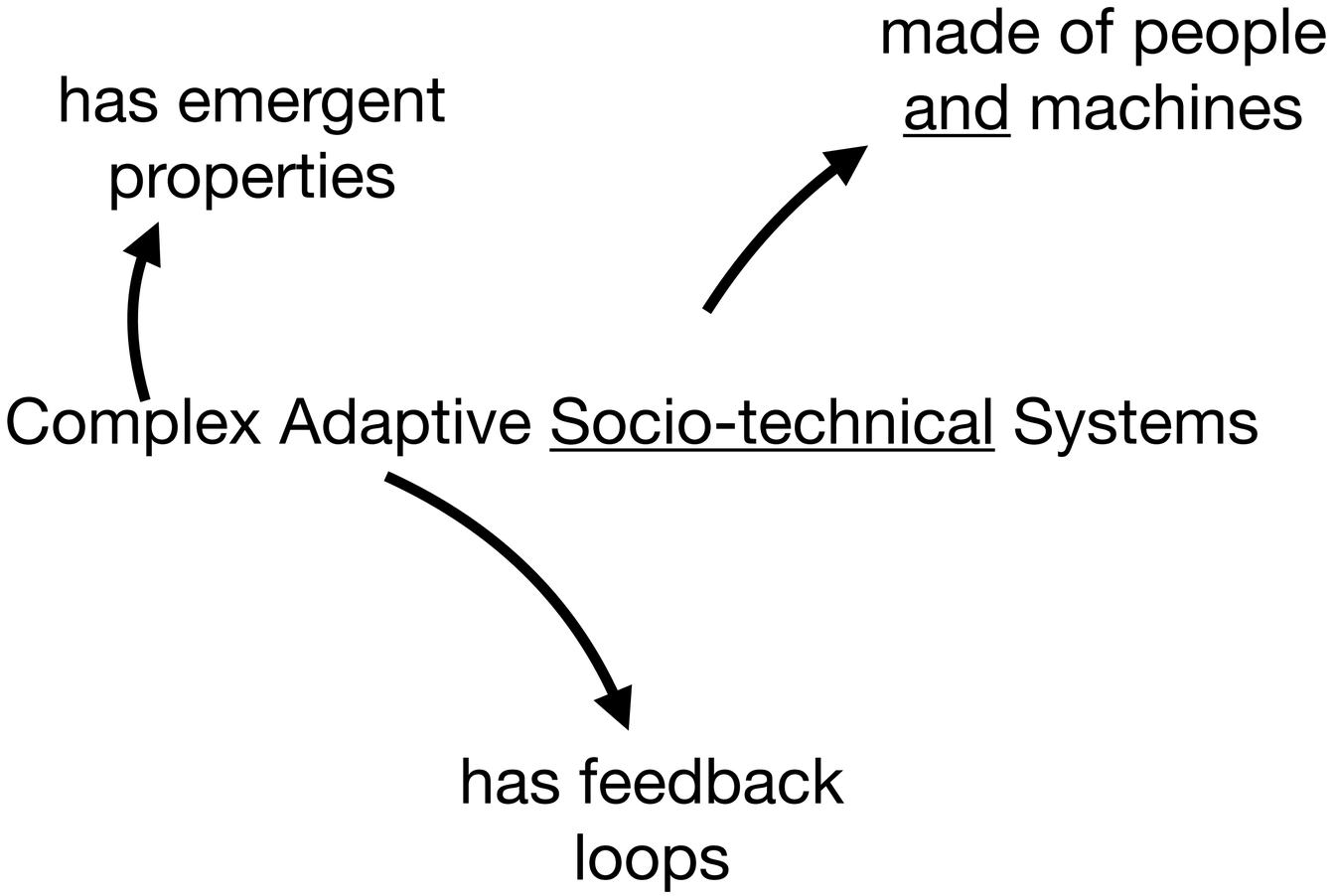
has emergent properties

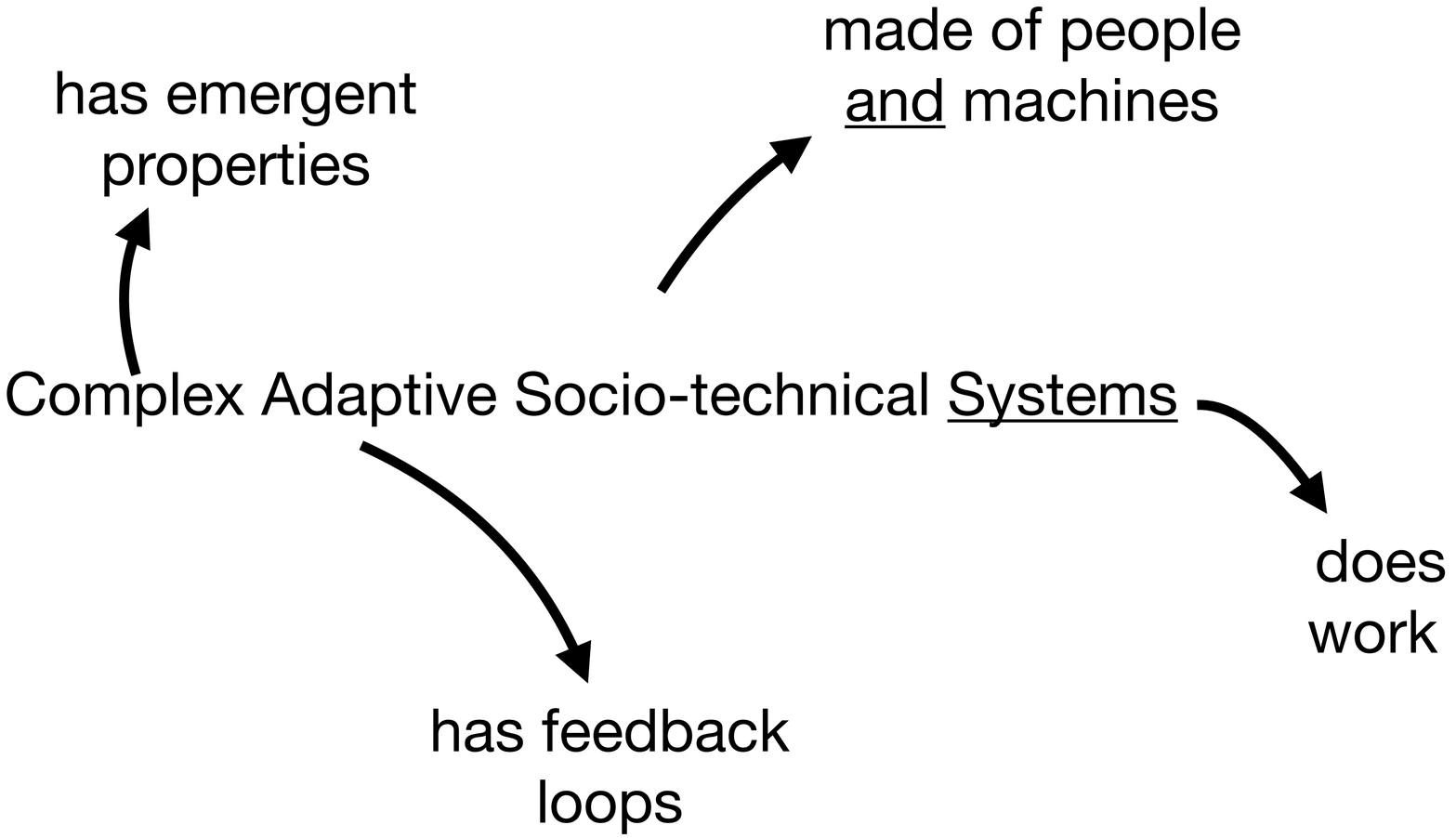
<u>Complex</u> Adaptive Socio-technical Systems

