

SRE NEXT 2022

Sensible Incident Management for Software Startups

Takayuki Watanabe
@Launchable, Inc.

Who?

Name: Takayuki Watanabe

Affiliation: Launchable, Inc.

Role: Software Engineer

Sns:

Blog: blog.takanabe.tokyo

Github: takanabe

Twitter: @takanabe_w

Interests:

- Developer Productivity
- Site Reliability Engineering
- Sustainability Engineering

Your takeaways

You can understand:

- Incident management has a life cycle.
- Incident response roles and structures exist to embody 3T mental models.
- Choosing strategies and tools makes incident managements at startups sensible.

Out of scope

- Fundamental SRE terminology (e.g. SLO, SLI, Error budget, Postmortem)

Disclaimer

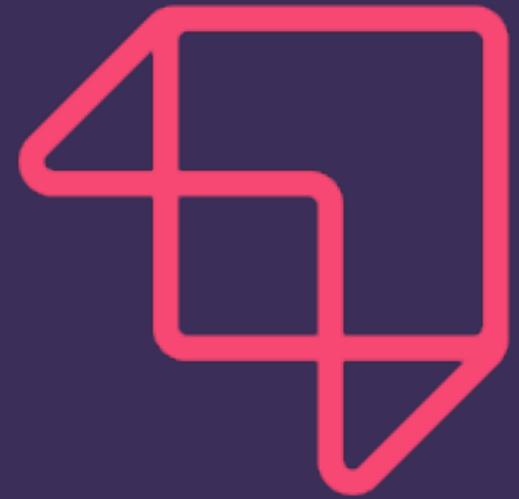
- This session refers a lot of existing incident management and SRE practices.
- But contains a lot of opinionated ideas and philosophy as well.
- So, the ideas might contradict to some people's.
 - Let's discuss on Twitter using #srenext with @takanabe_w

Today's agenda

- About Launchable
- Does a startup need incident management?
- Dissect incident management practices.
- 3T mental models and life cycles
- How can we improve incident management?
- Choosing right strategies and tools

Chapter 1:

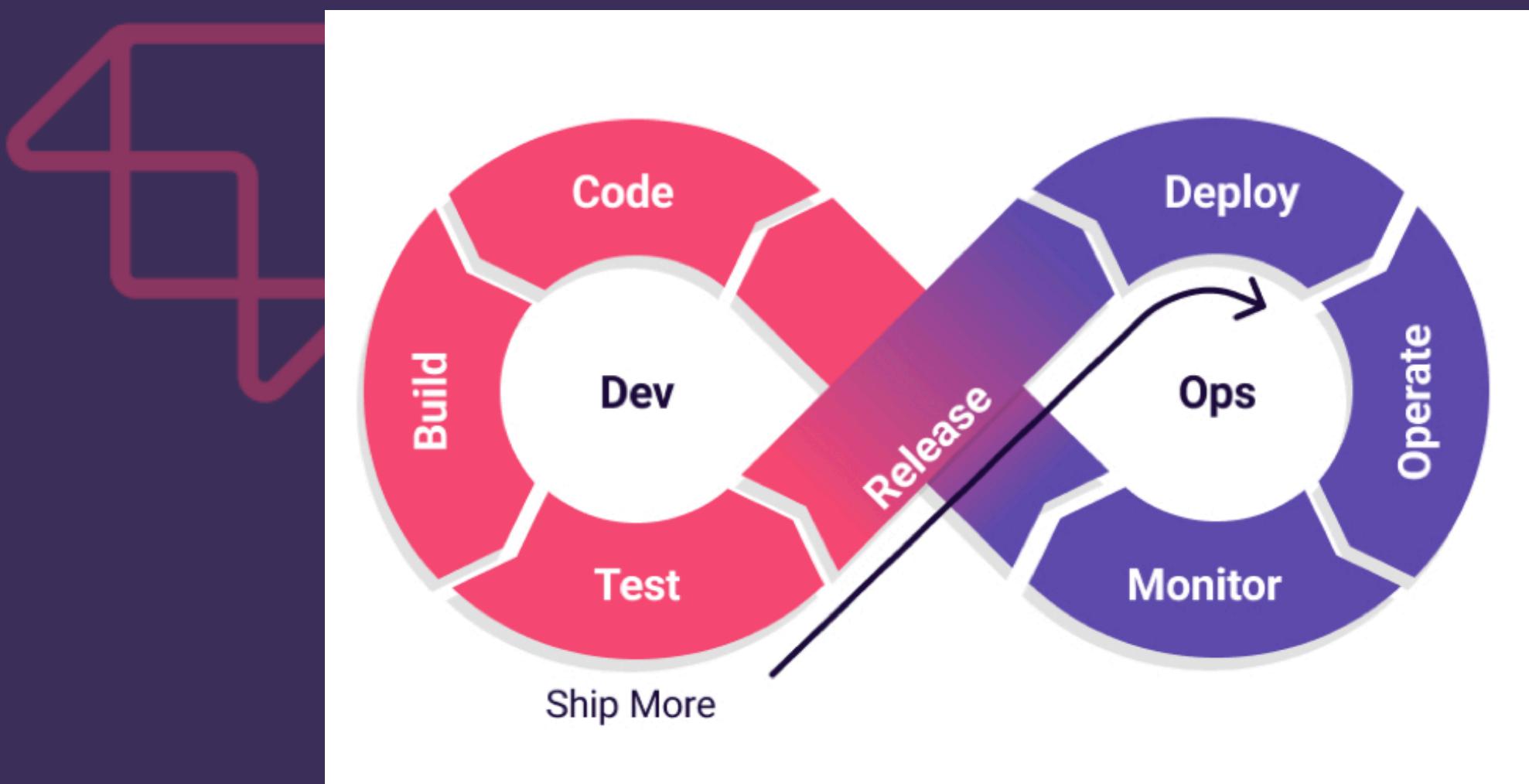
About Launchable



Launchable

What is Launchable?

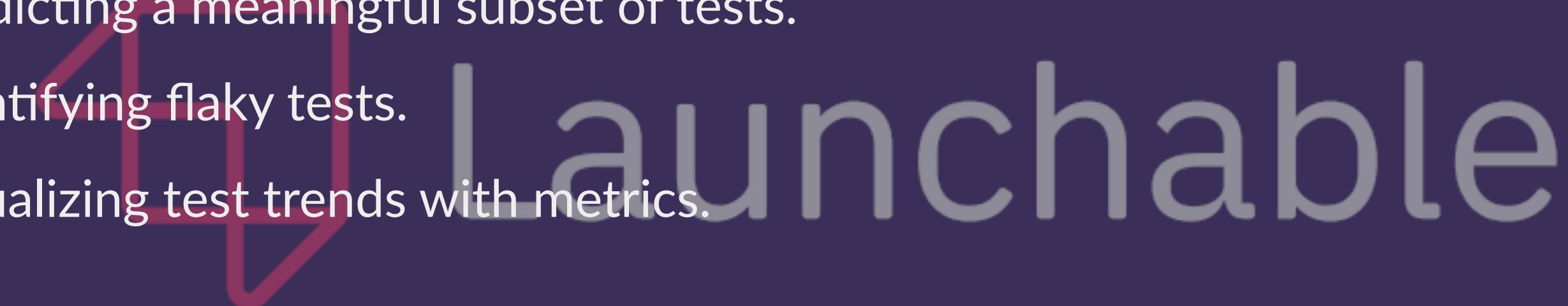
A SaaS accelerating software development cycles.



What is Launchable?

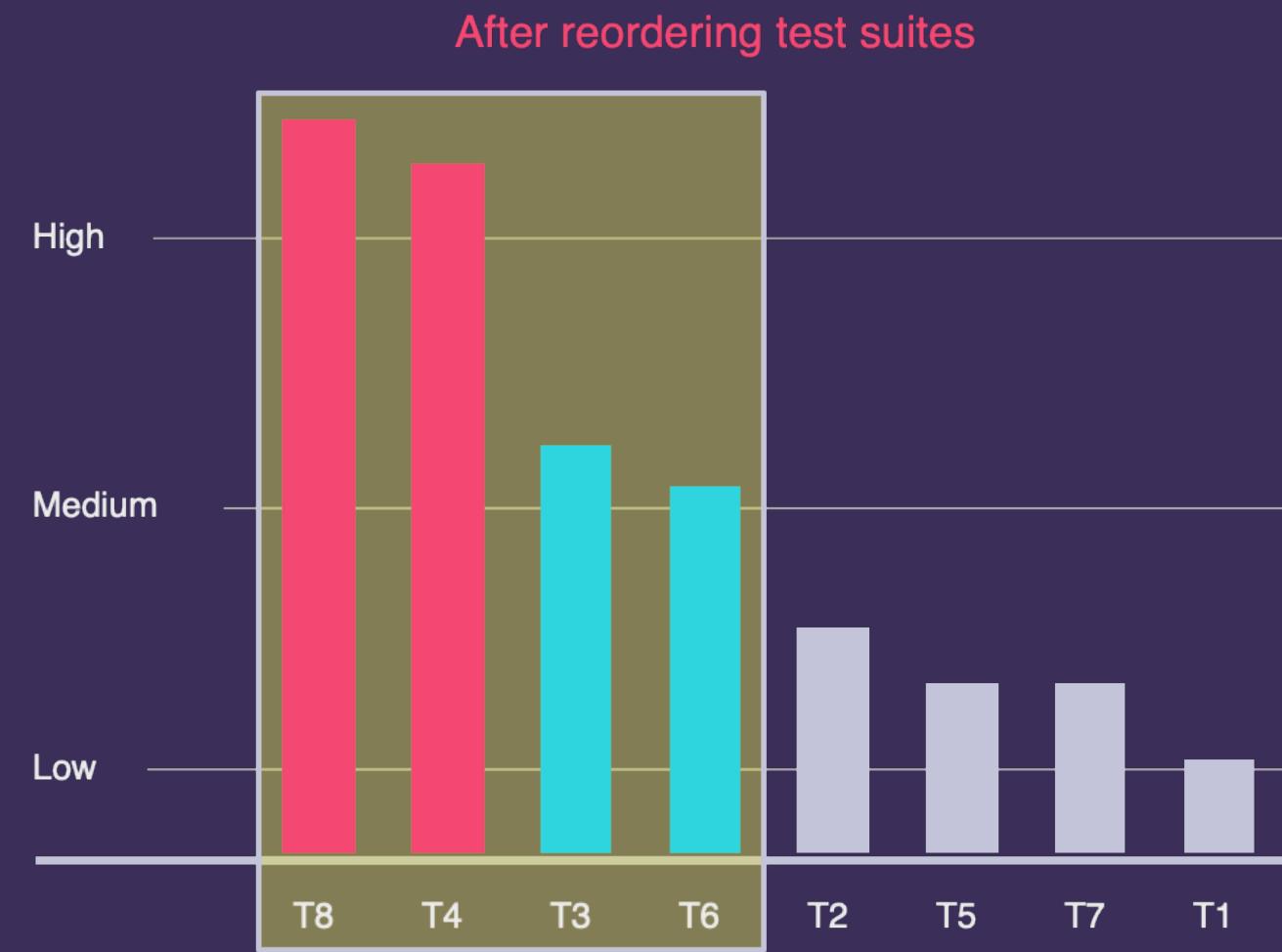
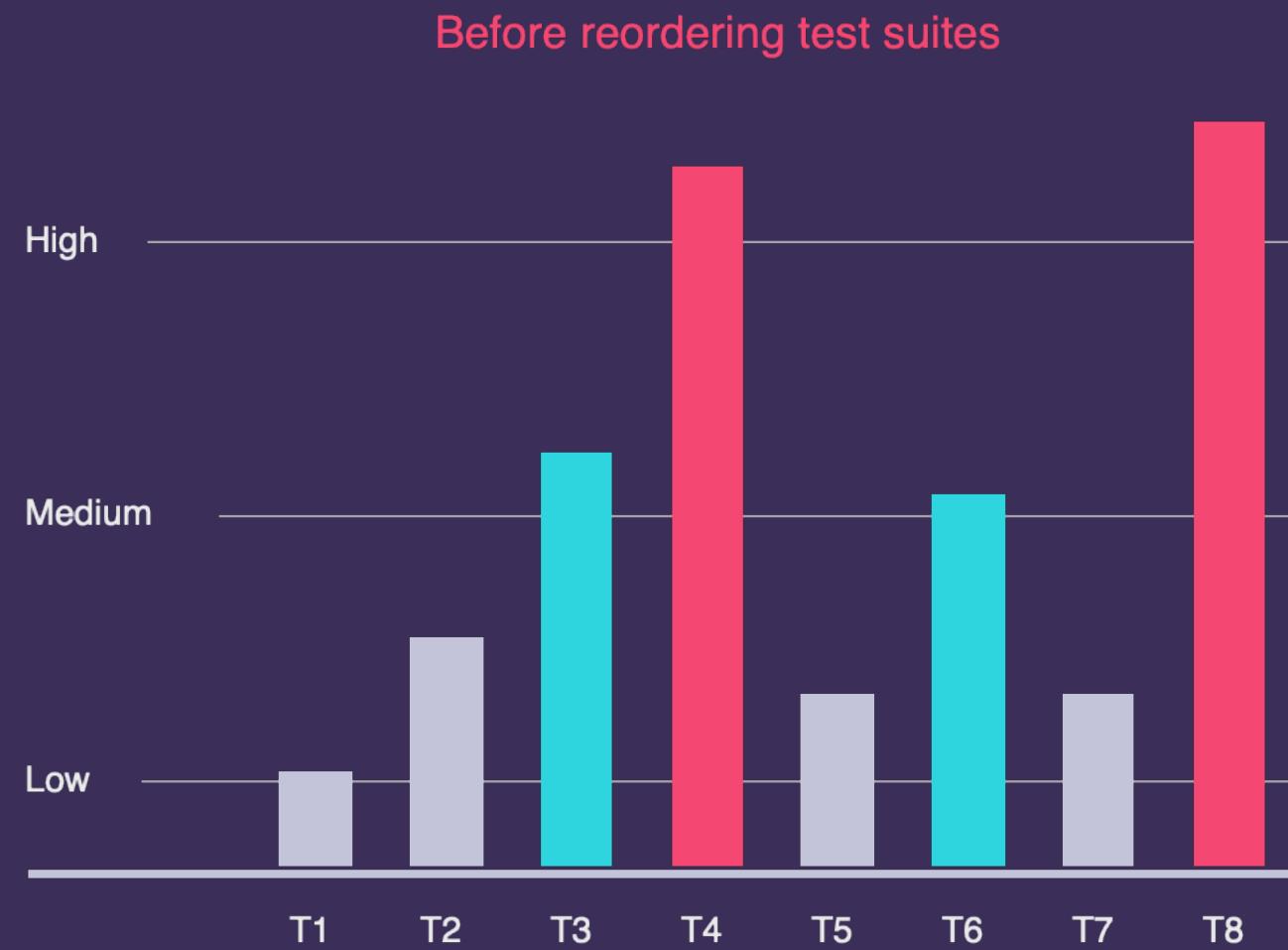
Current focus is **machine learning based test selections** by:

- Predicting a meaningful subset of tests.
- Identifying flaky tests.
- Visualizing test trends with metrics.



What is Launchable?

e.g. Reordering tests based on likelihood of failures.



Our team size

- Launchable is a startup
- **2 CEOs + 15 employees**
 - Software engineer (7 people)
 - Product manager
 - Marketing
 - Sales
 - etc...

Phases and the number of software engineers

Note: the numbers are estimated by the presenter based on previous experiences.

- Phase 0: Founding ~ 4 software engineers
- Phase 1: 5 ~ 10 software engineers
- Phase 2: 11 ~ software engineers



My SRE NEXT 2022 is about ...

Incident management at **software startups**

Does a startup need incident management?

Yes, it's obvious if products have customers.

Do you have enough engineering members?

No! but...

