CAST Incident Retrospectives

Moshe Zadka – https://cobordism.com

Acknowledgement of Country

Belmont (in San Francisco Bay Area Peninsula)

Ancestral homeland of the Ramaytush Ohlone people

I live in Belmont, in the San Francisco Bay Area Peninsula. I wish to acknowledge it as the ancestral homeland of the Ramaytush Ohlone people.

0.1 Basics

0.1.1 What is an incident?

What is an Incident?

Bad

Unplanned

Willing to stop

An incident is something that is:

- Bad: If it wasn't a problem, there's no issue.
- Unplanned: If the problem was *planned*, it is not an incident. This is not the same as saying the problem was *caused* by a planned thing. If a website was planned to shut down for three hours for an upgrade, and it shut down, this is fine. If an upgrade caused a three hour outage, it might be an incident.
- Willing to stop: An incident has to be something that, at least in principle, we are willing to put effort into stopping. If an earthquake took out our datacenter, and the risk of earthquakes is low, we might not be willing to put in the effort to stop it.

0.1.2 What is an incident retrospective?

What is an Incident Retrospective?

(Post-mortem)

Analyze

Improve

An incident retrospective is a process that allows us to analyze the incident and improve our future resilience based on the analysis. The process usually has both an async part, where someone does the research, analysis, and suggested improvement, and a synchronous part, where the analysis is validated and the team commits to the improvement plan.

0.1.3 Why is an incident retrospective?

Why is an Incident Retrospective?

Do better!

We do incident retrospectives because an incident teaches us about a lack of resiliency in our system. We want our systems to be more resilient. The "natural lesson" of an incident is a poor thing to waste.

0.1.4 What is root cause analysis?

What is root cause analysis?

Find cause...eliminate!

Root Cause Analysis, sometimes in the form of the "five whys", asks "what originally caused the problem?" The goal of an RCA is to *identify* the root cause, and then eliminate it.

Note that this is a crucial piece of RCA: the improvement is to *eliminate the root cause*. The theory underlying it is that any other step along the way would not have happened without the original problem.

0.1.5 Why not root cause analysis?

Why not root cause analysis?

More than one

Too specific

Not controllable

Why not do RCA-based analysis?

For one, there might be *more than one root* cause. In some situations, *two* things had to have happened: without either, the incident would not happen.

The root cause might also be *too specific*. If we eliminate the root cause, but another one can start the same chain of events, we have fixed the *wrong problem*.

Finally, the root cause might be *beyond our boundary of control*. If the root cause is "high customer traffic following a world-wide event", then unless we control the world, we cannot remove the cause.

Sometimes, RCA has morphed into "contributing cause analysis". This tries to address the issue by suggesting we look at "all" things that caused the incident. This is a patch on RCA which transforms a poor system to a poorly defined system.

Instead, the right way to fix the issue is to go back to the drawing board: how do we make systems more resilient? There is a field that deals with those issues: systems theory.

0.2 System Theory Basics

0.2.1 System

System

Components

You control

Systems theory, like any field, has some technical jargon. These are specific terms with precise meaning, different than their meaning in regular use.

A *system* is a collection of *components*. A component is only in the system if *you* control it.

This means, for example, your co-lo datacenter's power is *not* part of your system.

0.2.2 Environment

Environment
Influences behavior
Not system

The *environment* is anything that influences the behavior of the system but is not a part of it. This can include the people who use the system, the underlying network infrastructure, or even the weather!

0.2.3 Control

Control
Input
to output

A *control* is a part of the component. It is something that takes input, from the system or environment, or produces some result or output.

An example of a control is a thermostat. The input is a temprature, and the output is to connect or disconnect an electric circuit.

0.2.4 Loss

Loss
Undesired behavior

A system has things it should do. When it does not do these things, the technical term is "loss".

A loss can be "our website was unavailable for five seconds, so people could not see cat pictures" or "we leaked all customers credit cards and billions of dollars were stolen". A loss should always be something concrete that someone cares about.

0.2.5 Safety Control

Safety Control
Control
Prevent loss

A *safety control* is a control whose purpose is to prevent or reduce loss. For example, a *high availability load balancer* will prevent loss in the form of a site outage by routing to only healthy nodes.

A *safety control* can also be a fire alarm. While it does not prevent all loss from a fire, it alerts people to leave, thus preventing the loss of human life.

0.3 CAST Pt 1.: Analyze

0.3.1 Hazard

Hazard
Input to system
led to problem

When analyzing the system, the *hazard* is an input to the system that led to the problem. This is a little like the "RCA", but has a more concrete definition:

- It must come from outside the system
- There can be multiple hazards

The job of CAST is not to prevent the hazard. *Knowing* the hazard is important regardless.

0.3.2 Timeline

Timeline
What happened
When
The timeline should list everything relevant to the incident. The hazard does not need to be the first thing!

Dates of changes to the system that are relevant, for example, often predate the hazard. The timeline should not end before the incident was completely mitigated.

0.3.3 Loss

Loss
Why it was bad?

The next step is to explain the specific loss. There was no incident if nobody was harmed. Detail the specific loss incurred.

This can range anywhere from "three people lost a work hour each restoring the system" to... well.

0.3.4 Loss Analysis

Loss Analysis
Causal path
Hazard
Loss
The loss analysis should be based on the timeline. It should analyze the causal path that links the hazard to the incident. Each step in this should be documented in the timeline.

0.3.5 Safety Control Failures

Safety Control Failures
For each safety control:
Why it didn't prevent loss
The hazard would not lead to the loss if the safety controls performed correctly. For each relevant safety control, indicate why it failed to prevent the loss.

0.4 CAST Pt. 2: Improve

0.4.1 Safety Control Problems

Safety Control Problems

Suggest fixes!

For safety control that failed to prevent the loss, explain what problem led to failing to do so. Indicate how it could be fixed.

0.4.2 Missing Safety Controls

Missing Safety Controls

Suggest implementation!

Often, the incident shows the lack of appropriate safety controls. What additional safety controls were missing that would have prevented the loss?

0.4.3 Systemic Problems

Systemic Problems

Process

Incorrect assumptions

Finally, this is a catch-all category. What systemic problems in our processes led to this?

For example, safety controls might be of poor quality because they were rushed. Alternatively, maybe lack of testing of the controls is the issue.

0.4.4 Improvement Plan

Improvement plan

Concrete

Specific

List

An improvement plan should be written based on the issues above. Each item in the improvement plan should be *concrete* and *specific*. Concrete means that it is something someone can do. Specific means that it is clear when it has been done.

List all such items under the improvement plan.

0.5 Summary

0.5.1 Plan

CAST: Plan

Process ready

An incident is going to happen. Plan your CAST-based process before, so you know what do do after.

0.5.2 Apply

CAST: Apply

Incident?

Assign

Review

Act

After an incident, assign someone to do the CAST investigation. Once the report has been written, have the team review it to see if anything is missing or incorrect. After the team signs off on it, implement the improvement plan.

0.5.3 Improve

CAST: Improve

CAST is a Safety Control

CAST itself can be thought of a safety control. Improvement plans should check previous incident retrospectives, and see if anything needs to be done differently.