has emergent properties

Complex <u>Adaptive</u> Socio-technical Systems

has feedback loops


made of people
and machines
has emergent
properties
Complex Adaptive Socio-technical Systems
has feedback
loops


made of people
and machines
has emergent
properties
Complex Adaptive Socio-technical Systems
does
work
has feedback
loops

# STELLA

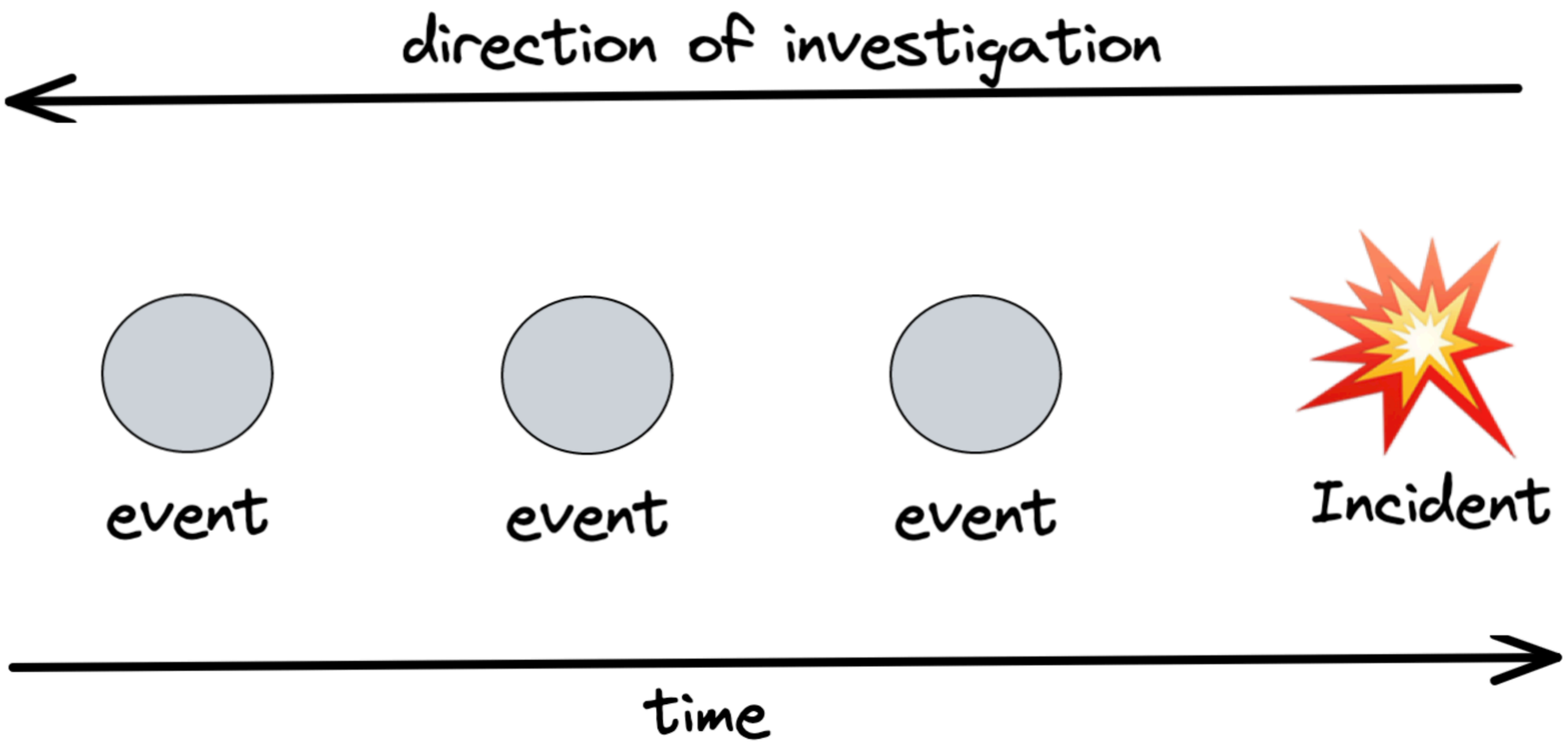
## Report from the SNAFUcatchers Workshop on Coping With Complexity

*Brooklyn NY, March 14-16, 2017*

**Winter storm STELLA**

Woods' Theorem: *As the complexity of a system increases, the accuracy of any single agent's own model of that system decreases rapidly.*

https://snafucatchers.github.io/


direction of investigation
event
event
event
Incident
time

# PIR Styles

1.

