Picking up the pieces

A guide to Post Incident Review

Picking up the pieces

A guide to Post Incident Review

Klee Thomas

Clean code enthusiast

Code Crafter

Lover of stupid shirts

Organiser of Newcastle Coders Group

Senior Software Developer at nib health funds

@kleeut

Agile

Pairing

Clean Code

TDD

Dev Ops

Continuous Integration

Continuous Delivery

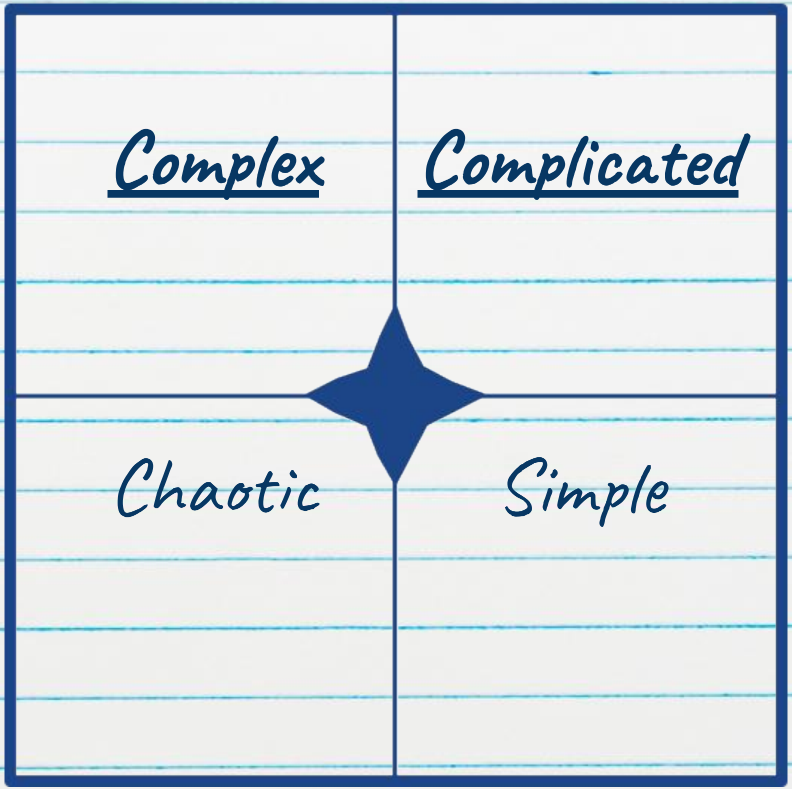
Etc

Something is going to go wrong

Our customers expect more from our software

We are building systems that are more complicated and complex.