Learning from previous careers¹

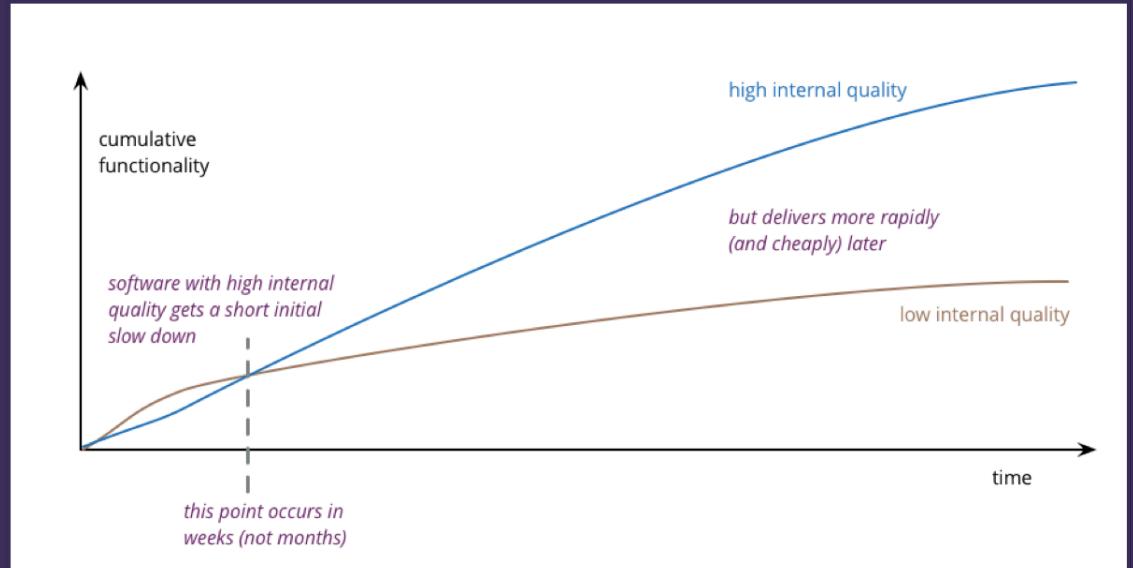
- I've worked at various sizes and stages.
 - Company A: **+300,000 people**
 - Company B: **+400 people** (Joined when they had +300 people)
 - Company C: **+150 people** (Joined when they only had less than 10 people)
- Product developments are always the highest priority concerns.
- Operation improvement != Product development velocity degradation.
- **We will never have enough engineering members to improve operations. Never.**



¹ SRE NEXT 2020: Designing fault-tolerant microservices with SRE and circuit breaker centric architecture

Are speed and quality trade-off?

- I personally don't think so^{2 3}.
- I believe sensible incident management accelerates our development velocity.



² martinFowler.com: Is High Quality Software Worth the Cost?

³ A Philosophy of Software Design, Chapter 3: Working Code Isn't' Enough, pp. 13 - 18.

Can we reframe the original question?

- We want to reframe "Does a startup need incident management?" to:
 - Which incident management processes won't change even for rapid developments?
 - Which processes should we improve?
- Let's dissect incident management practices in the industry.

Chapter 2:

Dissect incident management practices

What is incident management?

Incident management

- High level and overall process for handling incidents in an organization.

Incident response

- Part of incident management for actual technical steps including detection, reporting, mitigation, and recovery during incidents.

Existing practices

PagerDuty Incident Response

Incident Response > About

PagerDuty

Home This site documents parts of the PagerDuty Incident Response process. It is a cut-down version of our internal documentation, used at PagerDuty for any major incidents, and to prepare new employees for on-call responsibilities. It provides information not only on preparing for an incident, but also what to do during and after.

Getting Started Few companies seem to talk about their internal processes for dealing with major incidents. We would like to change that by opening up our documentation to the community, in the hopes that it proves useful to others who may want to formalize their own processes. Additionally, it provides an opportunity for others to suggest improvements, which ends up helping everyone.

On-Call

Being On-Call

Who's On-Call?

Alerting Principles

Before an Incident

What is this? A collection of pages detailing how to efficiently deal with any major incidents that might arise, along with information on how to go on-call effectively. It provides lessons learned the hard way, along with training material for getting you up to speed quickly.

ATLASSIAN Products For teams Support Try now Buy now My account

Incident Management Start your journey ITSM More Resources

The path to better incident management starts here

Start your journey

Incident Management at Google

Incident response provides a system for responding to and managing an incident. A framework and set of defined procedures allow a team to respond to an incident effectively and scale up their response. Google's incident response system is based on the [Incident Command System \(ICS\)](#).

Incident Command System

ICS was established in 1968 by firefighters as a way to manage wildfires. This framework provides standardized ways to communicate and fill clearly specified roles during an incident. Based upon the success of the model, companies later adapted ICS to respond to computer and system failures. This chapter explores two such frameworks: [PagerDuty's Incident Response process](#) and [Incident Management At Google \(IMAG\)](#).

Incident response frameworks have three common goals, also known as the "three Cs" (3Cs) of incident management:

- Coordinate response effort.
- Communicate between incident responders, within the organization, and to the outside world.
- Maintain control over the incident response.

When something goes wrong with incident response, the culprit is likely in one of these areas. Mastering the 3Cs is essential for effective incident response.

FEMA EMI

ICS Resource Center Resource Center Contents Important Notices USA.gov DHS

ICS Resource Center

Resource Center Contents

ICS Review Document A summary of key features and principles

NQS Position Task Books and EOC Skillsets Access to NIMS NQS Position Task Books (PTB), Guidelines and the EOC Skillsets and User Guide

NIST
National Institute of Standards and Technology
U.S. Department of Commerce

Special Publication 800-61 Revision 2

Computer Security Incident Handling Guide

e.g. Terminology

Examples of terminology⁴

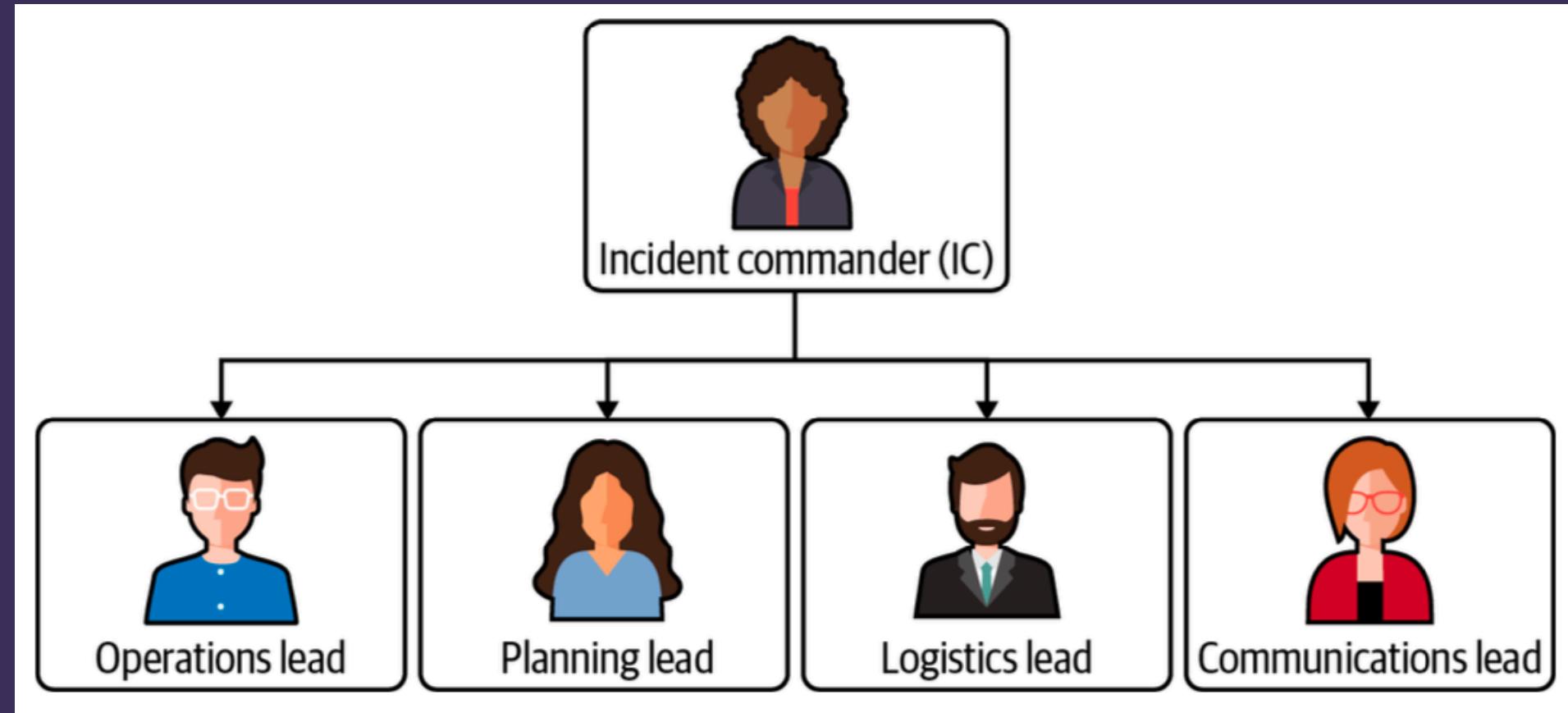
- CAN Reports
- Deputy
- Executive Swoop
- Grenade Thrower
- Incident Commander (IC)
- Resolver
- Severity
- Scribe
- Subject Matter Expert (SME)

⁴ <https://responsepagerduty.com/training/glossary/>

e.g. Roles

Examples of roles at Google

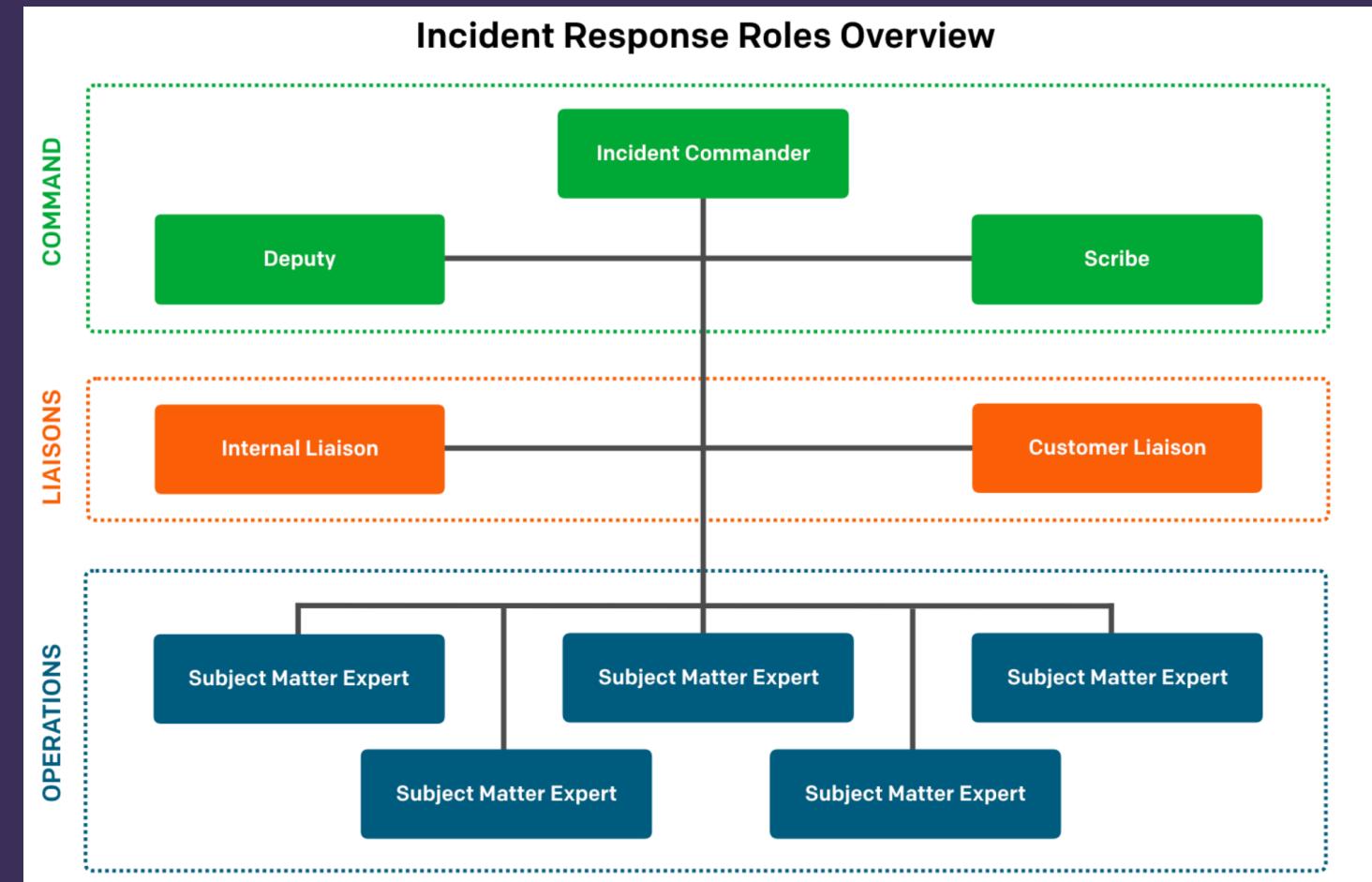
^{5 6}



⁵ Google SRE Workbook, Chapter 9: Incident Response

⁶ Anatomy of an Incident Google's Approach to Incident Management for Production Services, Chapter 4: Mitigation and Recovery, pp. 31-32.

Examples of roles at PagerDuty^{7 8}



⁷ PagerDuty Incident Response Documentation, Different Roles -

⁸ Google SRE Workbook, Chapter 9: Incident Response

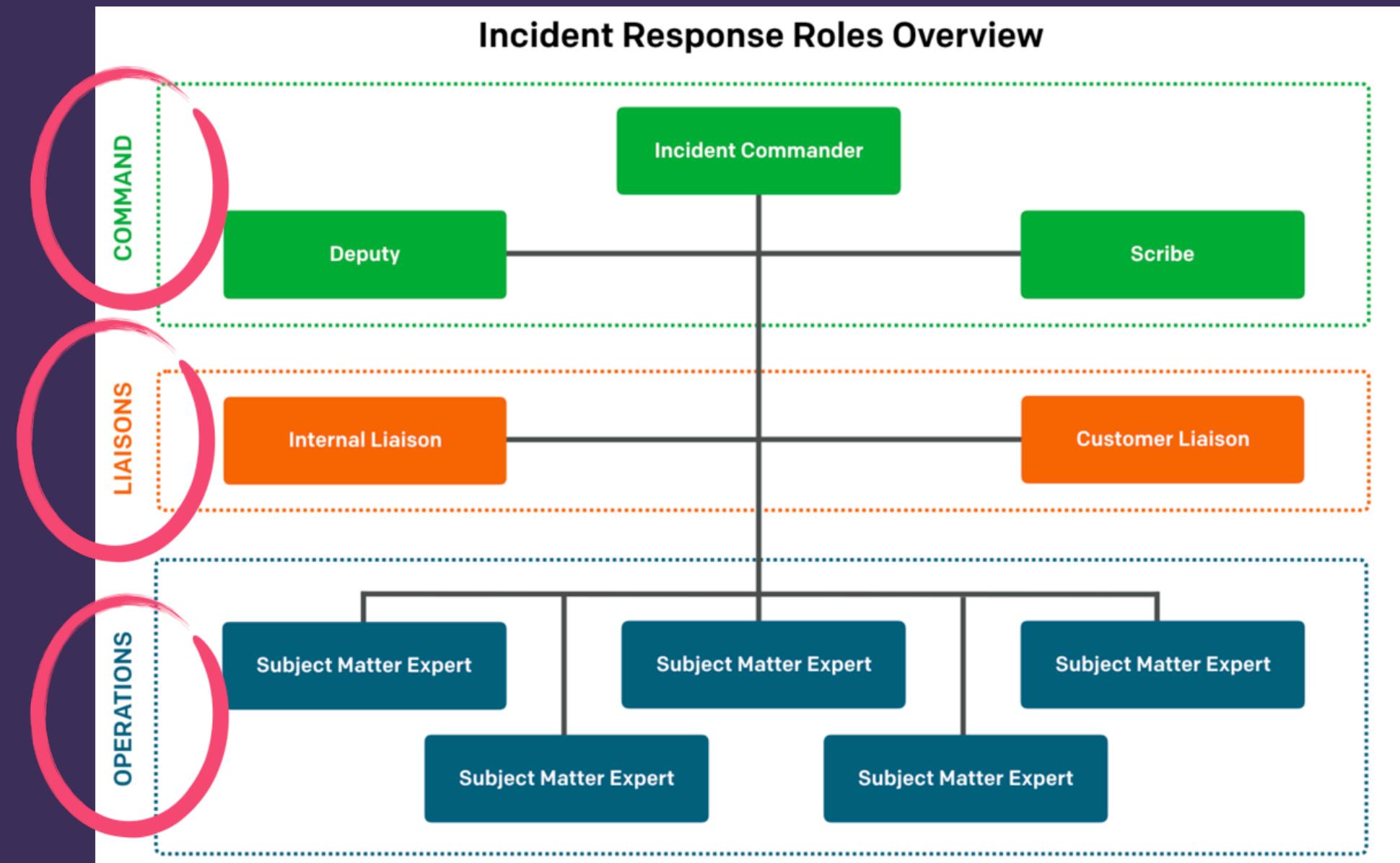
Too much!

Can we translate these practices
into more higher level concepts?

Chapter 3:

3T mental models and life cycles

Examples of roles at PagerDuty



Command role

- Responsibility is managing incident responses to align in organizations.
 - Understand ongoing operations
 - Understand who is doing what
 - Delegate sub-commander responsibility to others if necessary.
- Make incident response **tangible**.

Liason role

- Responsibility is smooth reporting and communications.
 - For both internally and externally.
- Make incident response **transparent**.