2. Why? Why? Why? Why? Why?

3. On Friday [redacted] forgot to disable the auto-scaling, so everything scaled down to 0 resulting in an outage for the whole weekend

4. Causal Map

# PIR Styles

1.

2. Why? Why? Why? Why? Why?

3. On Friday [redacted] forgot to disable the auto-scaling, so everything scaled down to 0 resulting in an outage for the whole weekend

4. Causal Map

5. Blame Aware After Action Review

# Part 2: Our Process

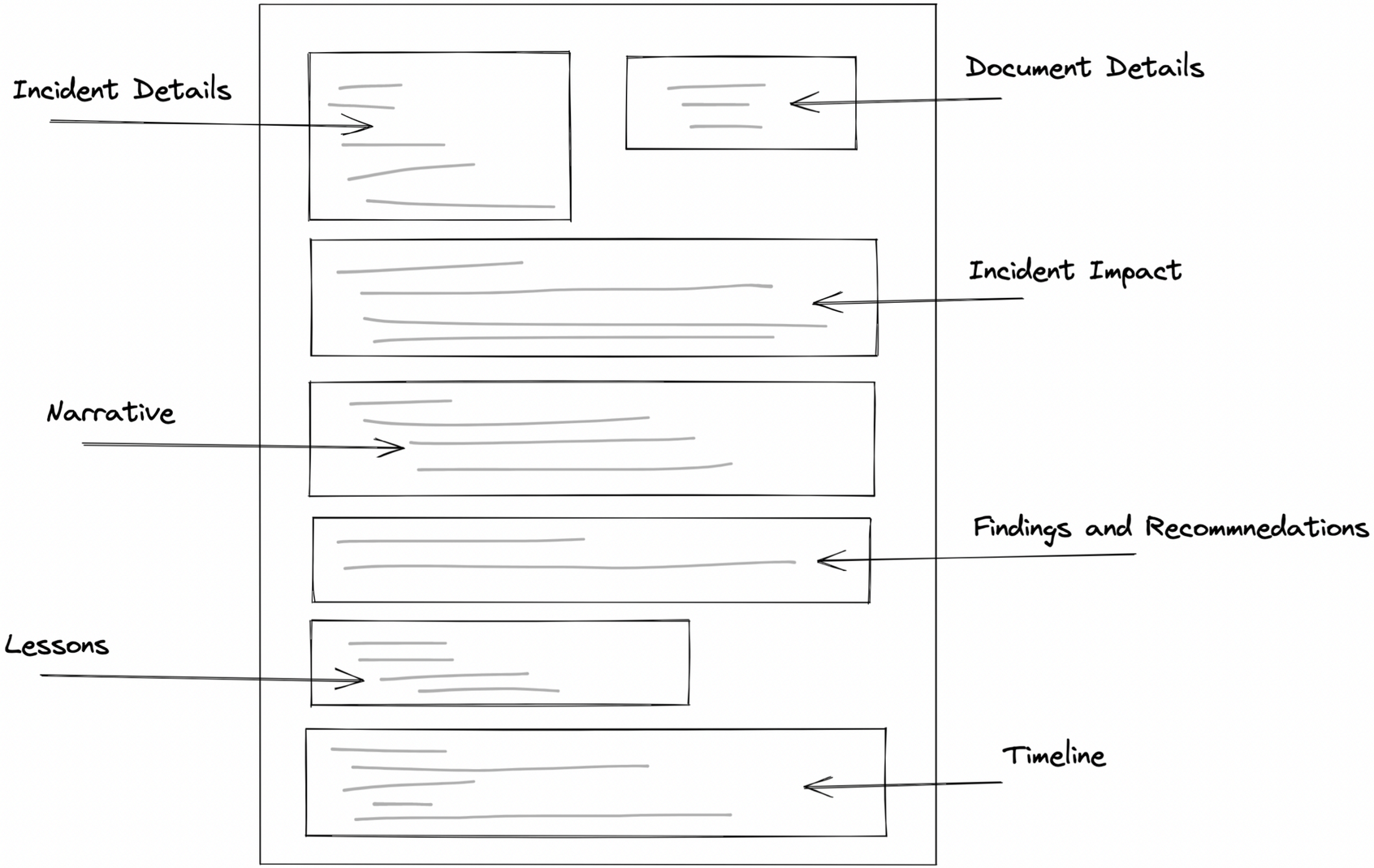
Acknowledge the incident

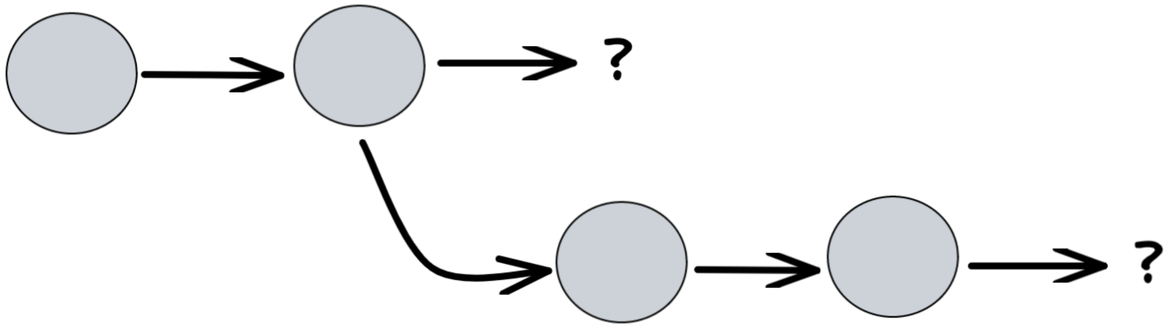
hold a ‘what we know about x incident’ meeting