Cynefin



Complex

Complicated

Chaotic

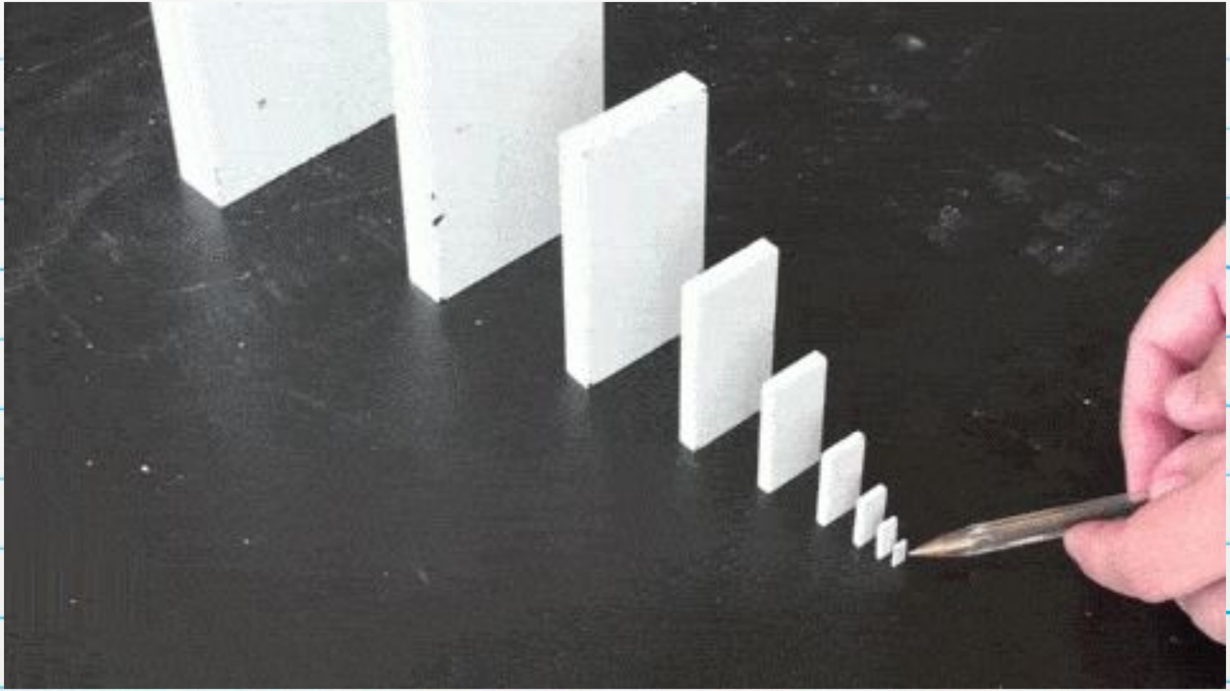
Simple

Something is going to go wrong

Our workforce is more and more transient.

Something is going to go wrong.

Create a prepared culture



@kleeut

Post Incident Review (PIR)

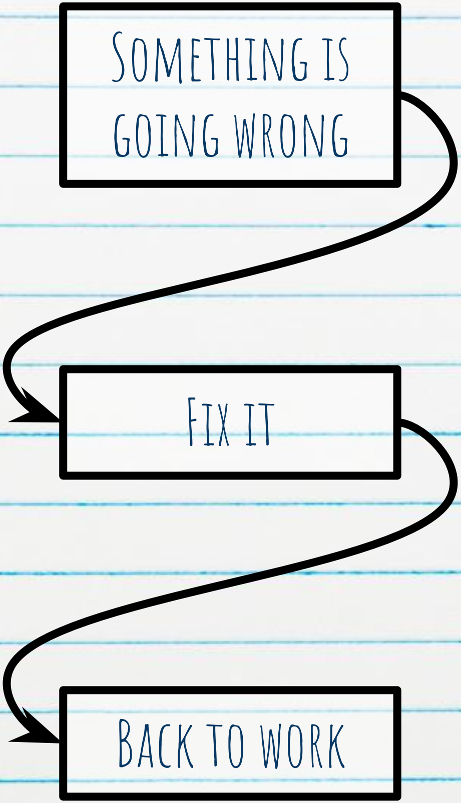
ANALYSIS OF AN INCIDENT

Exposing

Reflection on:

- What happened
- What went wrong
- How we responded
- How we can improve

The Flow of an incident

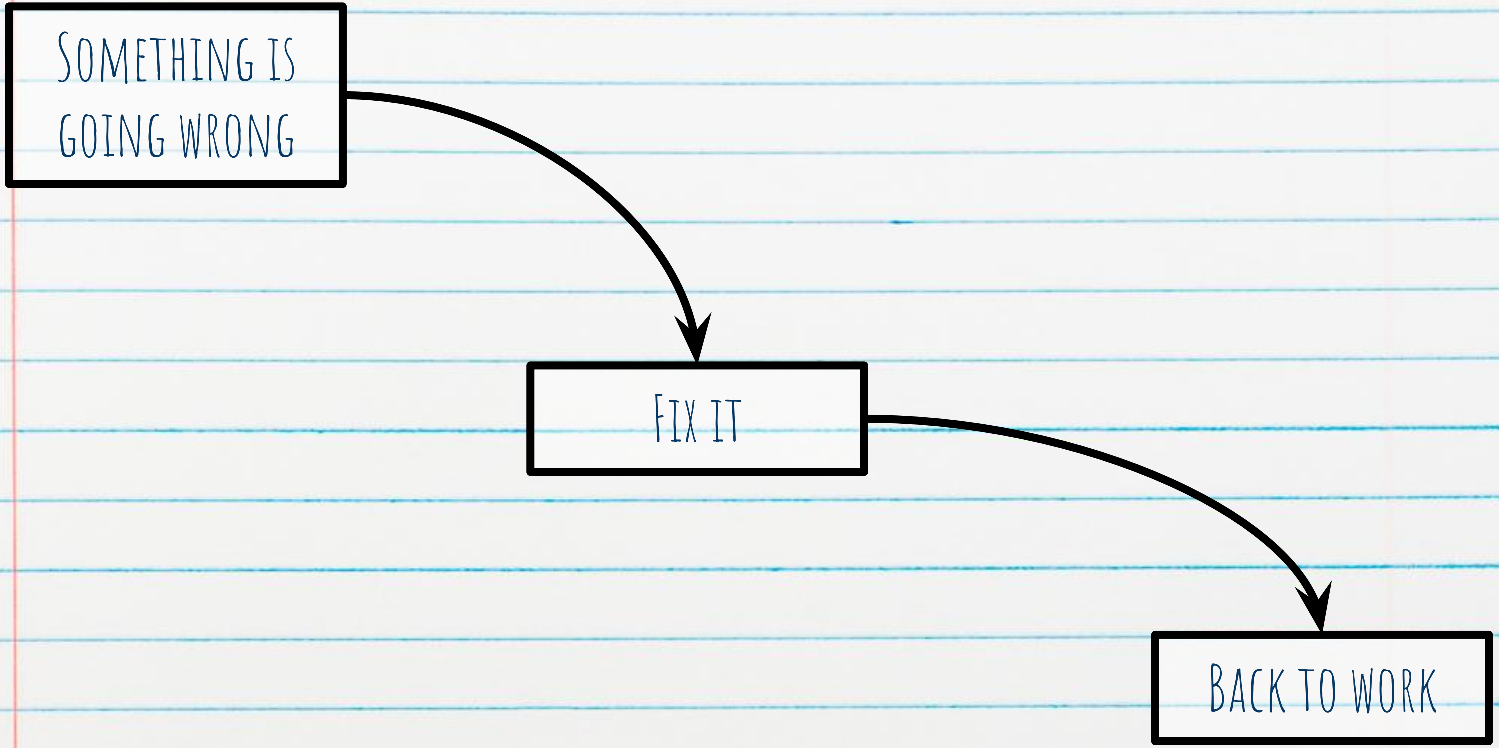


Something is going wrong

Fix it

Back to work

@kleeut



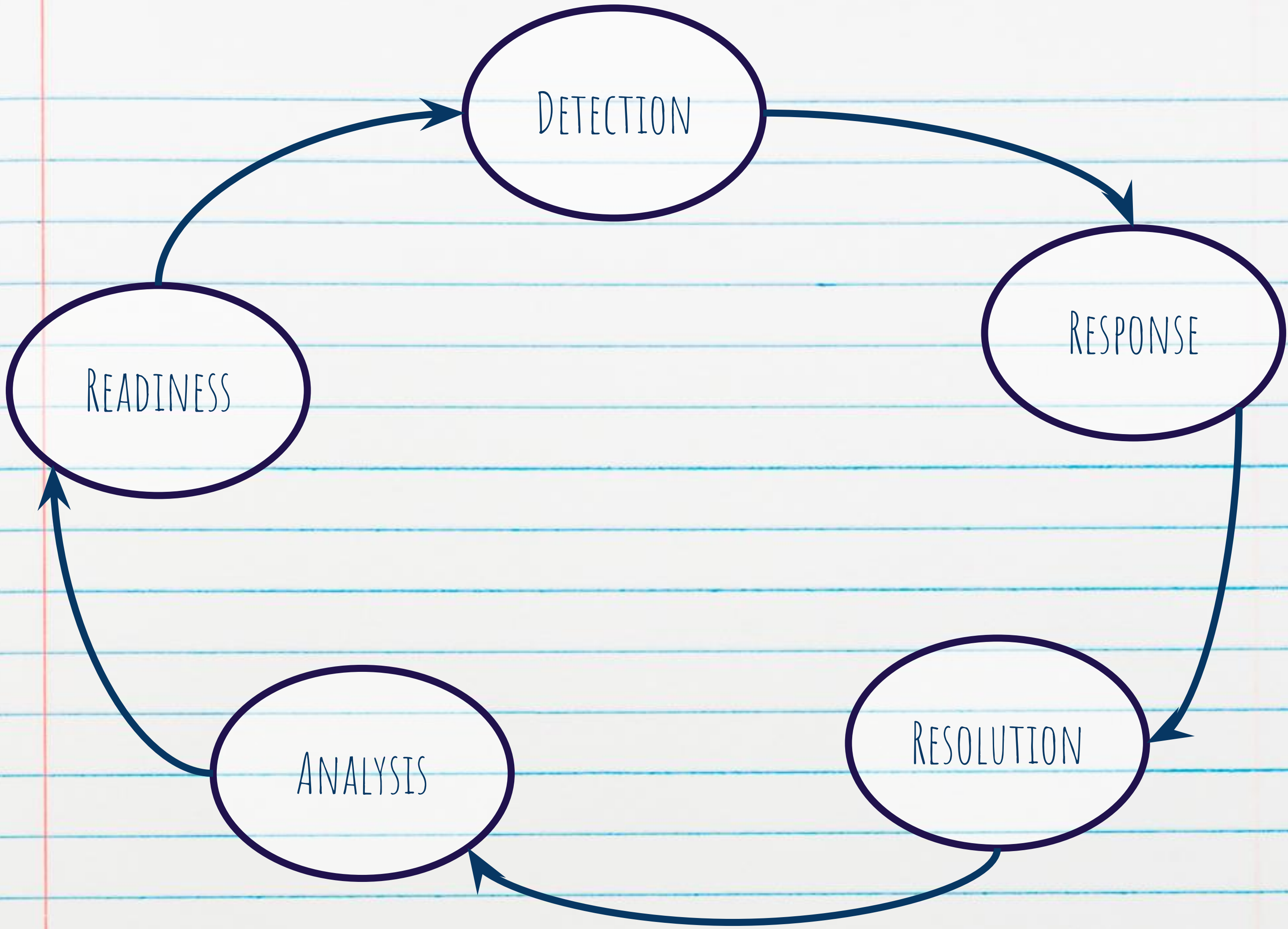
Something is going wrong

Fix it

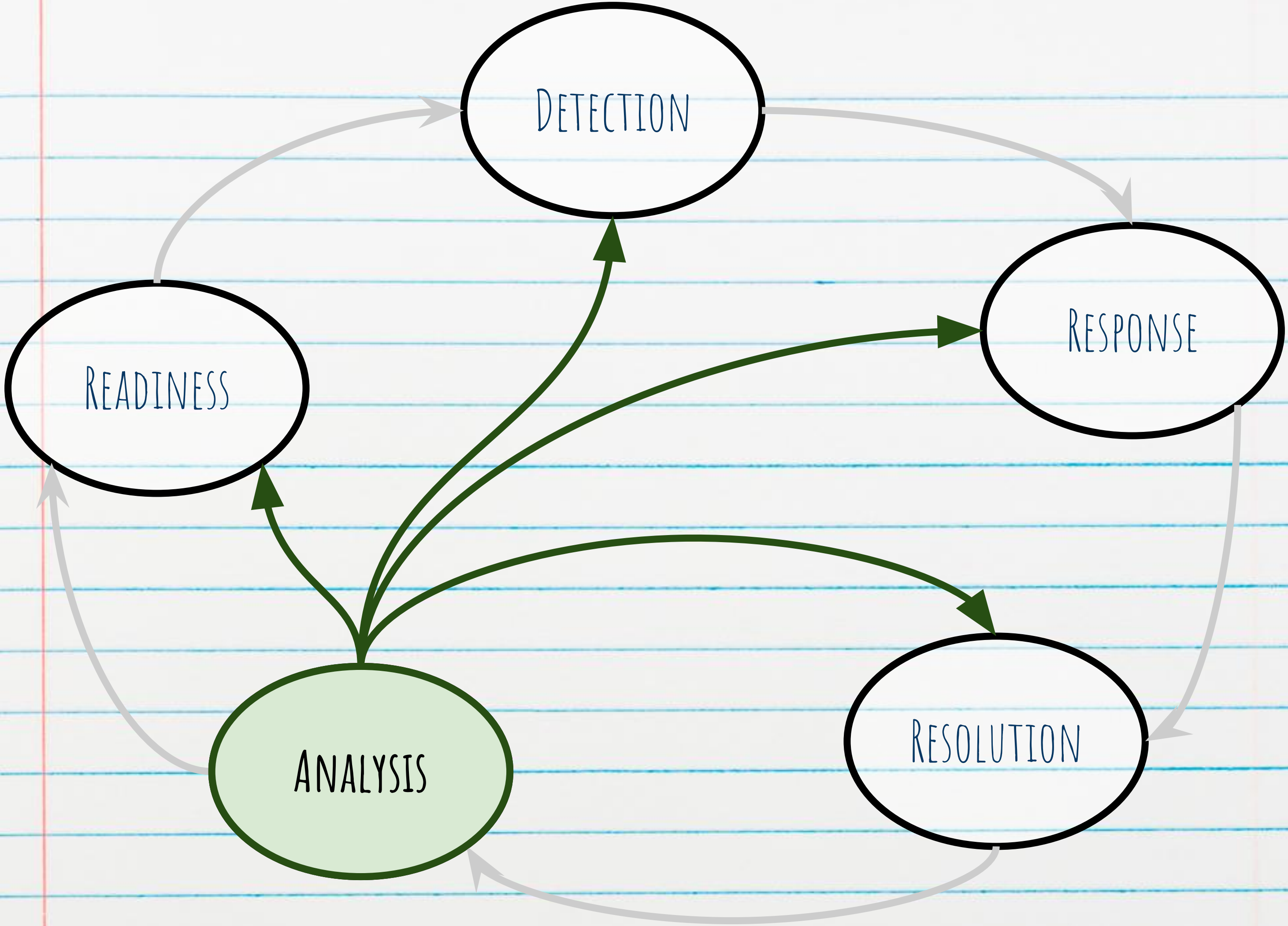
Back to work

@kleeut

Incident Life Cycle



@kleeut



@kleeut

When to run a PIR

As soon as possible

As Soon As Possible

Memory fades

We make fake memories

Within 2 days of resolution

Regularly

Do this for large and small incidents

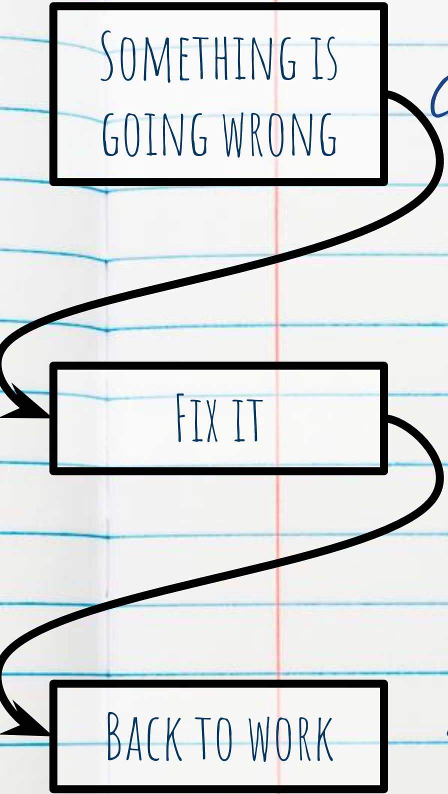
We learn more about the weaknesses in our system

We get practice at running reviews.

Path to great Post Incident Review

@kleeut

Example



Something is going wrong

Customers stopped being able to access https://klees-example.com.

Fix it

Ops added more disk space to the virtual machine.
Ops rebooted the server.
Customer requests went back to being fulfilled.

Back to work

Back to work

Root Cause Analysis

5 Whys

A great technique for Root Cause analysis

Get beyond the immediate answer

Just keep asking "Why?"

Why did the site go down?

- No disk space.

Why?

- Too many logs

Why?

- No log rolling

Why?

- Using a custom log manager

Why?

- John didnt want another dependency

- No disk space.

Why?

- Nobody added more space

Why?

- We didnt know space was low

Why?

- Bill turned off alerts

Why?

- Too many alerts over night

5 Whys - problems

No repeatable outcome

Root Cause analysis can lead to blaming an individual.

BLAME

Blame is natural and human

Blame happens when we're in pain

Blame leads to fear

Fear leads to hiding/misrepresenting facts

BLAME

If you dont blame a successful product launch on one person, why would you blame a failure on one person?