Operation role

- Responsibility is actual technical activities to solve issues.
 - Focus on triage, analysis, mitigation and recovery.
 - Communication with rest of organizations is not a primary concern.
- In many cases, operators produce root causes of incidents but don't blame them.
 - Nobody wants to cause incidents.
- All participants focus on assigned roles based on chain of **trust**.

3T mental models for incident response

The incident response roles embody **3T mental models**.

- **Transparency**
 - Keep information of incident responses reachable for everybody.
- **Tangibility**
 - Manage status of incidents.
 - Manage who handles what.
- **Trust**
 - Believe everybody makes best efforts during incidents.
 - Don't blame anybody because nobody wants to cause incidents.

High level view of incident management cycles



High level view of incident management cycles



High level view of incident management cycles



Examples:

- Incident management policy
- Documentation
- Reporting mechanism
- Observability
- Alerting policy
- Incident response training

High level view of incident management cycles



Examples:

- Alerting
- Triage
- Root-cause analysis
- Escalations
- Opening war rooms

High level view of incident management cycles



Examples:

- Rollback deployment (mitigation)
- Kill slow queries (mitigation)
- Fix bug (recovery)
- Add index to tables (recovery)

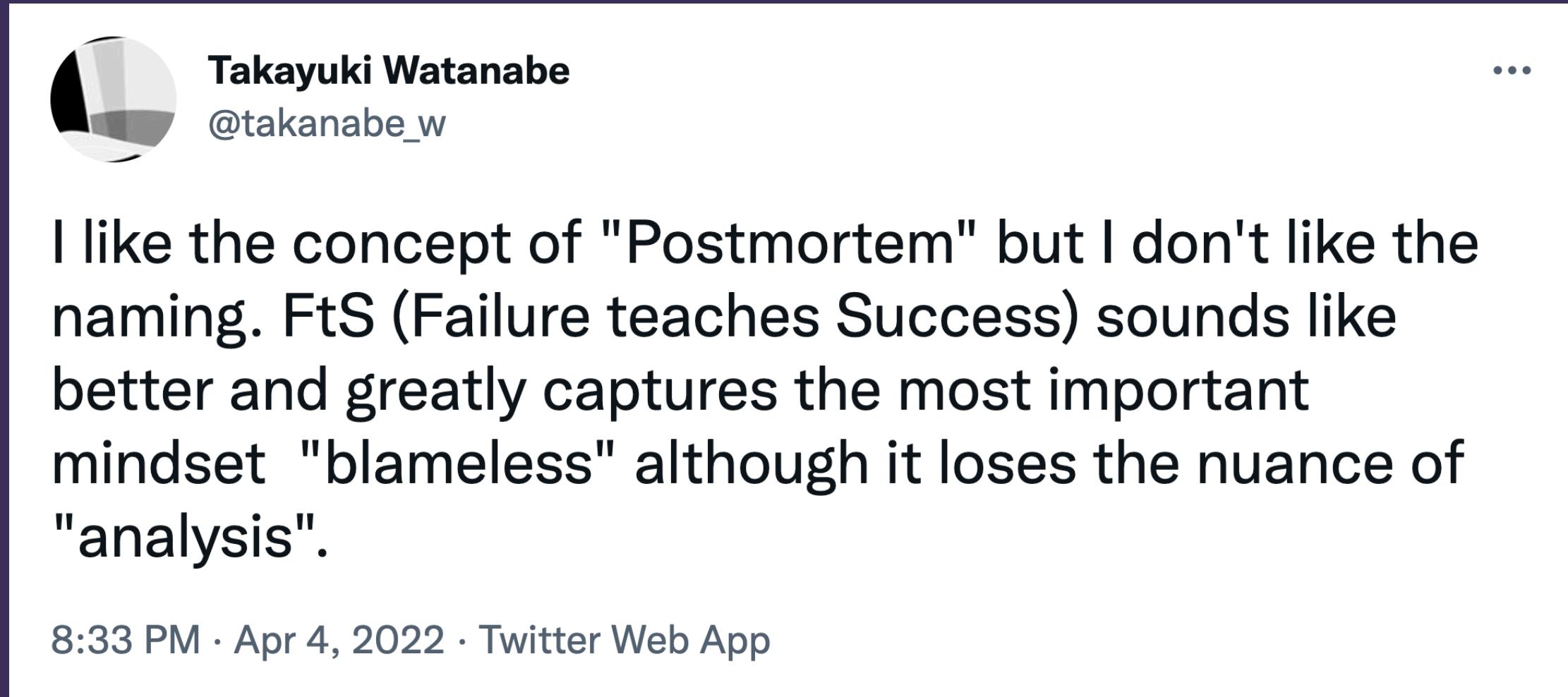
High level view of incident management cycles



Examples:

- Additional triage
- Preparation for postmortems
- Postmortems
- Handle action items raised at postmortems

Postmortem vs FtS



A screenshot of a Twitter post from user @takanabe_w. The post features a profile picture of a sailboat on water. The text reads: "I like the concept of "Postmortem" but I don't like the naming. FtS (Failure teaches Success) sounds like better and greatly captures the most important mindset "blameless" although it loses the nuance of "analysis".". The timestamp at the bottom left is 8:33 PM · Apr 4, 2022 · Twitter Web App.

https://twitter.com/takanabe_w/status/1510943694467186699

Chapter 4:

How can we improve incident management?

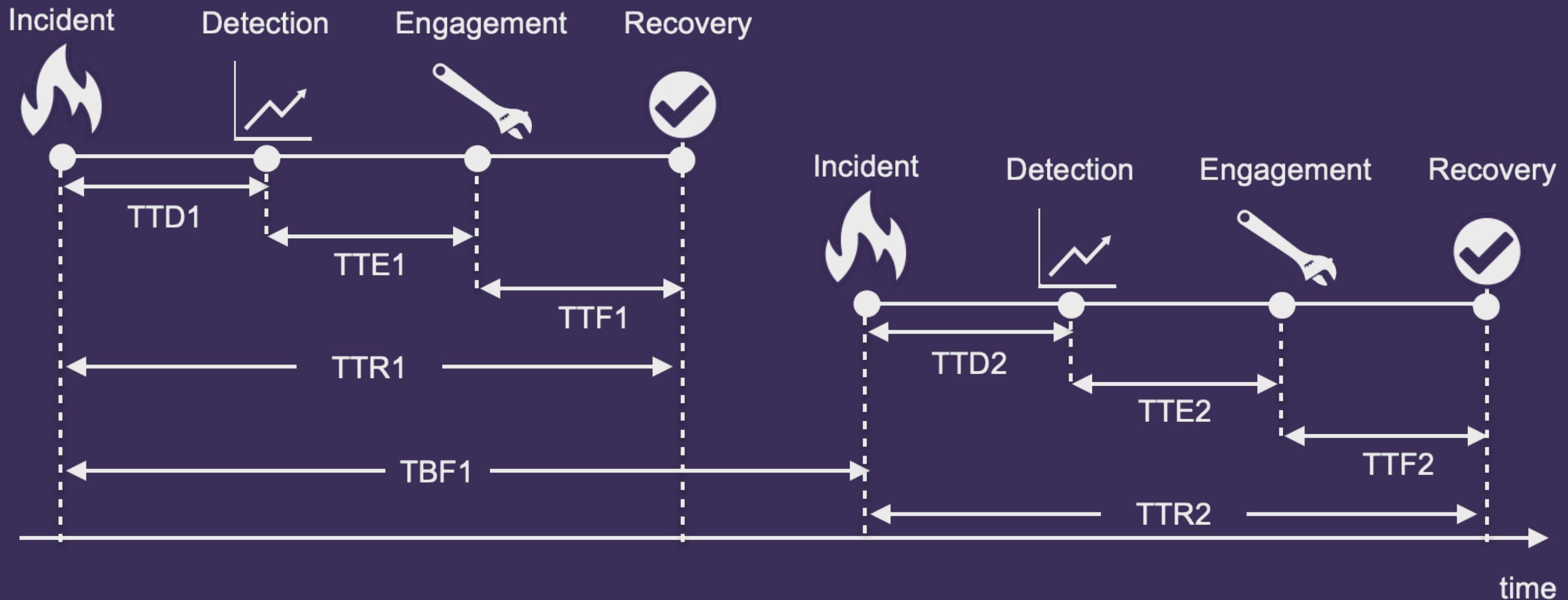
Where should we invest our time?



Where should we invest our time?



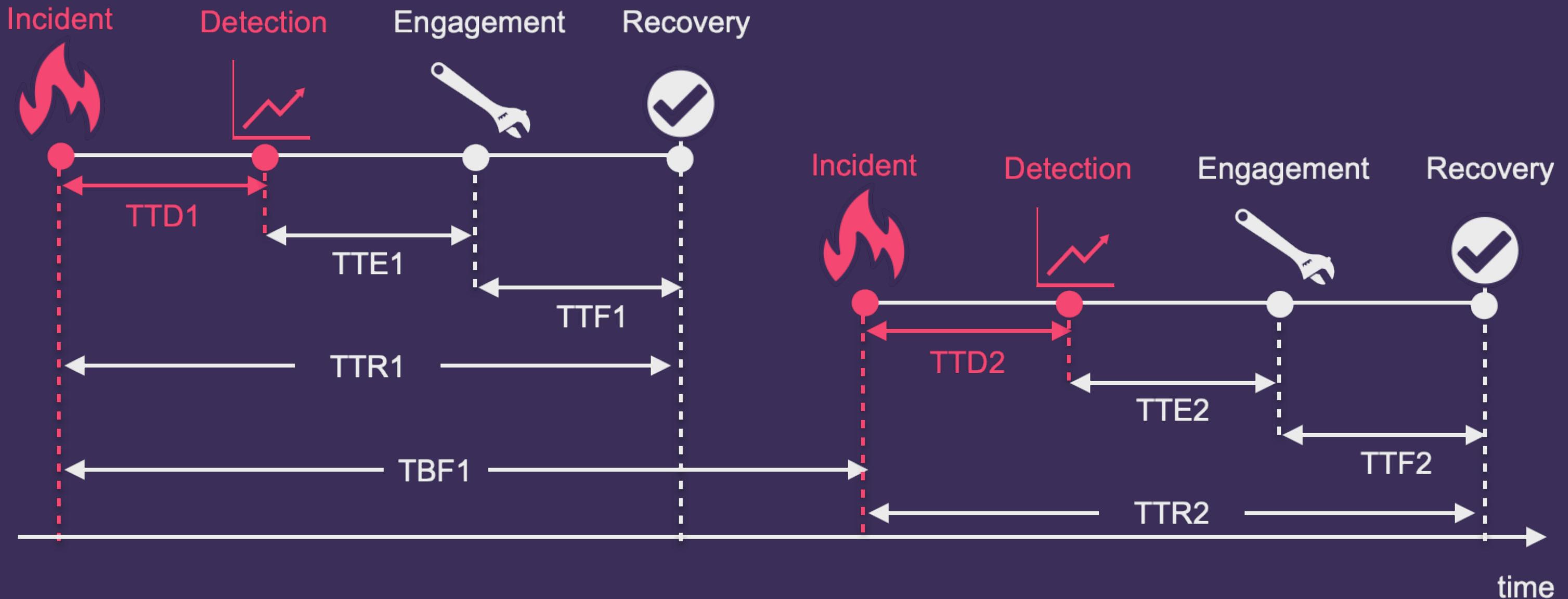
Key times of incident response



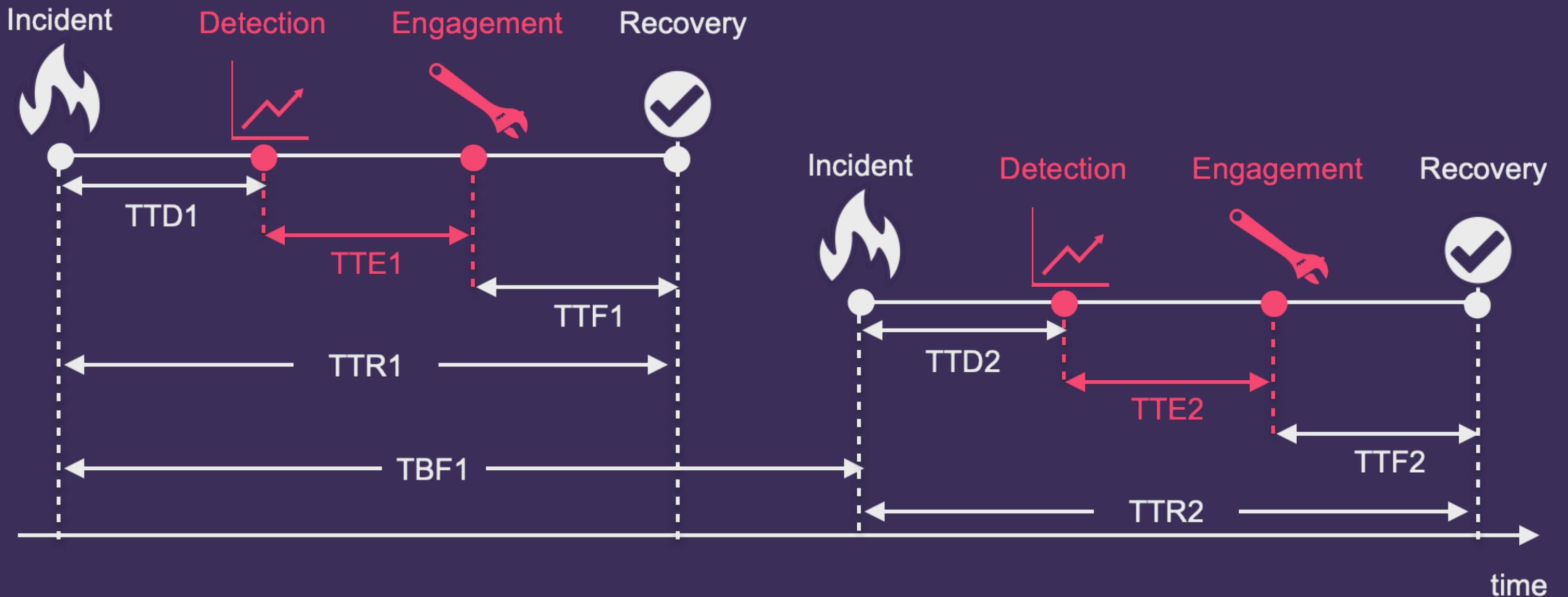
Key times of incident response

- Time **to detect (TTD)**
- Time **to engagement (TTE)**
- Time **to fix (TTF)**
- Time **to repair/recovery (TTR)**
- Time **between failures (TBF)**

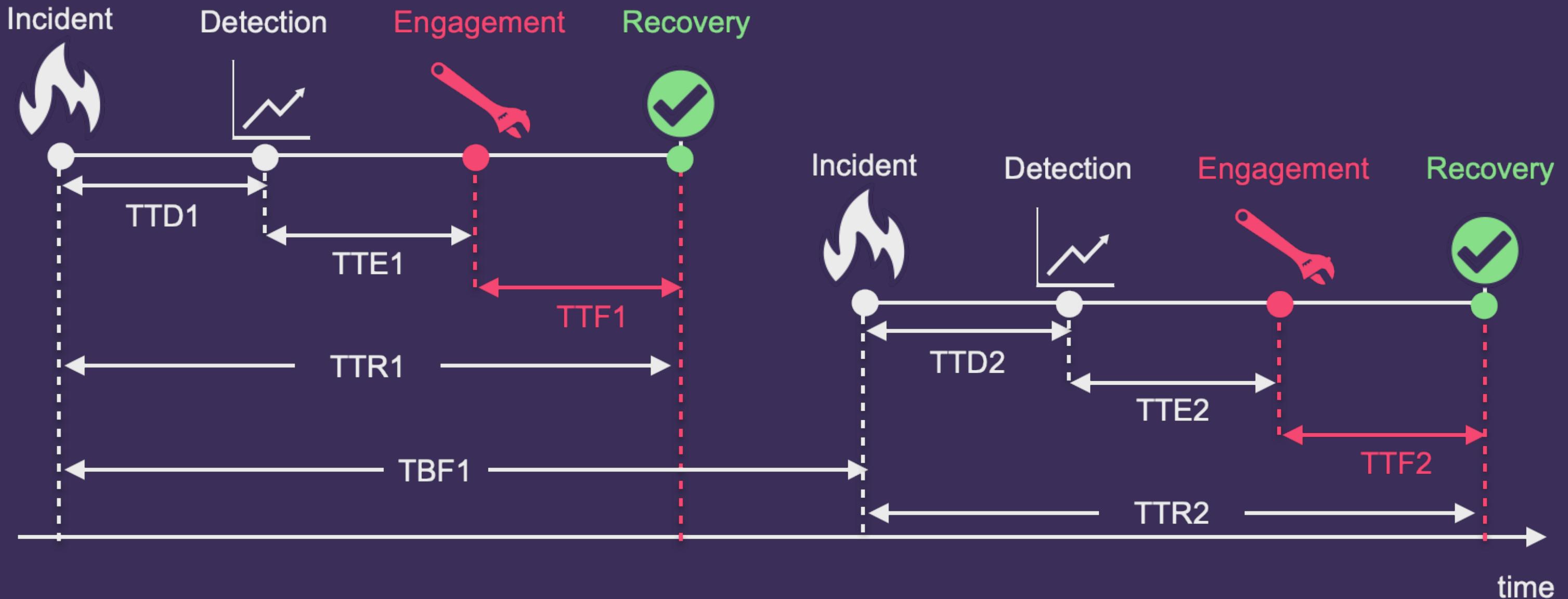
Time to detect (TTD)