# Record vs Report

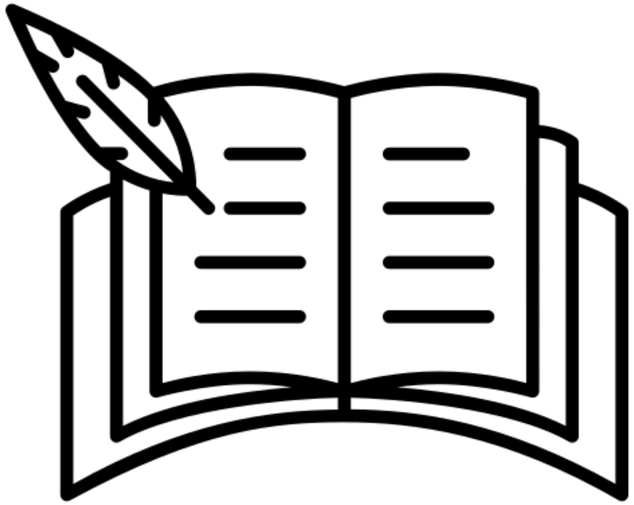
| | Incident Record | Incident Report |
|---|---|---|
| Incident Details | ✓ | ✓ |
| Incident Impact | ✓ | ✓ |
| Incident Narrative | ✓ | ✓ |
| Timeline | ? | ✓ |
| Incident Debrief | | ✓ |
| Lessons Learned | | ✓ |
| Possible Remediation Ideas | | ✓ |
| Recommendations | | ✓ |