Don't blame the person

Blame the process, not the people - Edward Deming

The Prime Directive

> *Regardless of what we discover,*
> *we understand and truly believe that everyone did the best job they could,*
> *given what they knew at the time,*
> *their skills and abilities,*
> *the resources available,*
> *and the situation at hand.*

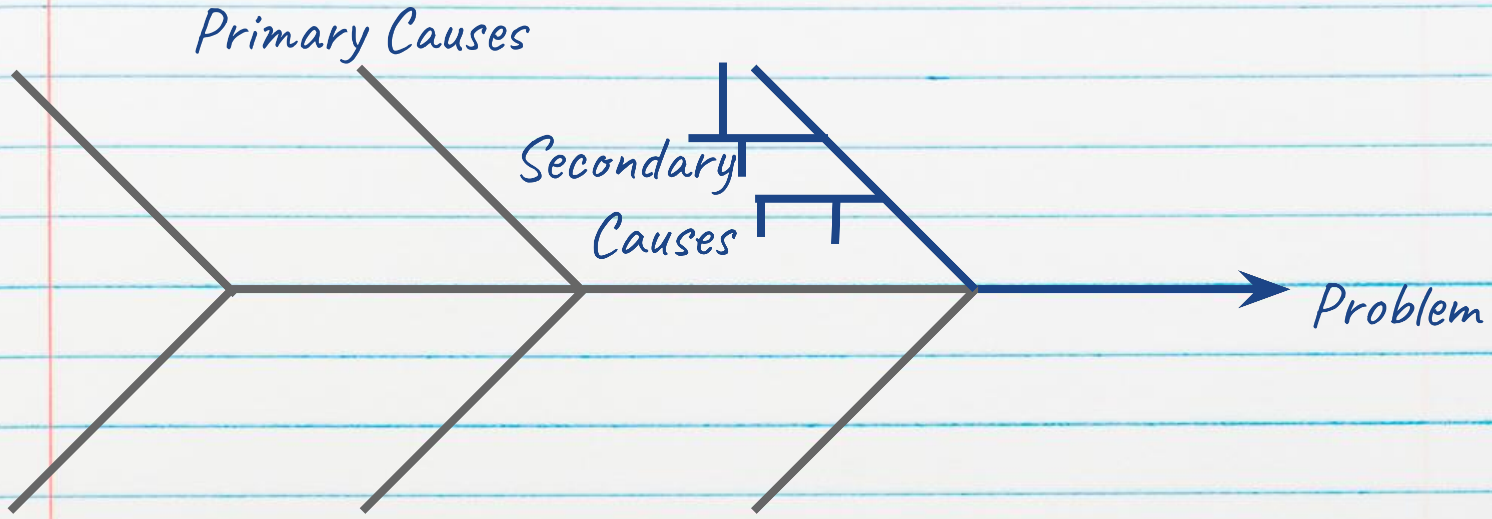
-Norm Kerth, Project Retrospectives: A Handbook for Team Review

@kleeut

Contributing factors

@kleeut

Ishikawa / Fishbone / Cause & Effect Diagram



@kleeut

Categories

6 “M”s - Manufacturing

Machines

Methods

Materials

Mind (People)