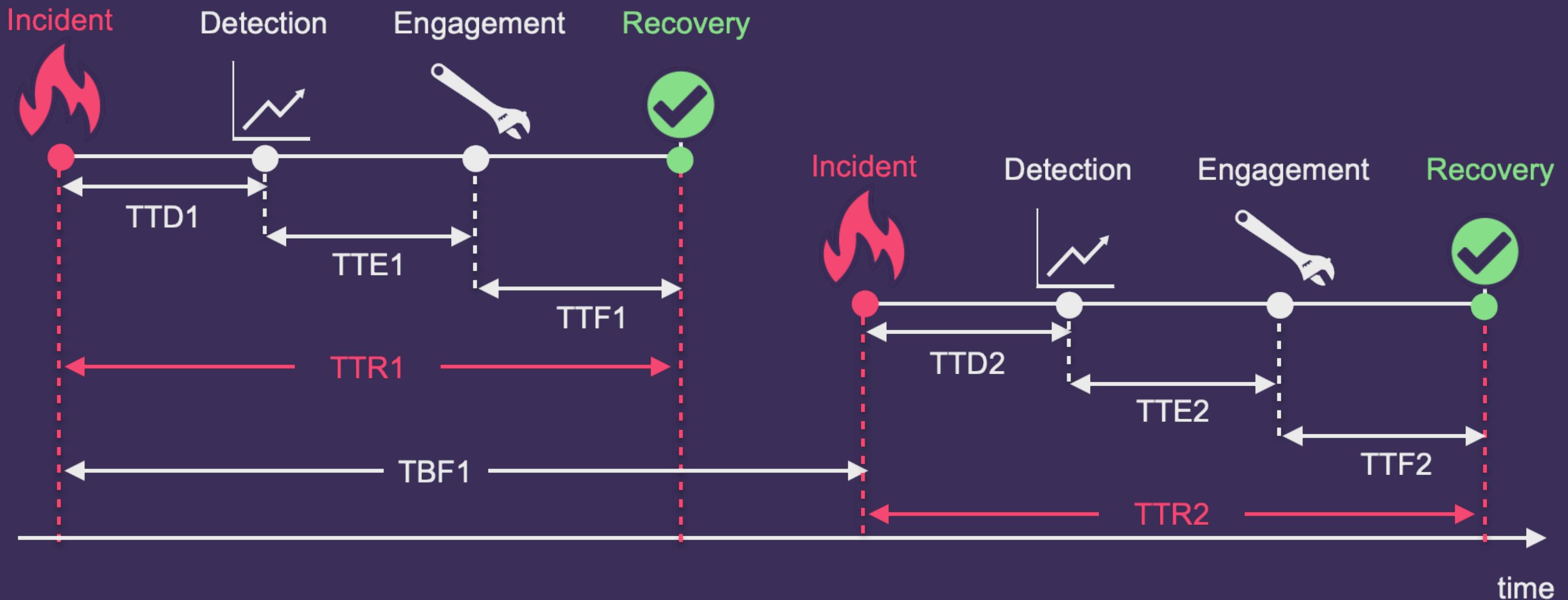
Time to engagement (TTE)



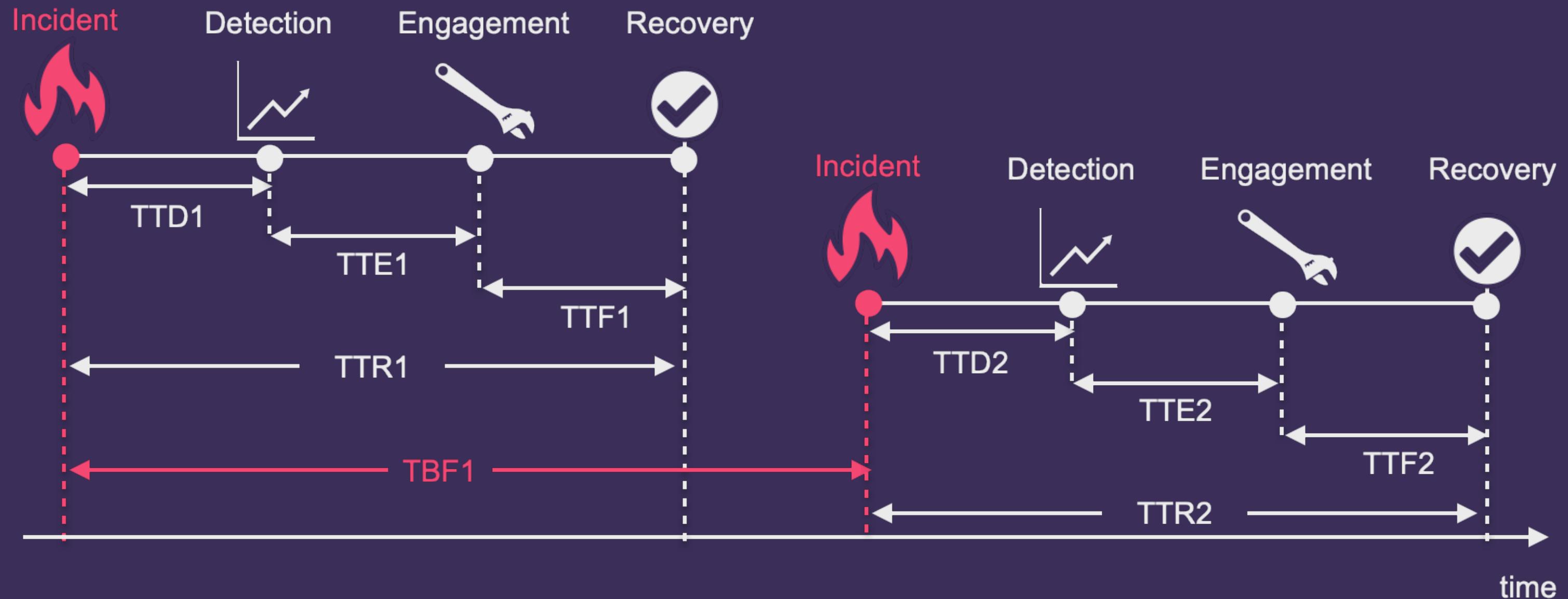
Time to fix (TTF)



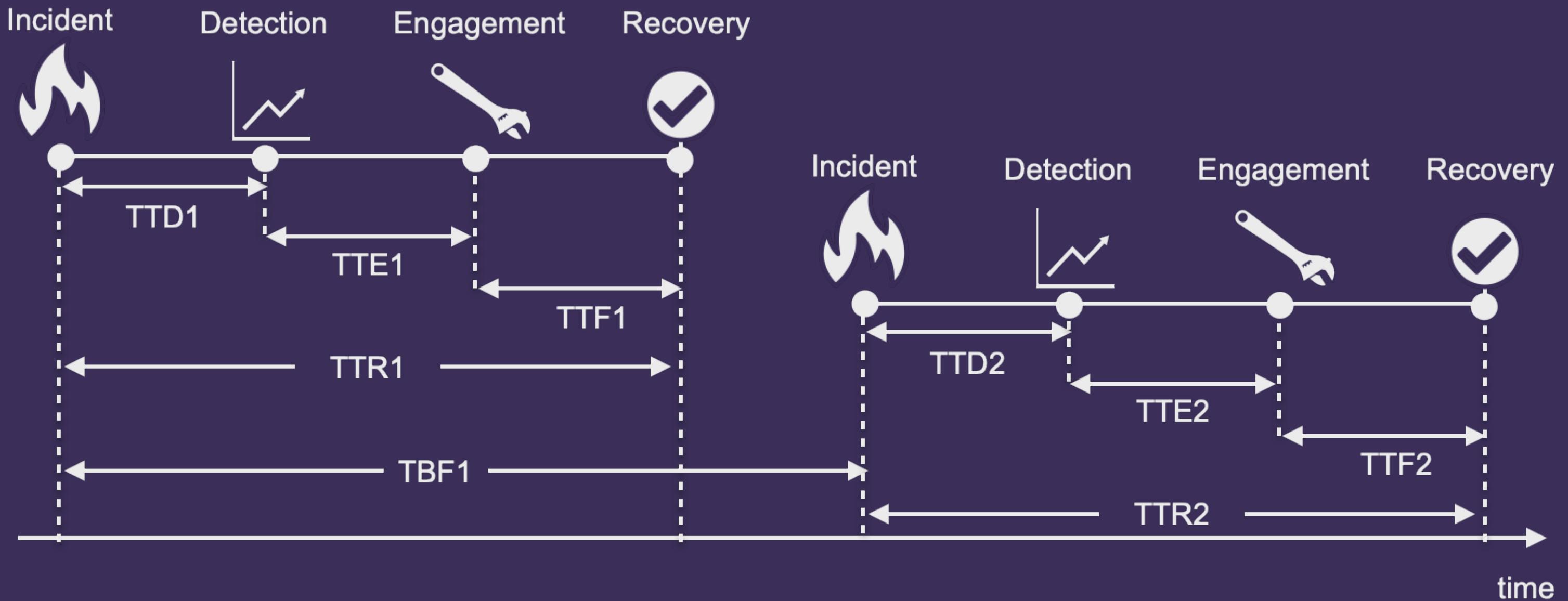
Time to recovery (TTR)



Time between failures (TBF)



Which time do we want to improve?



TTD at Launchable

Current status

- We've already had several detection mechanisms using Datadog and Sentry.

Solution

- Introduction of SLO and Error Budget makes our alerting criteria more clear.
 - But don't forget "Law of diminishing returns" to make decisions.

TTE at Launchable

Current status

- Easy enough to notice during office hours at Slack channels.
- We don't have on-call rotations ATM, which makes TTE uncontrollable.

Solution

- Apply follow-the-sun strategy to cover wide-range hours.
- Introducing on-call rotations and pager.
 - But we don't feel it's necessary now.

TTF at Launchable

Current status

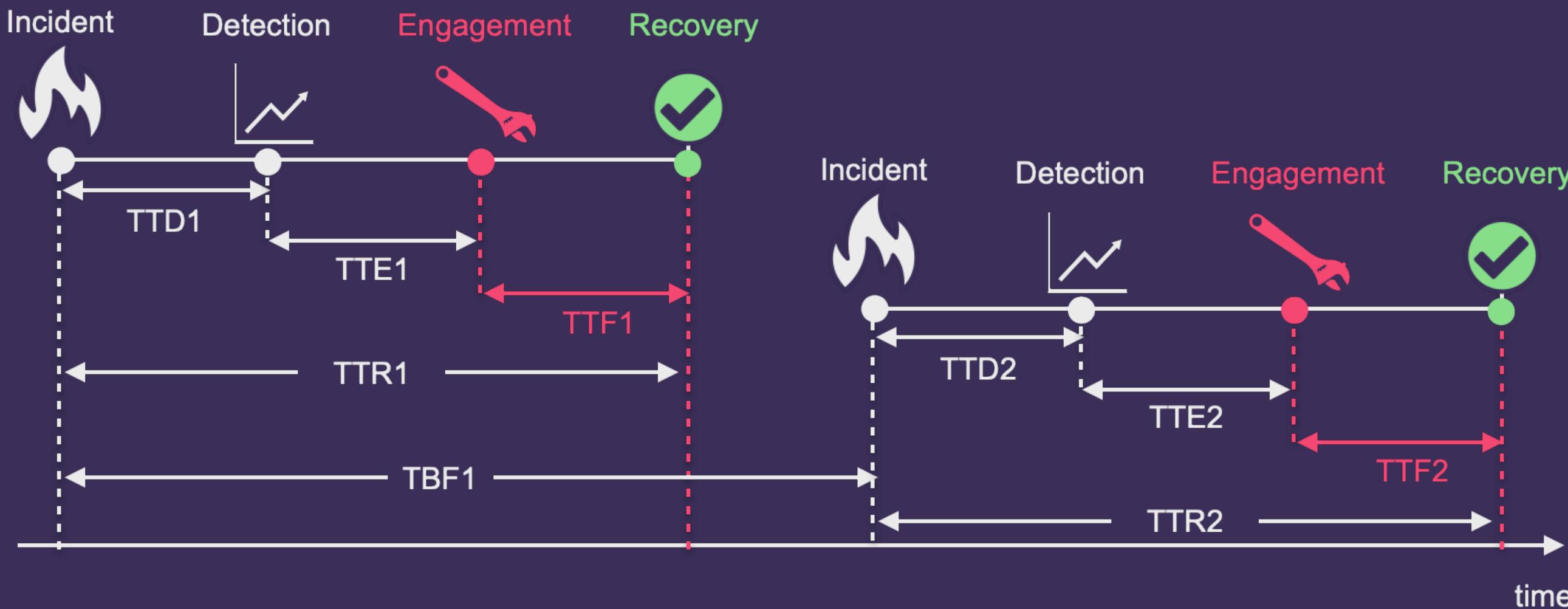
- We don't have enough observability mechanisms
- Depending on each developer's debug skill
- During this window, **developers cannot spend time on product developments.**

Solution

- Introducing more team-shared observability dashboards.
- Introducing more observability mechanism to drill down root causes.

Which time do we want to improve?

TTF improvement brings us high returns with small efforts.

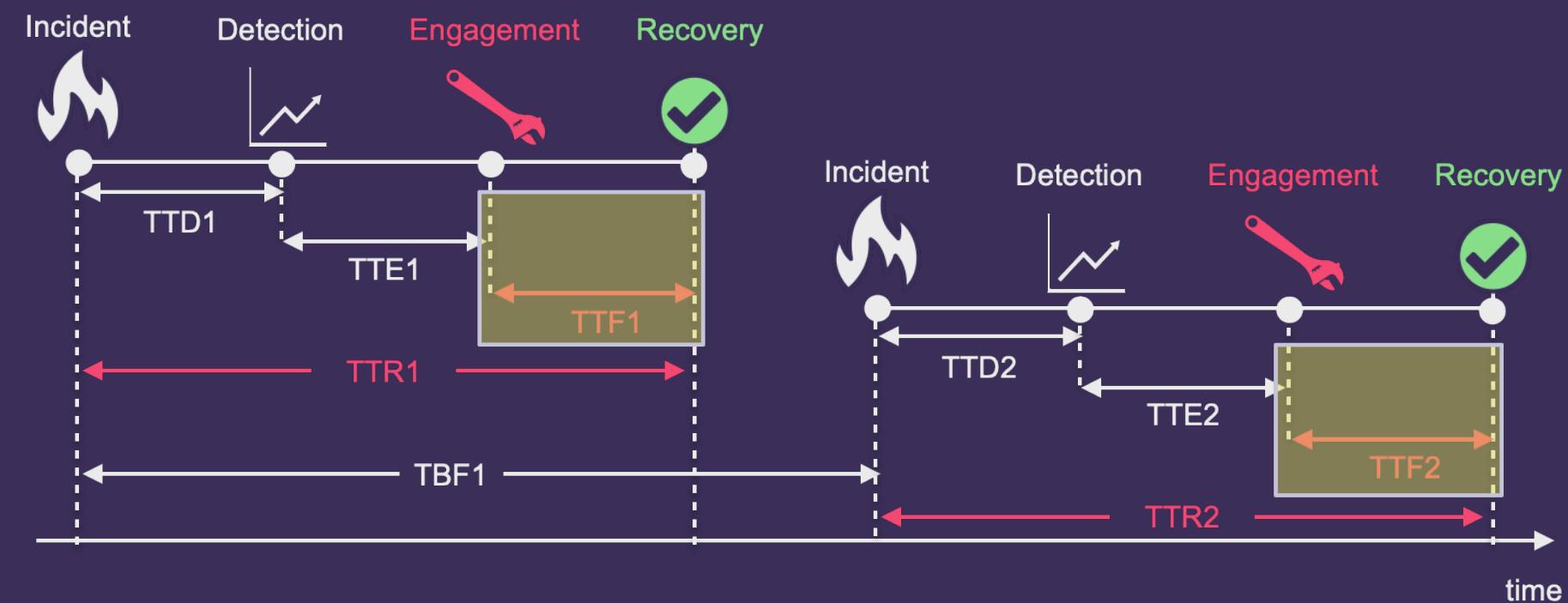


Which do we optimize? MTTR vs MTBF

- Short MTTR and long MTBF are the best
- **Short MTTR but short MTBF**
= Incidents frequently occur but are recovered quickly.
- Long MTTR but long MTBF
= Incidents don't occur frequently but once occur, they aren't recovered soon.