# Incident Template



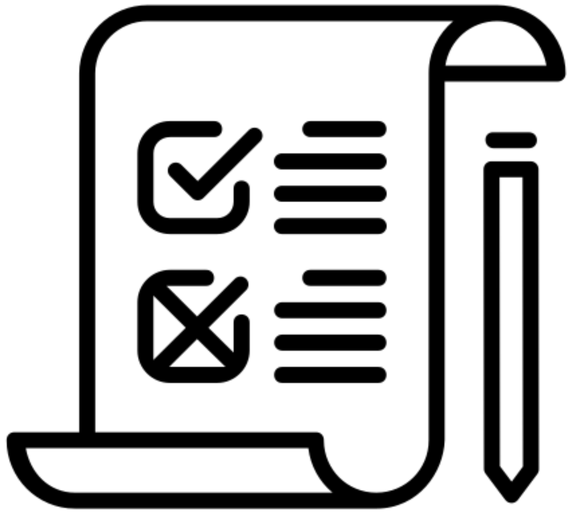


# Narrative



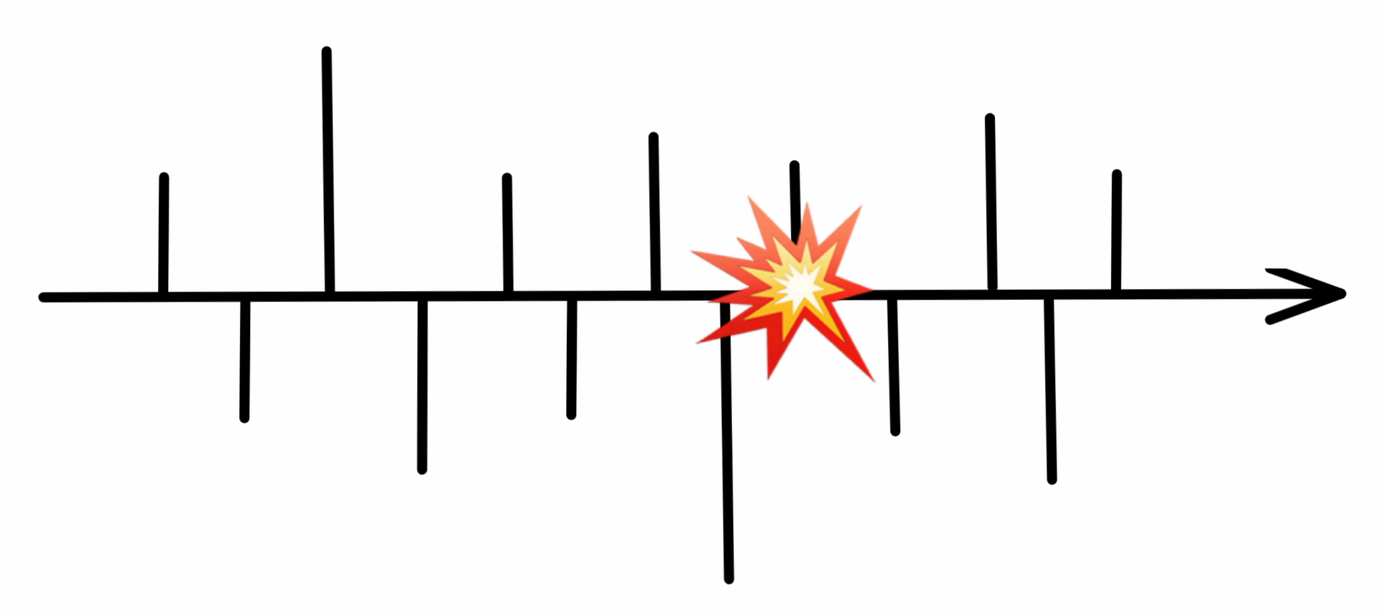
Lia Rahdiah
thenounproject.com

# Debrief



Cuputo
thenounproject.com

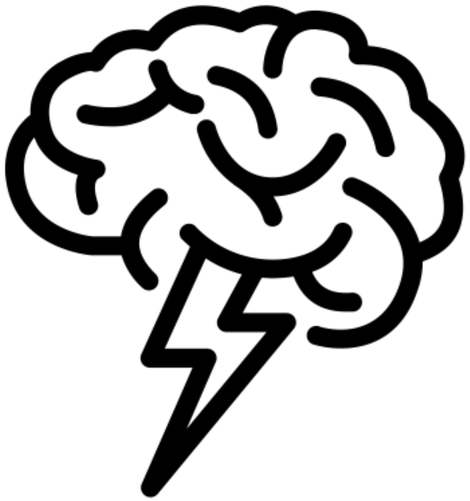
# Walking the Timeline



# Lessons

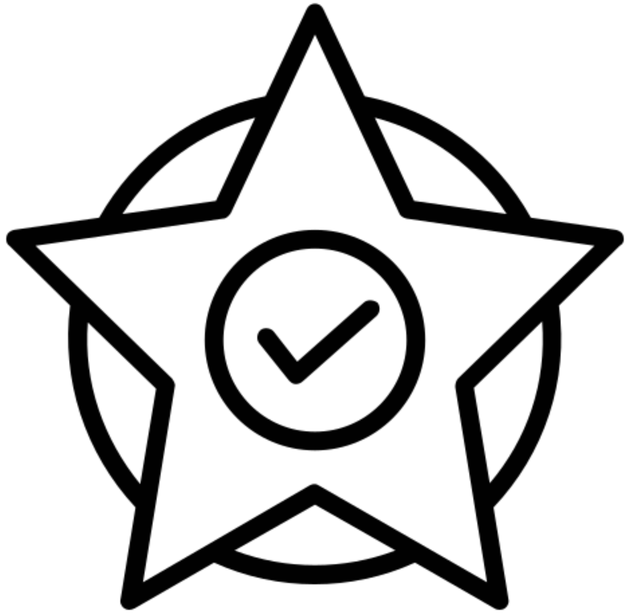
- What surprised us?
- What went well?
- What was difficult?
- Where did we get lucky?
- What don't we understand?