Measurement

8 P’s - Product Marketing

Product

Price

Promotion

Place

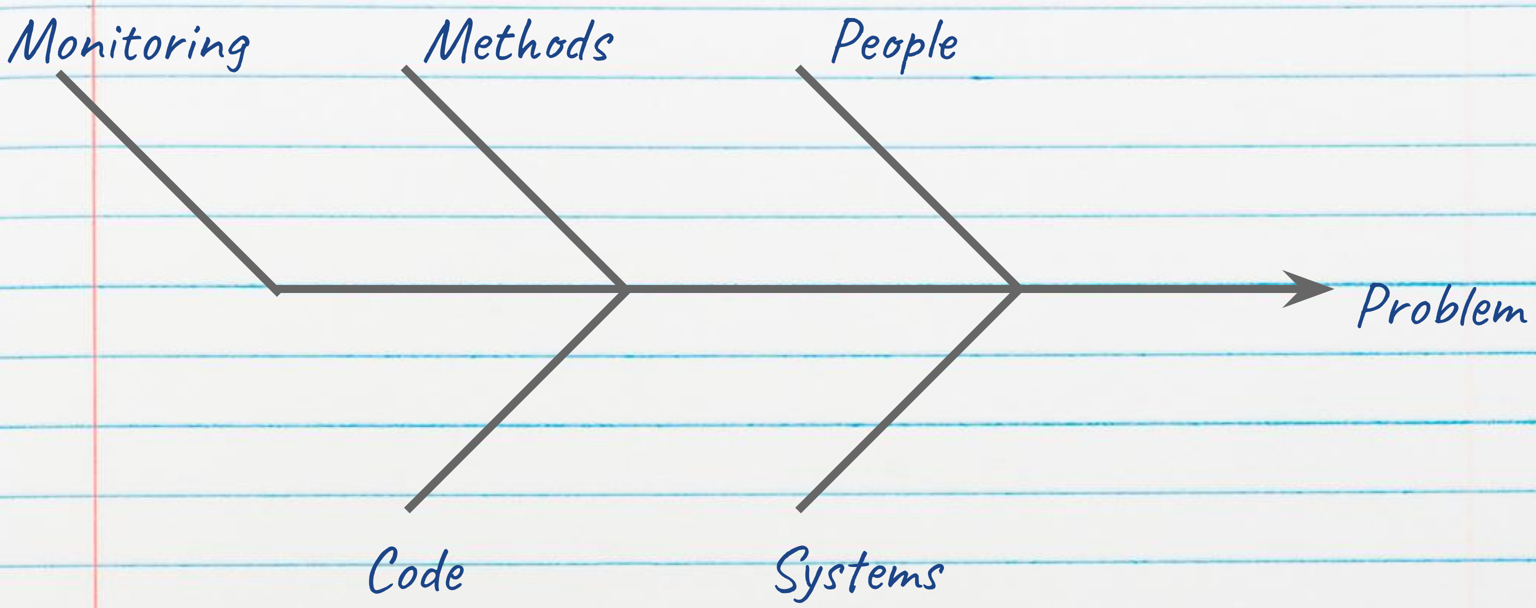
Process

People

Physical Evidence

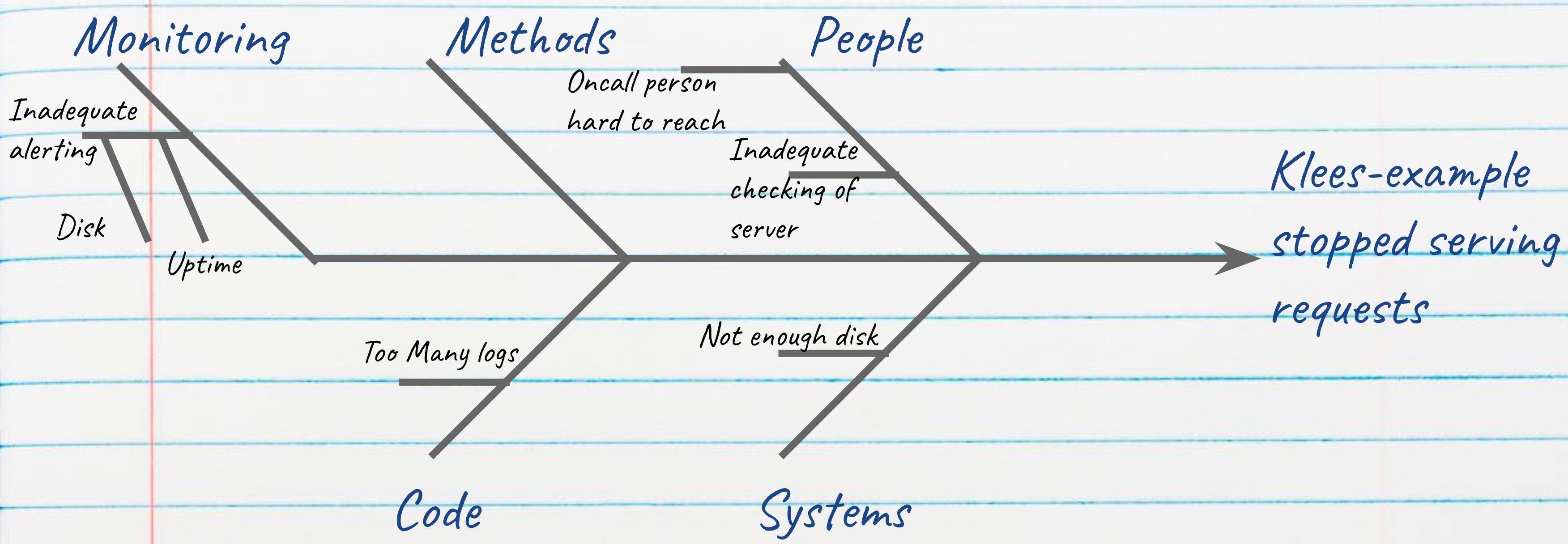
Performance

Ishikawa / Fishbone / Cause & Effect Diagram



@kleeut

Ishikawa / Fishbone / Cause & Effect Diagram


Monitoring
Methods
People
Inadequate alerting
Disk
Uptime
Oncall person hard to reach
Inadequate checking of server
Klees-example stopped serving requests
Too Many logs
Not enough disk
Code
Systems

@kleeut

Heuristics/Bias

- Subconcious
- Problem solving shortcuts
- Save time
- Make things more important than they are
- Risk ignoring valuable learnings

BIAS

Anchoring

- The first piece of evidence is the most relevant

Availability

- I can think of it therefore it's true

Confirmation

- Just because the outcome was good doesn't mean it was a good decision

@kleeut

BIAS

Hindsight

- The answer is obvious... If you know the answer

Outcome

- Could of, should of, why didn't

Bandwagon Effect

- Getting swept up in the crowd

@kleeut

How I run a PIR

The Prime Directive

> Regardless of what we discover,
> we understand and truly believe that everyone did the best job they could,
> given what they knew at the time,
> their skills and abilities,
> the resources available,
> and the situation at hand.

-Norm Kerth, Project Retrospectives: A Handbook for Team Review

@kleeut

Summary

Incident TL;DR;

Outline what happened

What was the resolution

What happened

Objective Timeline

Multiple points of view

- People Involved
- Automated Systems
- Chat Logs

Elaborate

Don't hide what happened

- What happened
- What did we do

Don't ask why X happened

- ask how it happened
- what factors informed the decision

Key Metrics

Who was involved

- Incident Commander
- Contributors

Time to Acknowledge:

Time to Recover:

Elapsed Time in each phase (Detection, Response, Remediation)