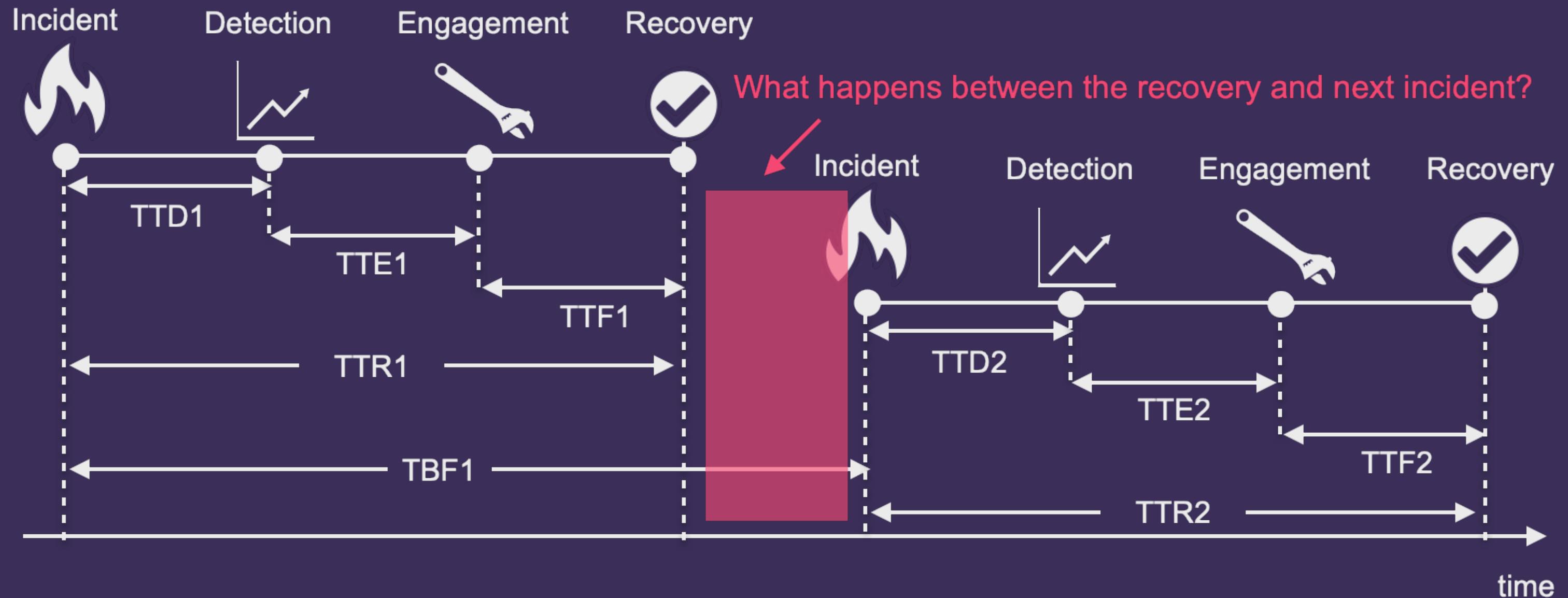
Startups should focus on MTTR improvement

- There is no evolution without high cadence iterations at startups.
- TTD and TTE are difficult to improve for us.
- **Reducing TTF results in reducing MTTR.**



Do we have other times
we haven't articulated?

Hidden key times of incident life cycles



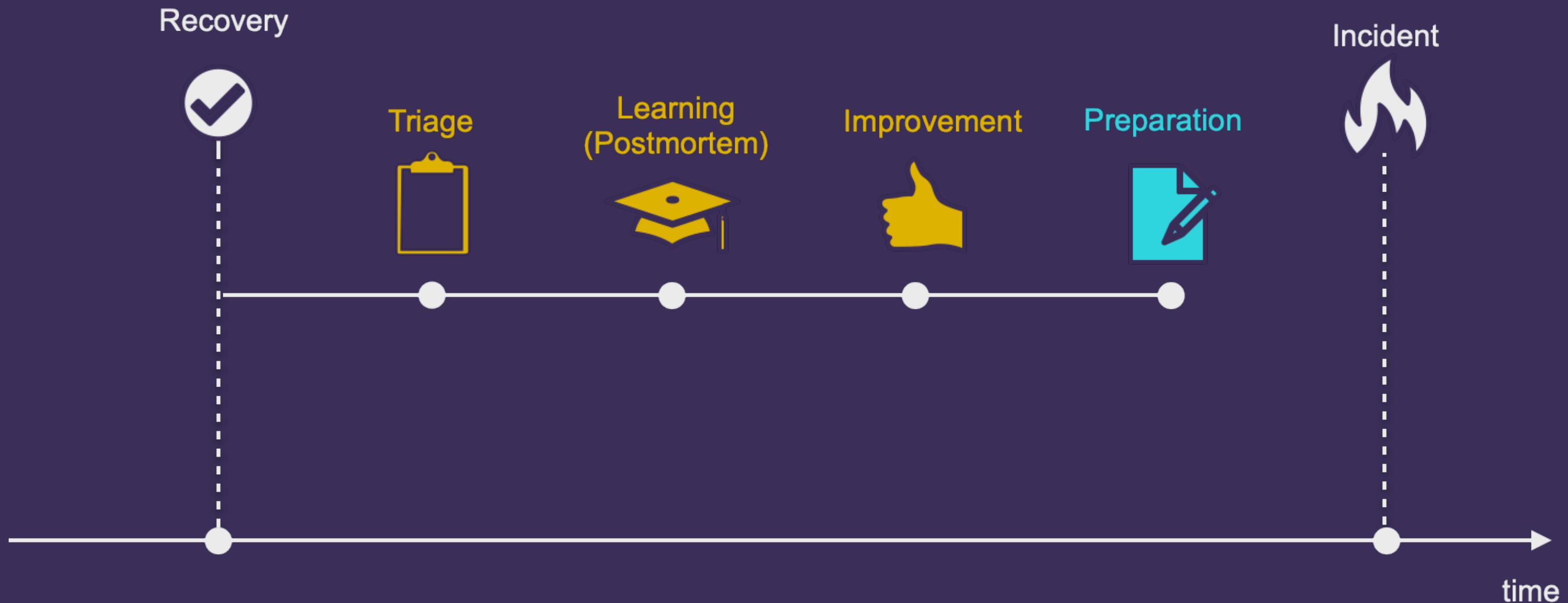
Hidden key times of incident life cycles



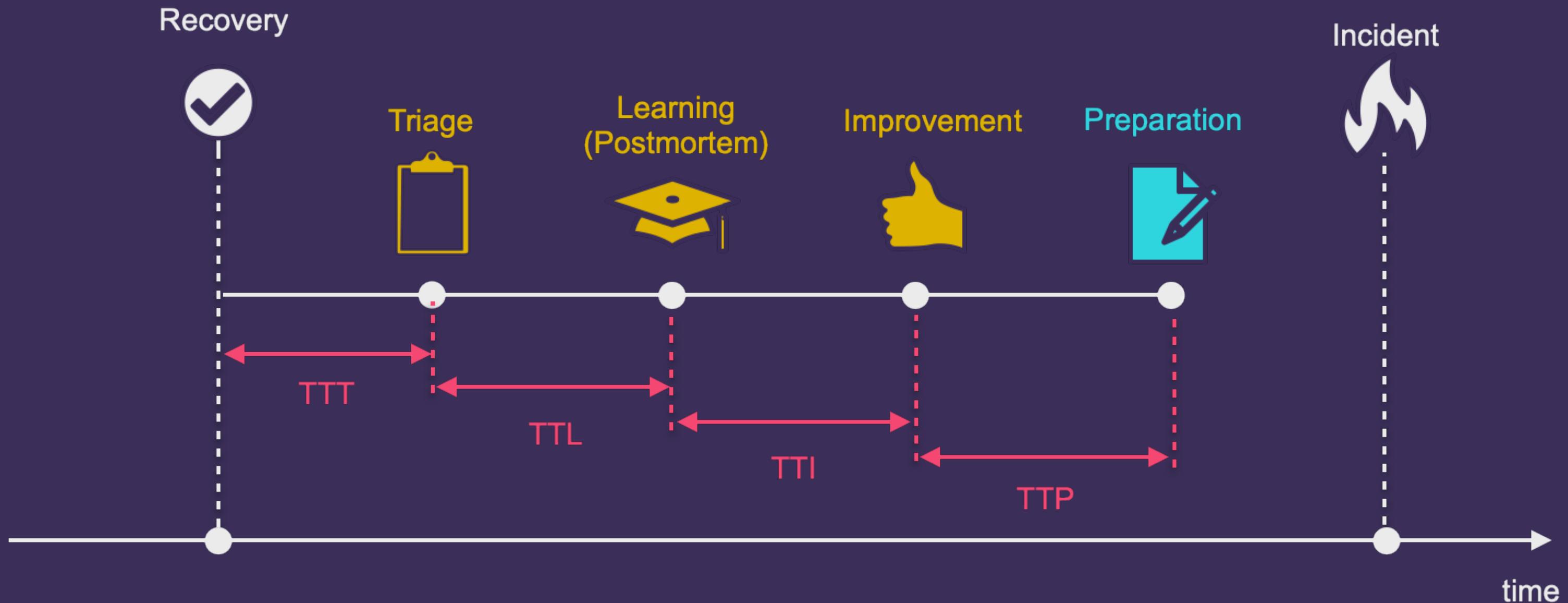
Hidden key times of incident life cycles



Hidden key times of incident life cycles



Hidden key times of incident life cycles



Hidden key times of incident life cycles

Don't underestimate the times we spend as post-incident activities.

- Time to (additional)triage (TTT)
- Time to learn (TTL)
- Time to improvement (TTI)
- Time to preparation (TTP)

Power question:
Which process do you hate?

Which process do you hate?

I personally don't want to spend time on the following processes.

- **Additional triages** to dig root causes.
- **Preparation for learning** (!= I don't like joining postmortem sessions).
- **Maintainance of incident management processes.**

Why additional triages?

- Startups don't have enough observability mechanisms.
 - We sometimes cannot find root causes (this is acceptable).
 - We tend to spend a lot of time here in that situation.

Why preparation for learning?

- Preparation for team-wise learning sessions take time.
 - Documenting for Postmortems.
 - Copy & paste dances to create timeline scattered various places.
 - Timeline needs to consider time-zones.
- There is a gravity which prevent people from announcing incident casually.
 - For startups, the most important activities are learning as a team.
 - If TTL is long, people cannot announce incidents casually.
 - As a result, postmortems ruin short MTTR with high cadence learning iterations.

Why maintenance of incident management processes?

- **Maintenance of incident management processes** contains:
 - Updating incident management policy.
 - Improving incident management structures.
 - Updating documents.
 - Training people to align with the updates.
- Characteristically, incidents don't occur frequently,
 - Too tough to memorize incident response processes for everybody.
 - In urgent situation, people don't read documents.

Can we reduce TTI?

- It's depending on action items coming from postmortems.
- No teams can handle all action items we discussed during postmortems.
- Common anti-pattern: people create too many action items and assign without priorities.

Approach for unbalanced action items

- Think of engineering members' capacity
- Prioritize and classify the work⁹ ¹⁰

⁹ Postmortem Action Items: Plan the Work and Work the Plan, USENIX SRECon 2017

¹⁰ Anatomy of an incident management, Chapter 5

Importance / Size / Urgency (ISU) Matrix

- Assignee's confidence is also valuable to declare.



ISU Matrix on GitHub Projects

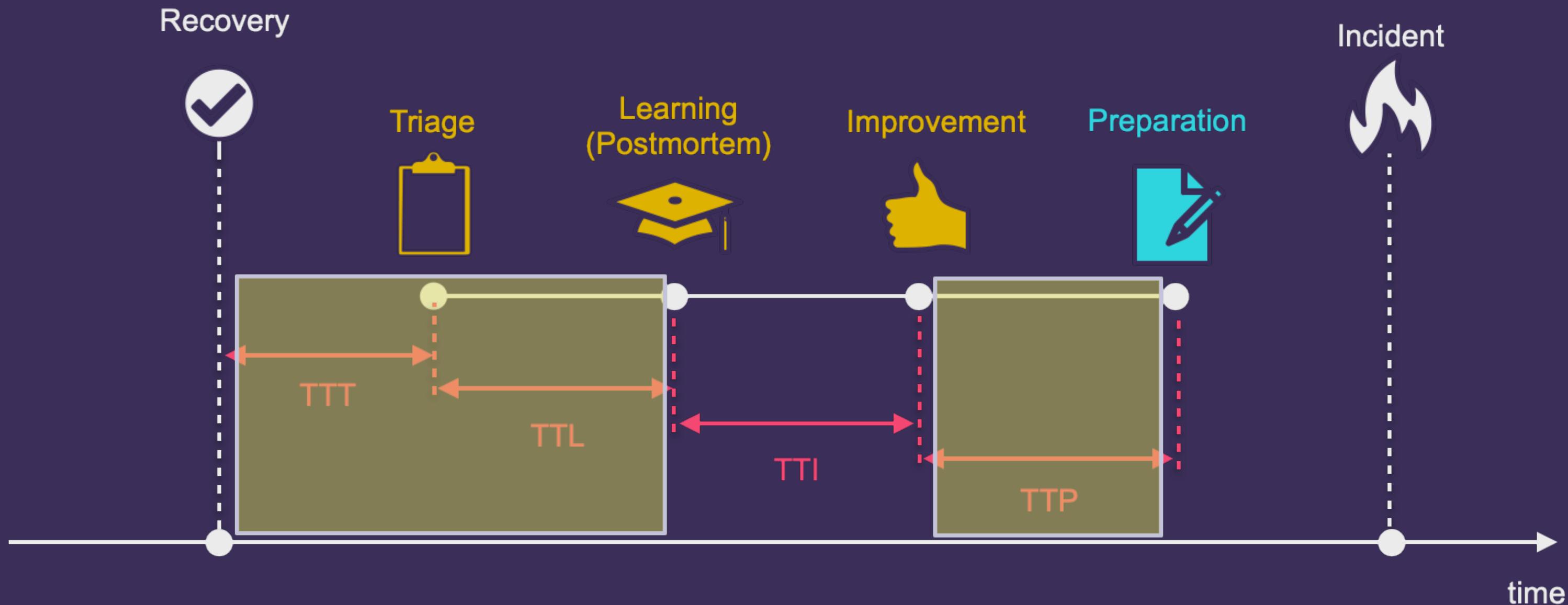
The screenshot shows a GitHub Projects board with a Priority & Size view. The board is organized into three main sections representing different iterations:

- Mar 23 - Mar 30**: Contains 4 items (1-4). Item 1 is highlighted with a blue border. All items are marked as Done.
- Mar 31 - Apr 13 (Current)**: Contains 3 items (5-7). Item 5 is Done (Low priority, Huge impact). Item 6 is In Progress (Low priority, Medium impact). Item 7 is an Action item (Medium priority, Huge impact).
- No Iteration**: Contains 3 items (8-10). Item 8 is Backlog (Medium priority, Medium impact). Item 9 is Backlog (Low priority, Small impact). Item 10 is Triage (Want) (Medium priority, Medium impact).

The columns represent Status, Urgency, Impact, and Expected time. A Beta button is visible in the top right corner.

| Title | Status | Urgency | Impact | Expected time |
|------------|---------------|---------|--------|---------------|
| 1 | Done | Low | Medium | a few days |
| 2 | Done | Low | Small | a day |
| 3 | Done | Medium | Medium | a week |
| 4 | Done | High | Huge | a day |
| + Add item | | | | |
| 5 | Done | Low | Huge | a few hours |
| 6 | In Progress | Low | Medium | a day |
| 7 | Action items | Medium | Huge | a day |
| + Add item | | | | |
| 8 | Backlog | Medium | Medium | a week |
| 9 | Backlog | Low | Small | a day |
| 10 | Triage (Want) | Medium | Medium | a day |

My focus is reduction of TTT, TTL, and TTP



Chapter 5:

Choosing right strategies and tools

Phases and the number of software engineers

Note: the numbers are estimated by the presenter based on previous experiences.

- Phase 0: Founding ~ 4 software engineers
- Phase 1: 5 ~ 10 software engineers
- Phase 2: 11 ~ software engineers



Let's reframe the original question again

- Reframe "Does a startup need incident management?"
- At startups, **how can we:**
 - **Build an incident management structure enforcing the 3T mental models?**
 - **Improve the "times" of the incident management life cycle?**

Evolution of incident management at Launchable

| Improvement target | Actions from phase 0 to 1 | Actions from phase 1 to 2 |
|---------------------------|--|--|
| Transparency | - Encourage push communication | - Encourage pull communication - Create war rooms - Share status pages |
| Tangibility | - Automate parts of incident response flow | - Automate entire incident response flow - Introduce incident lead role |
| Trust | - Introduce blameless culture | - Split lead and operation roles for complex incidents |
| Time to Engagement (TTE) | - Automate incident announcements | - Automate entire incident response flow - Introduce on-call rotations - Expand follow-the-sun coverages |
| Time to Fix (TTF) | - Introduce observability | - Improve observability |
| Time to Triage (TTT) | - Introduce observability | - Improve observability |
| Time to Learn (TTL) | - Introduce postmortem template | - Generate postmortem |
| Time to Preparation (TTP) | - Create incident management policies | - Enforce incident management policies - Self-service incident response trainings |

Phase 0: Founding ~ 4 software engineers

No strategy

- Product does not have customers.
- We don't need incident responses.
- Build incident management structure based on product growth.
- All members do everything if necessary.