# Brainstorming



Fran Couto
thenounproject.com

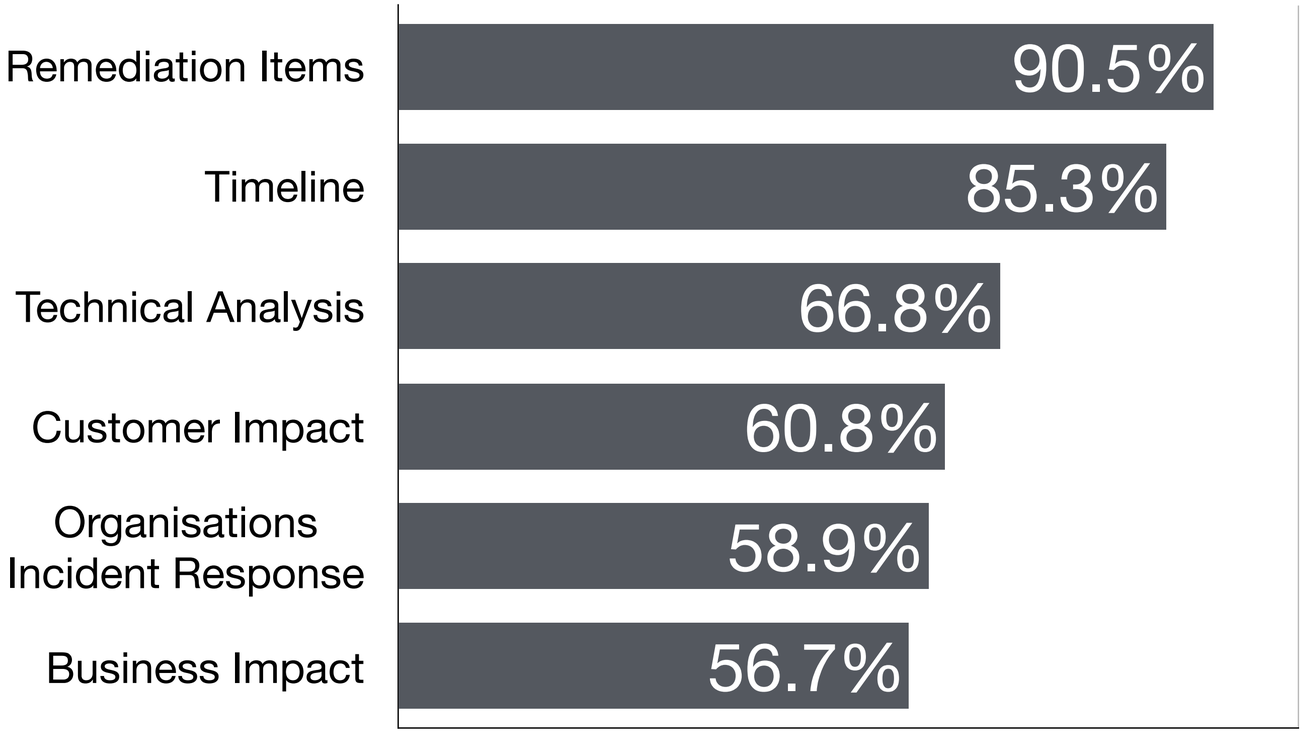
# Recommendations



Vectors Market
thenounproject.com

# Part 3: What’s missing and Improvements

# What’s in most PIRs?



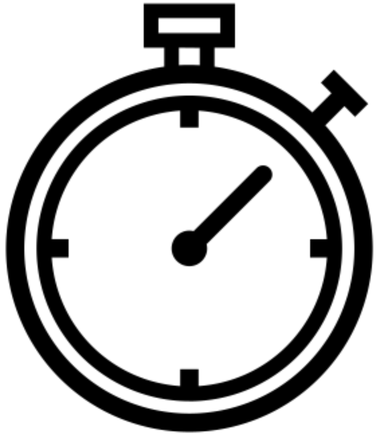

- J Paul Reed

Maps Context and Tribal Knowledge

learning > fixing

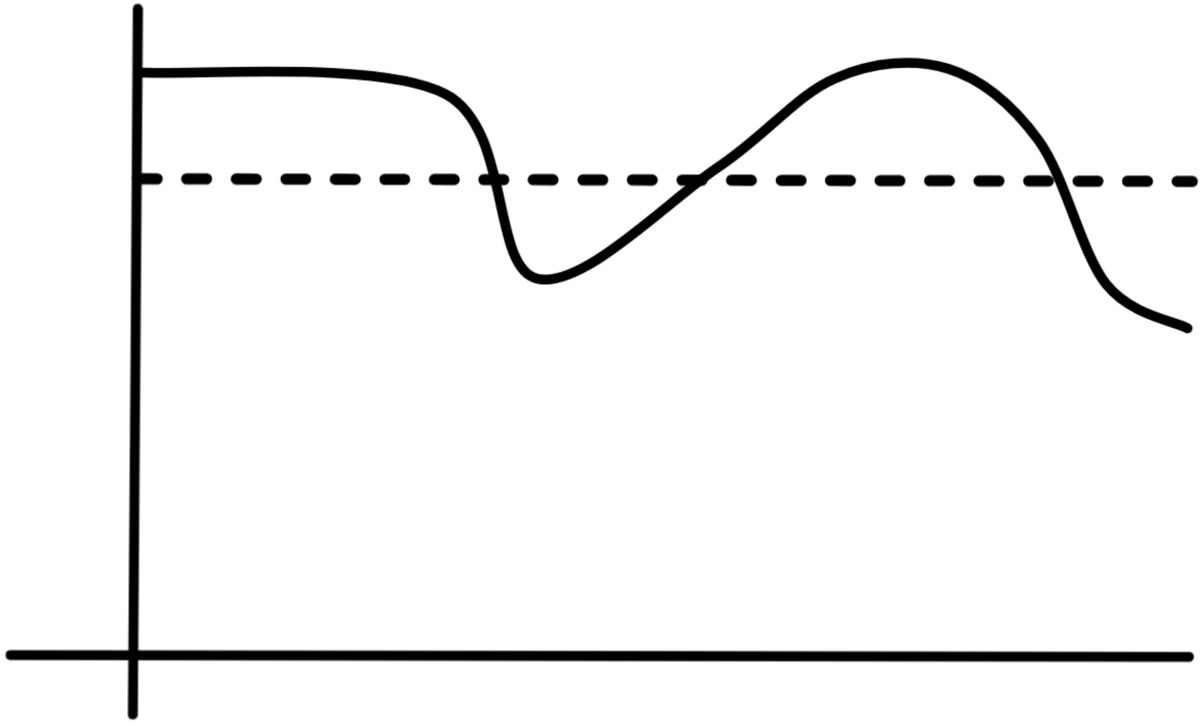
# Metrics

~~MTTx~~



Andri The Blue
thenounproject.com

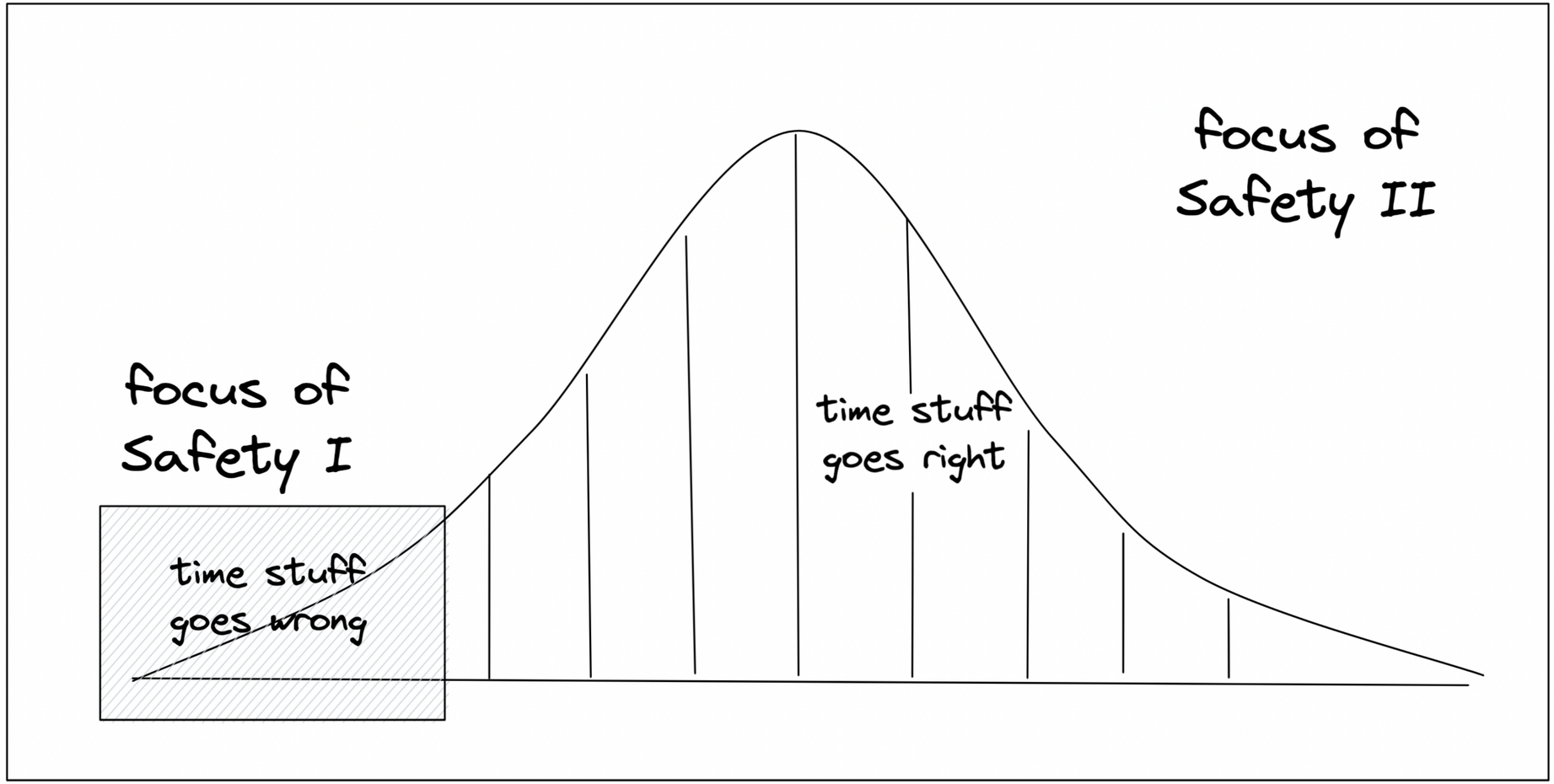
# Service Level Objectives



# Improvements

Kukuh Wachyu Bias
thenounproject.com

# What’s next?


focus of
Safety II
focus of
Safety I
time stuff
goes right
time stuff
goes wrong

Thank you

# Debrief Ground Rules

Thanks for joining, this is the debrief for the X incident which occurred on Y.