Severity: (e.g. fatal, critical, moderate, low, false alarm)

Example

Summary:

On January 13 klees-example.com stopped serving requests. We were able to get it back on line within 20 minutes by allocating more disk space to the server.

Timeline

2019-01-12 23:30 - Logs show Disk utilisation passes 90 %

2019-01-13 09:30 - Logs show 503 responses start occuring in the routers

2019-01-13 09:35 - Logs show No 200 responses in routers at all

2019-01-13 09:40 - Customer calls service desk

2019-01-13 09:41 - Service desk contacts dev via Slack

2019-01-13 09:43 - Devs refer to Ops via Slack

2019-01-13 09:45 - Ops identify 100% disk usage on vmke01

2019-01-13 09:46 - Ops increase virtual disk space by 15%

2019-01-13 09:47 - Ops restart server

2019-01-13 09:49 - Logs show 200 responses in routers

@kleeut

Who was involved

- @Jane, @Bill, @Fred

Time to Acknowledge: 11 minutes

Time to Recover: 20 Minutes

Elapsed Time in each phase:

- Detection: 11 Minutes,
- Response: 3 Minutes,
- Remediation: 4 Minutes

Severity: Fatal

What went well?

For all the bad stuff something must have gone well.

Look at all the phases.

How can you be more ready

What could we improve?

There are going to be areas that didn't work so well.

Be aware of blame.

- Understand what lead to actions.
- Identify processes that may have failed or been missing.

Look at all the phases

How can you be more ready

Action Items

Document them as they come up (Parking Lot)

Small or large, Immediate and long term

Commit to some, but not necessarily all.

Add them to your issue trackers, Assign them

Feed back into all stages of the life cycle.

Overview

The incident lifecycle:

Detection -> Response -> Remediation -> Analysis -> Readiness.

Avoid blame with an objective and honest timeline of events

Identify what went **well** and what went **poorly**

Track your actions

Run reviews often even on small things

@kleeut

Klee Thomas

@kleeut