Incident management system



All employees

Phase 1: 5 ~ 10 software engineers

Environmental changes from phase 0 to 1

- When products have customers, we need an incident management.
- The more software engineers join, the more incidents happen.

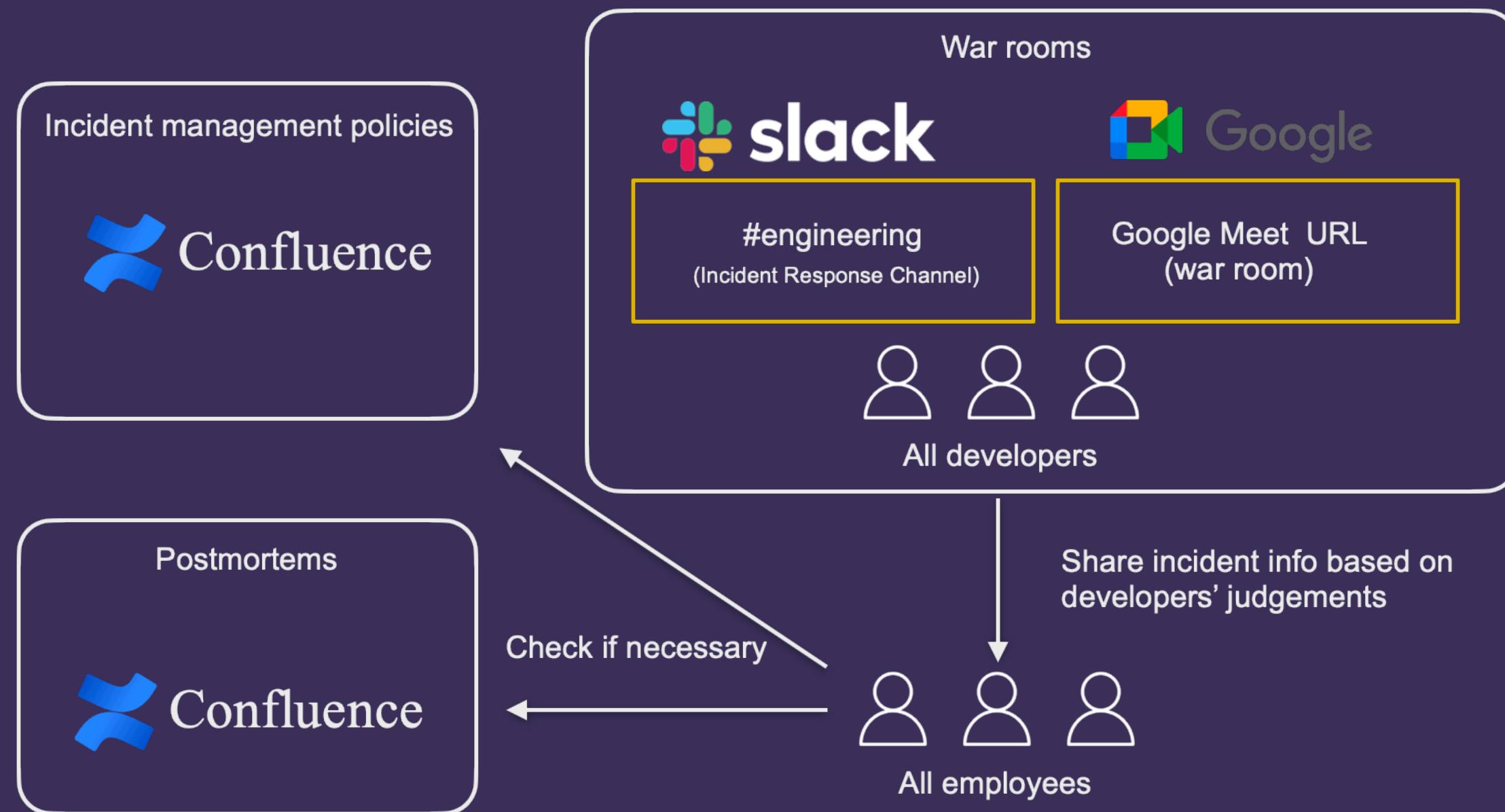
Strategy

Make everything **simple** and **easy** to follow

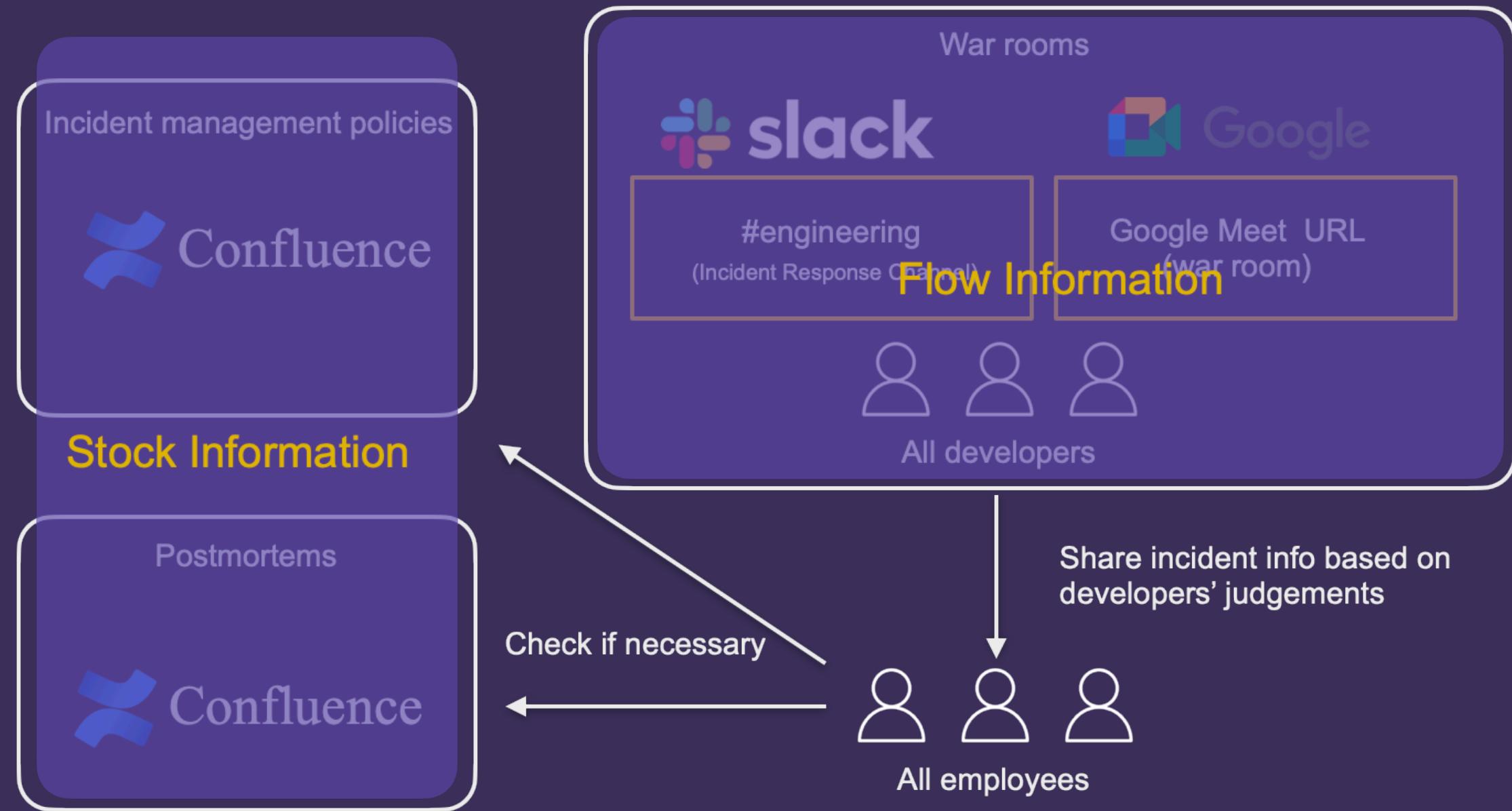
Incident management changes from phase 0 to 1

| Improvement target | Actions from phase 0 to 1 | Actions from phase 1 to 2 |
|---------------------------|--|--|
| Transparency | - Encourage push communication | - Encourage pull communication - Create war rooms - Share status pages |
| Tangibility | - Automate parts of incident response flow | - Automate entire incident response flow - Introduce incident lead role |
| Trust | - Introduce blameless culture | - Split lead and operation roles for complex incidents |
| Time to Engagement (TTE) | - Automate incident announcements | - Automate entire incident response flow - Introduce on-call rotations - Expand follow-the-sun coverages |
| Time to Fix (TTF) | - Introduce observability | - Improve observability |
| Time to Triage (TTT) | - Introduce observability | - Improve observability |
| Time to Learn (TTL) | - Introduce postmortem template | - Generate postmortem |
| Time to Preparation (TTP) | - Create incident management policies | - Enforce incident management policies - Self-service incident response trainings |

Incident management system (phase 0 to 1)



Incident management system (phase 0 to 1)



Foundation of incident management policies

- We maintain policies on Confluence.

The screenshot shows a Confluence page with the title "Incident management". Below the title is a small profile picture placeholder and some metadata: "2 min read" and "10 people viewed". A yellow callout box contains a warning message: "⚠ You may be opening this page in such a rush. Take your time. Take your time and relax. Good things don't happen if you are in a hurry to deal with them. Everyone will help you. Let's calm down first." The main content area below the callout has a heading "What if an incident happens in the production?" followed by a list of instructions:

Here's a short list of things to do if you find a problem

1. Let's start with a [Slack#engineering](#) report. "There's a fault in the production! I'll create a ticket now."
 - a. No need to explain in detail. First of all, your report that the failure is happening and that you are aware of it.
2. Post a link to your ticket on Slack.
 - a. We will write the actual response status on this ticket.
3. Write [Postmortems](#) once the incident is resolved.

Foundation of incident management policies

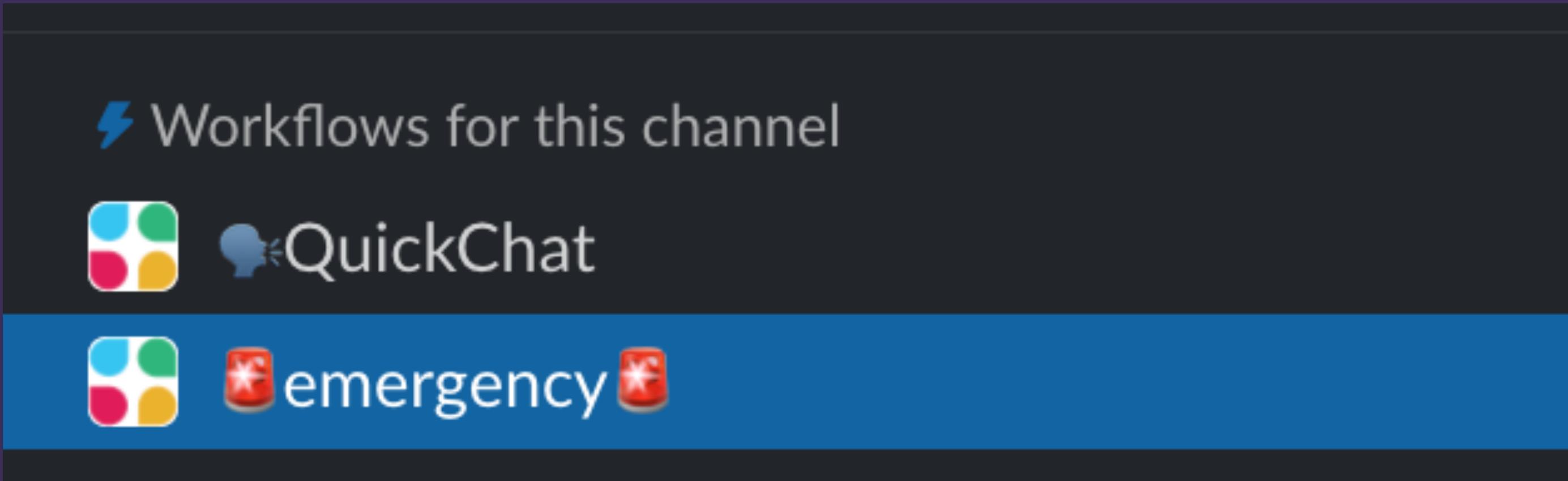
- We maintain policies on Confluence.

The screenshot shows a Confluence page with the following details:

- Page Path:** Product / ... / Incident management
- Page Title:** Incident level
- Page Statistics:** 2 min read • 11 people viewed
- Information Box:** An info icon with the text: "This page only defines the incident level. The actions to be taken when an incident actually occurs are described in [Incident management](#)".
- Section:** Customers' builds cannot stop ([Engineering Manifesto](#))
More and more customers are implementing our services into their build processes. We have to take responsibility for the quality of our services, including their availability. For this reason, we need to take measurements. First of all, I am going to measure how much failure is occurring and how often. Define the level of failure and record the number of times each level of failure has occurred.
- Section:** Incident severity level
Sev3 will process during regular business hours. Otherwise, please respond to emergencies. Please use the emergence button of the #engineering channel on Slack.
- Section:** Sev3: Minimum impact on customers

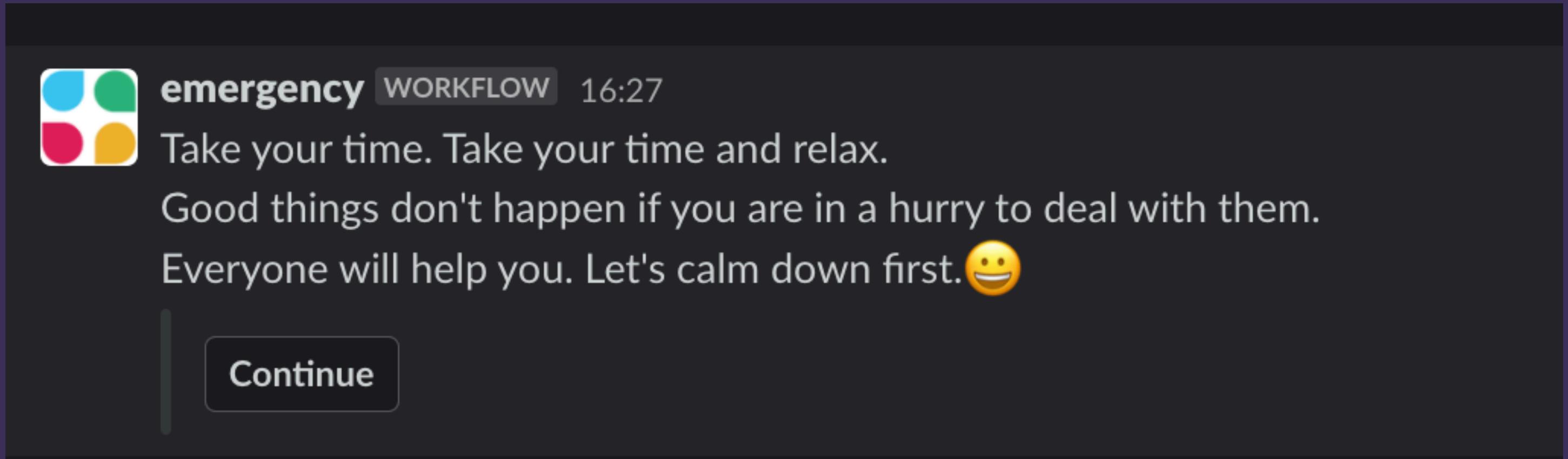
Automation of incident escalations

- We escalate incidents using Slack Workflow.
- We handle incidents in Slack channel and Google Meet.



Automation of incident escalations

- We escalate incidents using Slack Workflow.
- We handle incidents in Slack channel and Google Meet.



Automation of incident escalations

- We escalate incidents using Slack Workflow.
- We handle incidents in Slack channel and Google Meet.

emergency WORKFLOW 16:27
@channel Incident has occurred!

[Incident management](#)

meet.google.com/ [REDACTED]

id.atlassian.com
[Log in with Atlassian account](#)
Log in to Jira, Confluence, and all other Atlassian Cloud products here. Not an Atlassian user? Sign up for free.

Introduction of postmortem

- We keep all postmortems on Confluence.
- We create a new postmortem page using a Confluence template feature.

The screenshot shows a Confluence page titled "Postmortems". At the top, there is a "Create postmortem" button. Below it is a table with columns: Title, Status, Incident Date, and Completion Date. The table lists four incidents:

| Title | Status | Incident Date | Completion Date |
|--|--|---------------|--------------------|
| 2022-03-29 intermittent 401 errors in the production web app | IN PROGRESS | 29 Mar 2022 | 3 comments, 1 like |
| 2022-03-29 Demo workspace didn't show the test run detail page | IN PROGRESS | 28 Mar 2022 | |
| | Mitigated / the root cause is likely found | 16 Feb 2022 | |
| | FINISHED | 16 Feb 2022 | |
| | Mitigated | 16 Feb 2022 | |

Introduction of postmortem

- We keep all postmortems on Confluence.
- We create a new postmortem page using a Confluence template feature.