We are going to be blame-aware, we recognise that it’s natural to want to blame a bad outcome on a bad decision or action, but we know this isn’t useful.

So we’ll work from the assumption that no one comes to work to do a bad job, and everyone made the best decisions they could with the information they had.

We now know the outcome of these decisions, and hindsight bias means that these outcomes seem far more likely to us now than they did at the time. So if you find yourself being judgemental, try and be curious instead.

We want to avoid talking about counterfactuals, what people could have done, or should have done, and instead focus on what actually happened and try to put ourselves into their shoes and understand how they came to make the decisions and take the actions that they did.

It's my job as facilitator to try and keep us on time, so if we start going off-topic I might ask that we park some conversations for later.

This is a collaborative session, so please ask if you have any questions or more if you have more details about an event please add them.

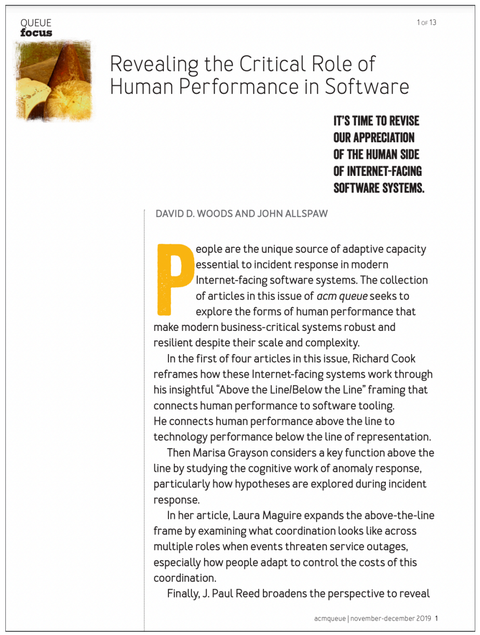
We're going to spend the first half of the meeting reviewing what happened by walking the timeline, and the second half is where we'll discuss what we've learnt and brainstorm some ideas that could help improve things for future incidents.

Any questions?

# Further Reading

## acmqueue: Human Factors

Vol. 17 No. 6 – November-December 2019



QUEUE focus

1 of 13

Revealing the Critical Role of Human Performance in Software

IT'S TIME TO REVISE OUR APPRECIATION OF THE HUMAN SIDE OF INTERNET-FACING SOFTWARE SYSTEMS.

DAVID D. WOODS AND JOHN ALLSPAW

People are the unique source of adaptive capacity essential to incident response in modern Internet-facing software systems. The collection of articles in this issue of *acm queue* seeks to explore the forms of human performance that make modern business-critical systems robust and resilient despite their scale and complexity.

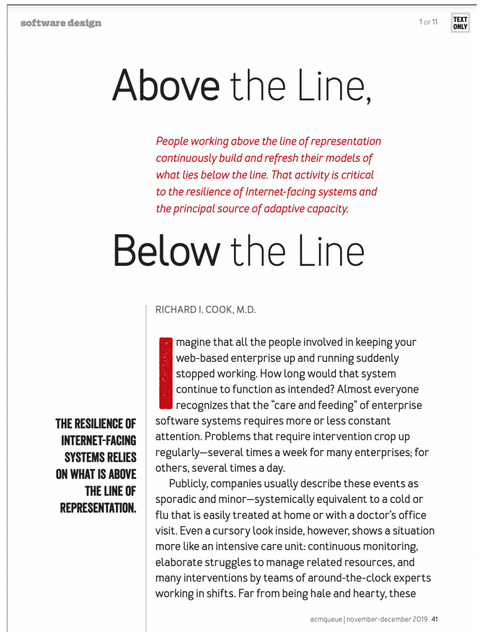
In the first of four articles in this issue, Richard Cook reframes how these Internet-facing systems work through his insightful "Above the Line/Below the Line" framing that connects human performance to software tooling. He connects human performance above the line to technology performance below the line of representation.

Then Marisa Grayson considers a key function above the line by studying the cognitive work of anomaly response, particularly how hypotheses are explored during incident response.

In her article, Laura Maguire expands the above-the-line frame by examining what coordination looks like across multiple roles when events threaten service outages, especially how people adapt to control the costs of this coordination.

Finally, J. Paul Reed broadens the perspective to reveal

acmqueue | november-december 2019 1



software design

1 of 11

TEXT ONLY

Above the Line,

*People working above the line of representation continuously build and refresh their models of what lies below the line. That activity is critical to the resilience of Internet-facing systems and the principal source of adaptive capacity.*

Below the Line

RICHARD I. COOK, M.D.

THE RESILIENCE OF INTERNET-FACING SYSTEMS RELIES ON WHAT IS ABOVE THE LINE OF REPRESENTATION.

Imagine that all the people involved in keeping your web-based enterprise up and running suddenly stopped working. How long would that system continue to function as intended? Almost everyone recognizes that the "care and feeding" of enterprise software systems requires more or less constant attention. Problems that require intervention crop up regularly—several times a week for many enterprises; for others, several times a day.