The screenshot shows a Confluence page titled "2022-04-08 #{Add simple Title}". At the top left, there's a "Page Properties" sidebar with fields for Status (set to IN PROGRESS), Incident Date, Completion Date, and Incident Level (with a link to "See Incident Level Definition"). Below the sidebar, the main content area contains sections for "What is a postmortem for?", "Rules", and "Participants".

What is a postmortem for?

- Settle down and share what happened during the incident with other team members
- Collect incident cases and make them available for later reference
- Find root causes and explore measures to prevent a recurrence

Rules

- Do not blame anyone
- Consider how we can systematically prevent a recurrence
- Consider as many solutions as we can, regardless of whether we actually do or don't do. Then we choose which we tackle.

Participants

Very simple and easy to follow

Postmortems as strong fact-based data

- Even we cannot solve root causes, you can use the postmortems as data.

Separate [REDACTED] from [REDACTED] for maintainability and extensibility

Created by [REDACTED]
Last updated: Mar 24, 2022 • 4 min read • 6 people viewed

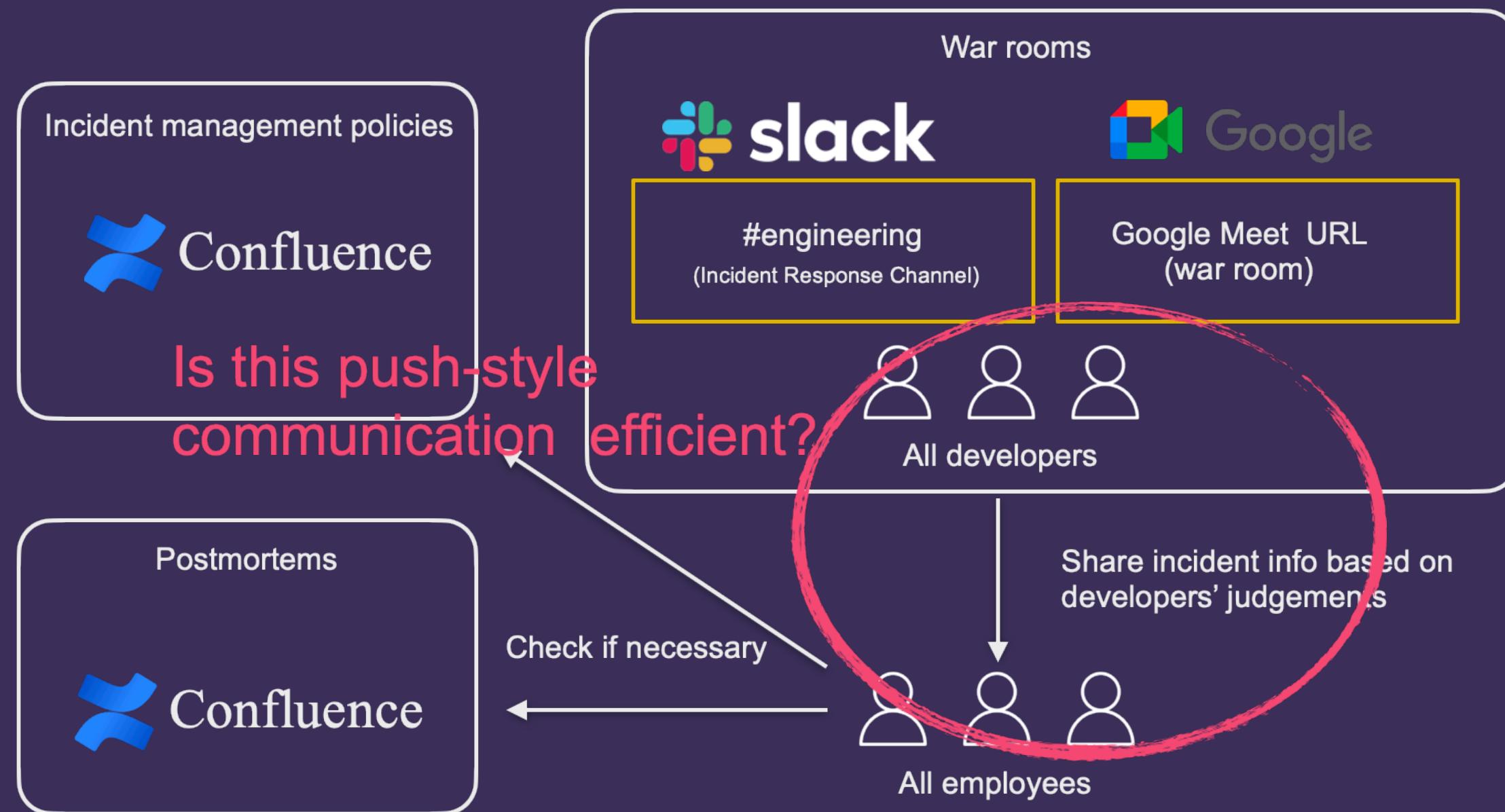
| | |
|-------------------|--------------|
| Status | PROPOSED |
| Driver | @[REDACTED] |
| Date | Mar 22, 2022 |
| Team Central | [REDACTED] |
| Expected audience | engineers |

Why

- We have high ratio of [REDACTED]-related incident compared to API server portion of [REDACTED]. Out of 19 incidents occurred in 2021 and 2022, 14 of them are caused by [REDACTED], which we are experiencing one [REDACTED]-related incident per month. Doubtless to say high availability of the [REDACTED] system would improve service level as well as engineer resource focused to contribute on important product.

Can we improve the incident management?

e.g. Does this what human should take care?



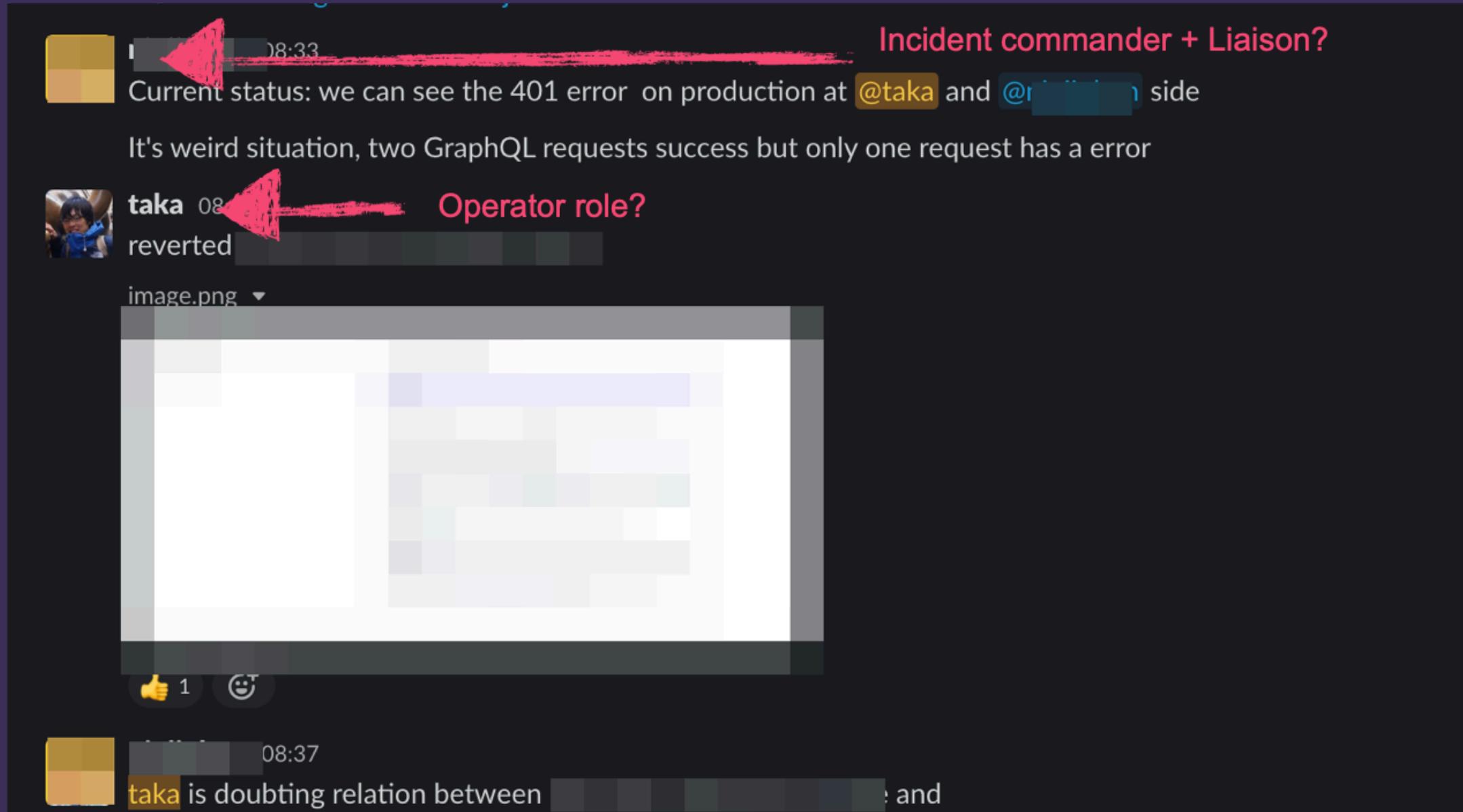
e.g. Does this what human should take care?

- We don't have solid policy but policy does not scale.
- Employees are living in Japan and US.
- Sharing all information on Slack is easy to miss.

Problems

- [REDACTED]
- [REDACTED]
- [REDACTED] and [REDACTED] knew the error, but we didn't forward it to US members

e.g. Do we need roles?



Phase 2: 11 ~ ?? software engineers

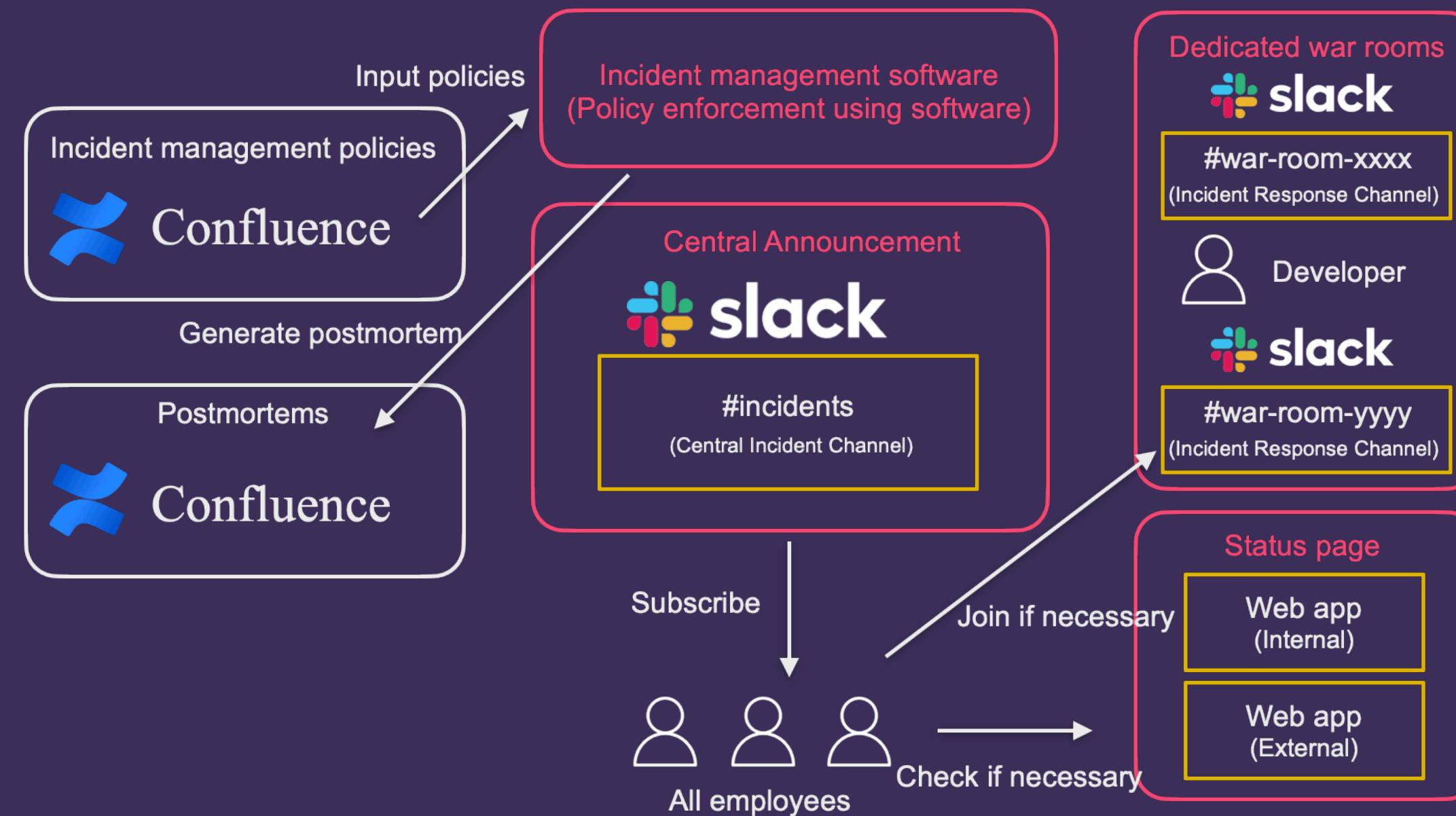
Environmental changes from Phase 1 to 2

- Our products have more customers.
- The more software engineers join, the more incidents happen.
- Increase of employees and time-zone gaps make sync and push-style communications tough.
 - In the first place, Launchable encourages async and written communications.

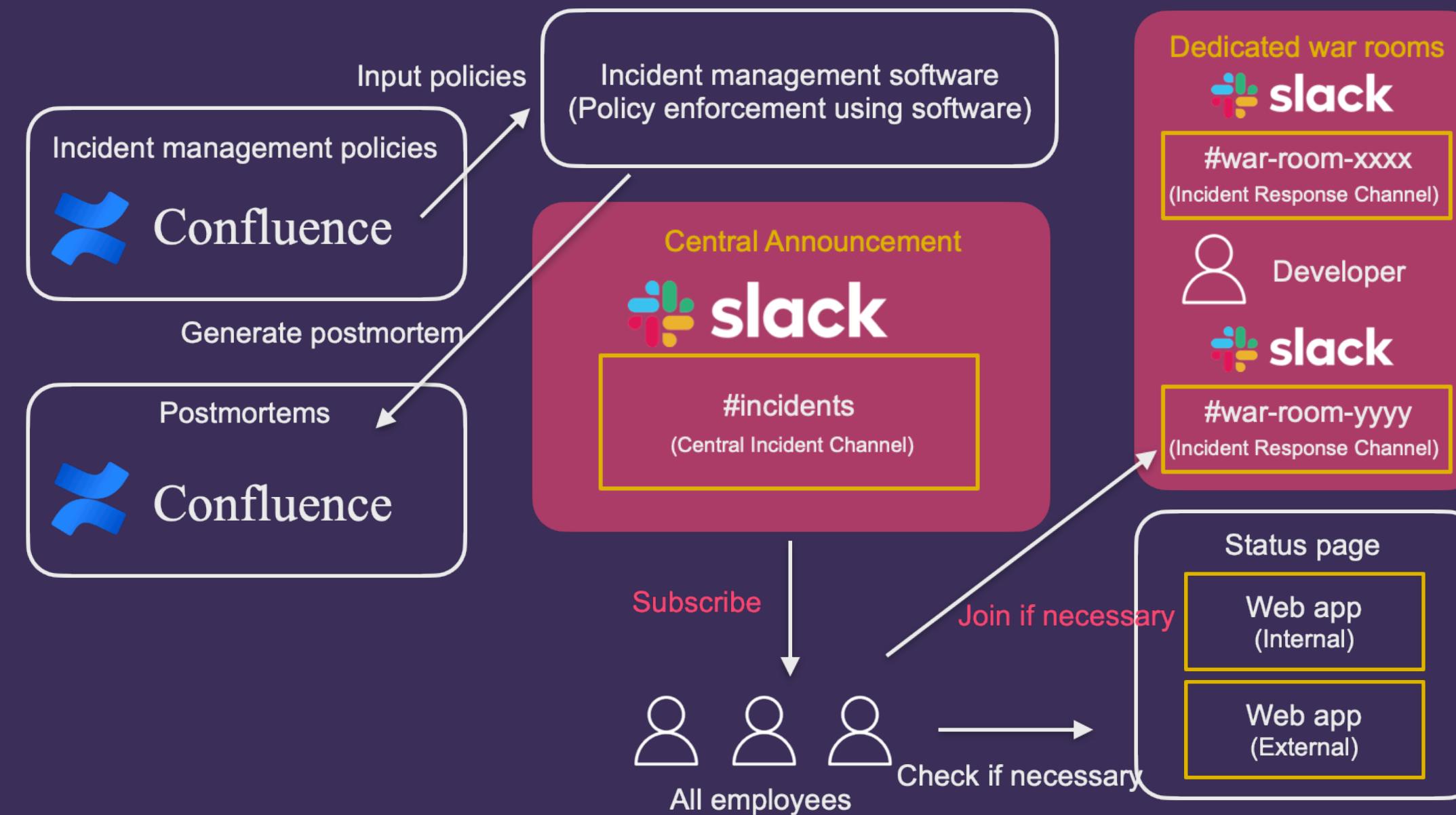
Strategy

- **Enforce incident management policies by software** not by documents.
- Involve appropriate people based on **pull-style communications**.
- **Use the current tool chains** in the company.
 - Too many new tools degrade teams' performance.
 - Use Slack as interactive communication places to keep flow info.
 - Use Confluence to keep stock info (non-urgent communications).