Publicly, companies usually describe these events as sporadic and minor—systemically equivalent to a cold or flu that is easily treated at home or with a doctor's office visit. Even a cursory look inside, however, shows a situation more like an intensive care unit: continuous monitoring, elaborate struggles to manage related resources, and many interventions by teams of around-the-clock experts working in shifts. Far from being hale and hearty, these

acmqueue | november-december 2019 41

quality assurance

1 of 19

TEXT ONLY

Beyond the "FIX-IT" Treadmill

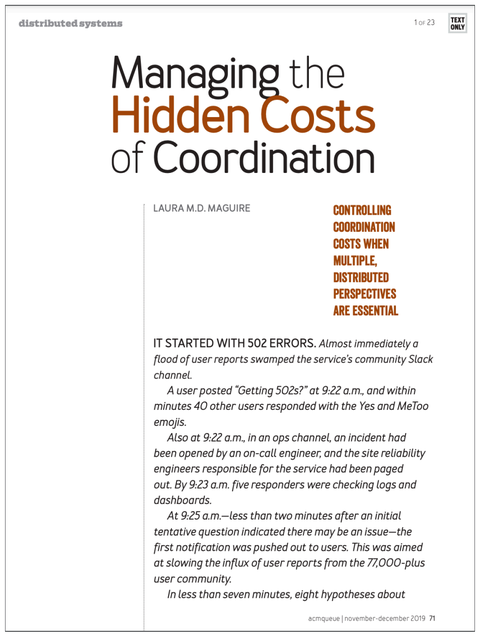
THE USE OF POST-INCIDENT ARTIFACTS IN HIGH-PERFORMING ORGANIZATIONS

J. PAUL REED

Of all the traits the technology industry is known for, self-reflectivity and historical introspection don't rank high on the list. As industry legend Alan Kay once famously quipped, "The lack of interest, the disdain for history is what makes computing not-quite-a-field." It is therefore somewhat cognitively dissonant, if not fully ironic, that the past few years have seen renewed interest in the mechanics of retrospectives and how they fit into the daily practice of our craft.

Of course, retrospectives are not new, in software development at least. For more than 15 years capital-A Agile software development methods have been extolling the virtues of a scheduled, baked-in reflection period at the end of each development sprint. (Whether these actually occur in organizations practicing Agile remains an open question.) Those same 15 years have also seen a tectonic

acmqueue | november-december 2019 94



distributed systems

1 of 23

TEXT ONLY

Managing the Hidden Costs of Coordination

LAURA M.D. MAGUIRE

CONTROLLING COORDINATION COSTS WHEN MULTIPLE, DISTRIBUTED PERSPECTIVES ARE ESSENTIAL

IT STARTED WITH 502 ERRORS. *Almost immediately a flood of user reports swamped the service's community Slack channel.*

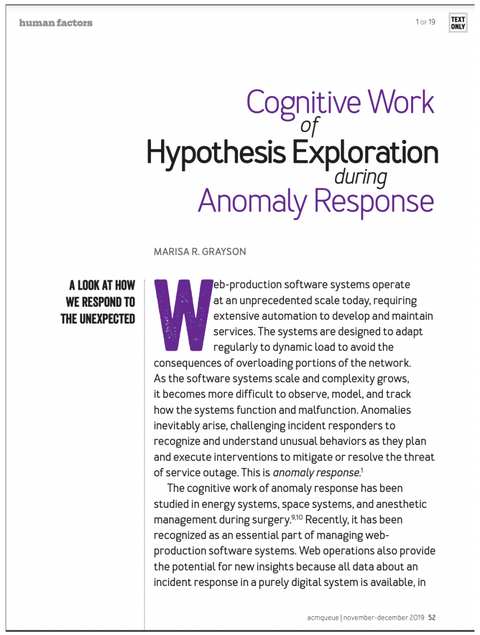
*A user posted "Getting 502s?" at 9:22 a.m., and within minutes 40 other users responded with the Yes and MeToo emojis.*

*Also at 9:22 a.m., in an ops channel, an incident had been opened by an on-call engineer, and the site reliability engineers responsible for the service had been paged out. By 9:23 a.m. five responders were checking logs and dashboards.*

*At 9:25 a.m.—less than two minutes after an initial tentative question indicated there may be an issue—the first notification was pushed out to users. This was aimed at slowing the influx of user reports from the 77,000-plus user community.*

*In less than seven minutes, eight hypotheses about*

acmqueue | november-december 2019 71



human factors

1 of 19

TEXT ONLY

Cognitive Work of Hypothesis Exploration during Anomaly Response

MARISA R. GRAYSON

A LOOK AT HOW WE RESPOND TO THE UNEXPECTED

Web-production software systems operate at an unprecedented scale today, requiring extensive automation to develop and maintain services. The systems are designed to adapt regularly to dynamic load to avoid the consequences of overloading portions of the network. As the software systems scale and complexity grows, it becomes more difficult to observe, model, and track how the systems function and malfunction. Anomalies inevitably arise, challenging incident responders to recognize and understand unusual behaviors as they plan and execute interventions to mitigate or resolve the threat of service outage. This is *anomaly response*.[1]

The cognitive work of anomaly response has been studied in energy systems, space systems, and anesthetic management during surgery.[9,10] Recently, it has been recognized as an essential part of managing web-production software systems. Web operations also provide the potential for new insights because all data about an incident response in a purely digital system is available, in

acmqueue | november-december 2019 52

**Dr. Johan Bergstrom**
Three analytical traps in accident investigation (mechanistic reasoning)

**Eurocontrol - Skybrary**
Local Rationality

**Dr Sidney Dekker**
Just Culture
Malicious Compliance
The Psychology of Accident Investigation

**Jens Rasmussen**
Risk management in a dynamic society: a modelling problem

**James Reason**
Human error: models and management

**Gary Klein, Neil Hintze, and David Saab**
Thinking Inside the Box: The ShadowBox Method for Cognitive Skill Development

**Rene Amalberti**
The paradoxes of almost totally safe transportation systems

**Erik Hollnagel, Jörg Leonhardt, Tony Licu, Steven Shorrock**
From Safety-I to Safety-II: A White Paper

**J Paul Reed**

Blameless postmortems don’t work. Here’s what does.

Maps, Context, and Tribal Knowledge: On the Structure and Use of Post Incident Analysis Artefacts in Software Development and Operations

**John Allspaw**

Blameless Postmortems

Each necessary, but only jointly sufficient

The infinite how’s, or the dangers of the five why’s

Moving past shallow incident data

The multiple purposes and audiences of post incident reviews

How learning is different from fixing

Trade-offs under pressure: Heuristics and observations of teams resolving internet service outages