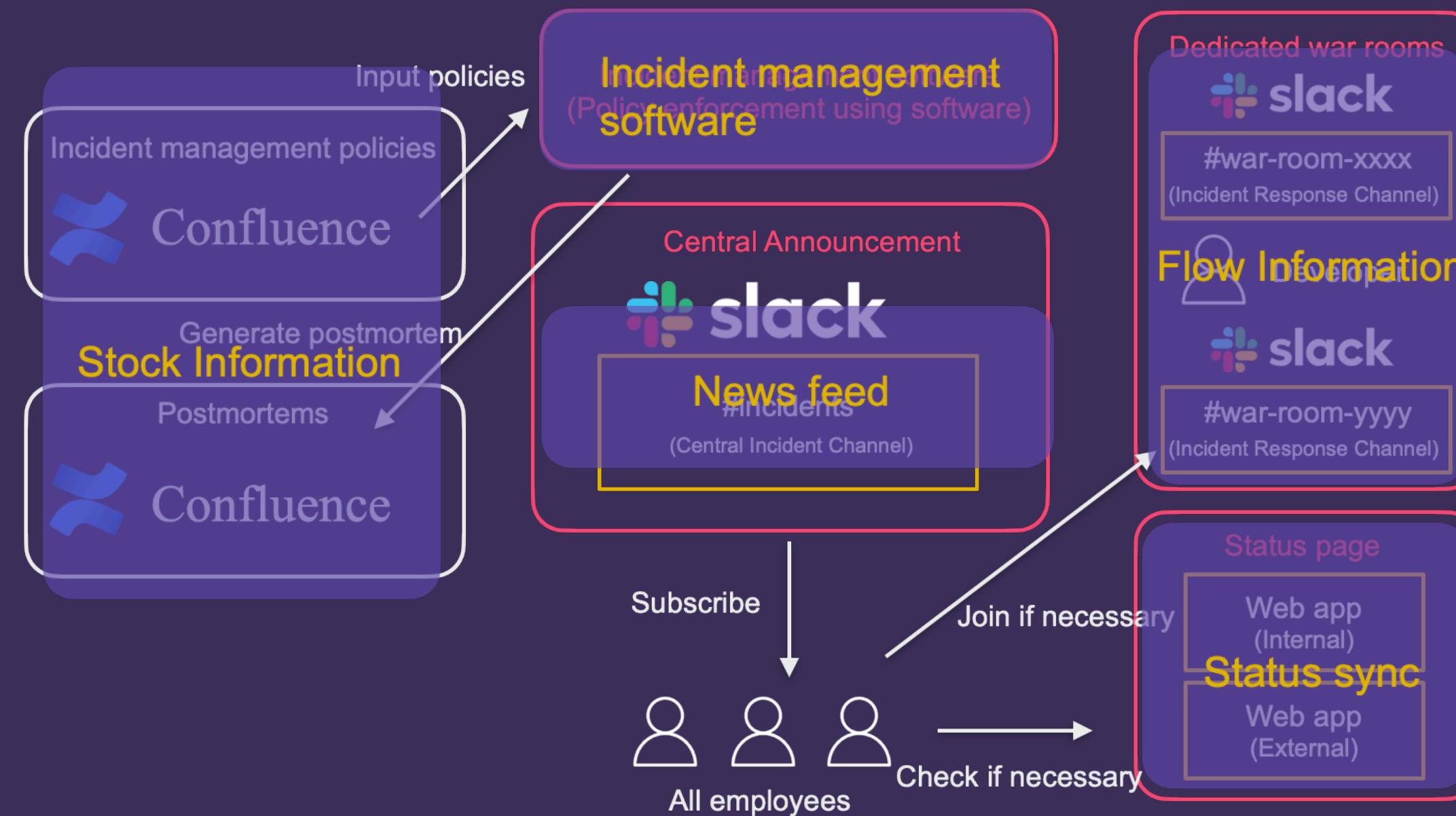
Incident management system (phase 1 to 2)



Incident management system (phase 1 to 2)

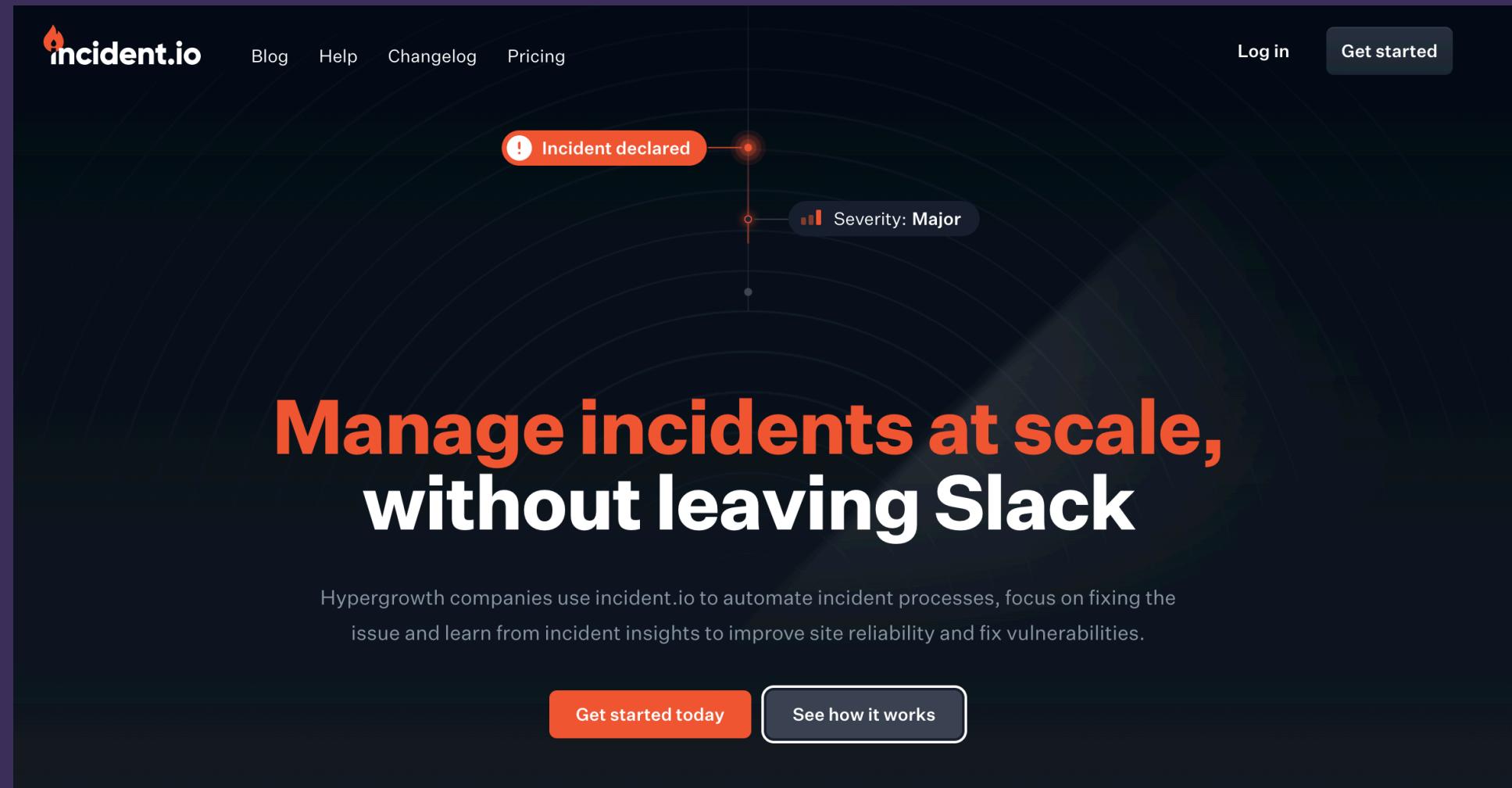


Incident management system (phase 1 to 2)



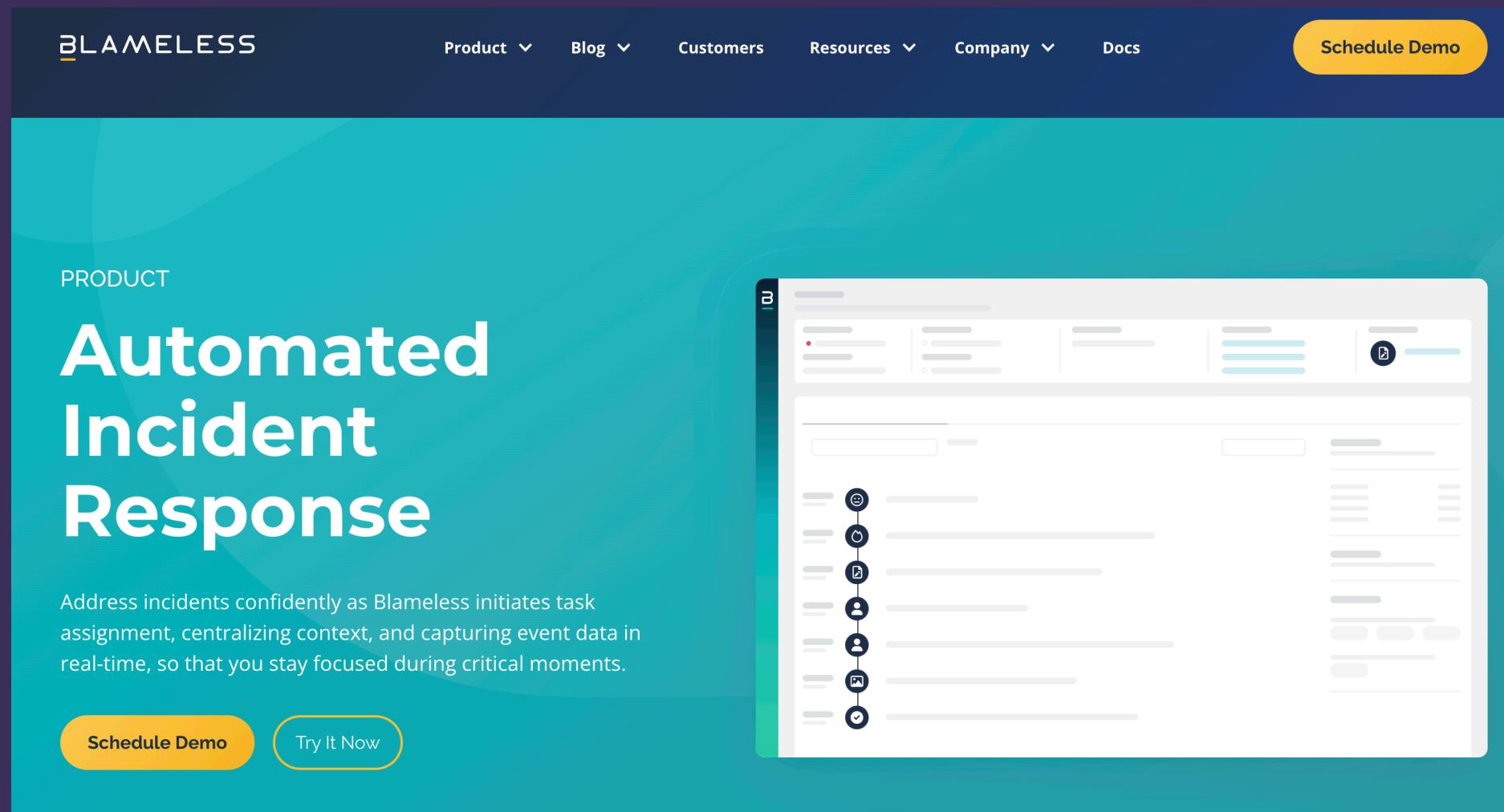
SaaS: incident.io

- <https://incident.io/>



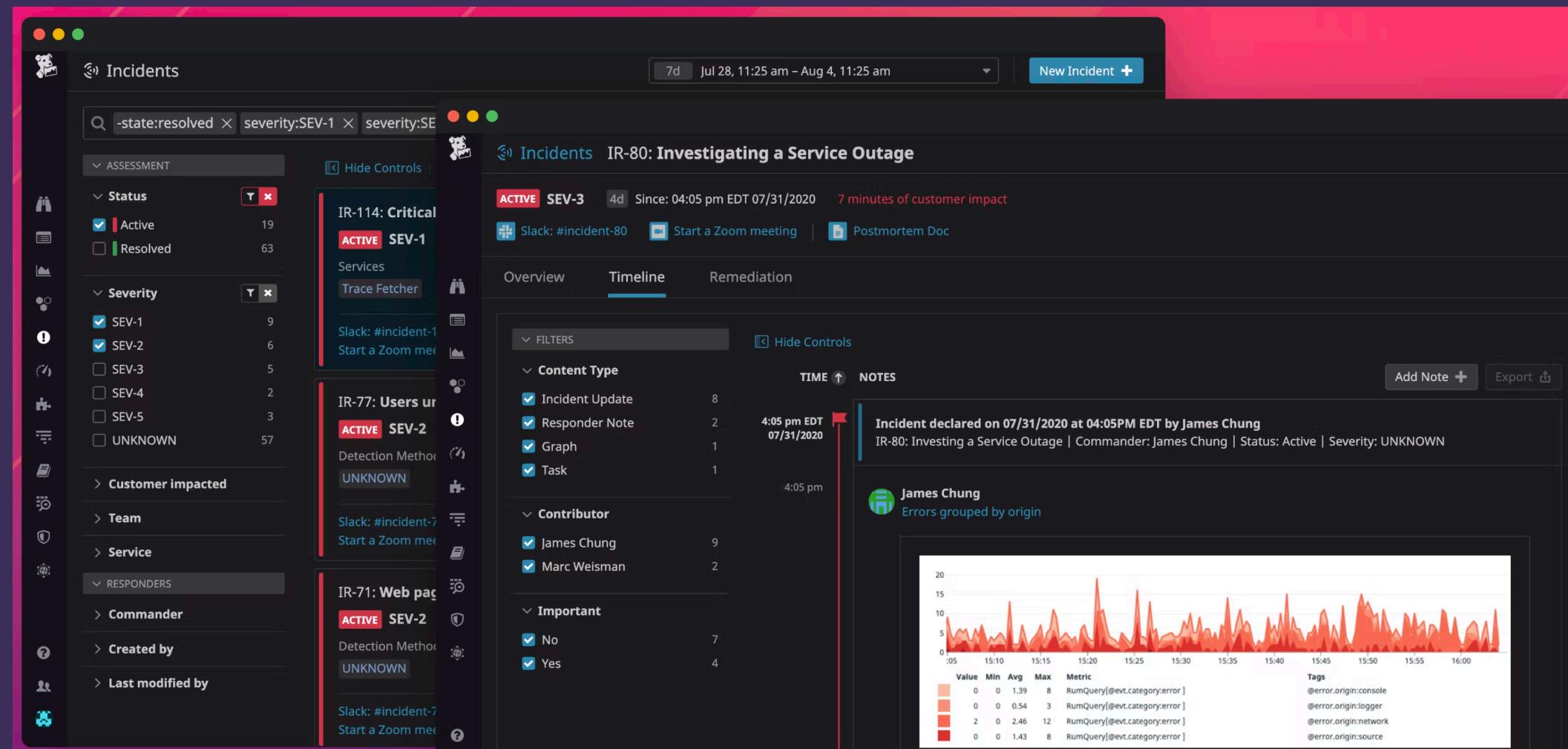
SaaS: Blameless

- <https://www.blameless.com/product/incident-resolution>



SaaS: Datadog Incident

- <https://www.datadoghq.com/blog/incident-response-with-datadog/>



SaaS: Grafana Incident

- <https://go2.grafana.com/incident-beta-interest.html>

The image shows two screenshots related to Grafana Incident. On the left is the 'EARLY ACCESS PROGRAM' landing page for 'Grafana Incident'. It features the Grafana Labs logo at the top, followed by a large heading 'Grafana Incident' and a subtext: 'Grafana Incident makes incident management easier by automating common tasks and giving teams a dedicated place to work.' Below this is a form with a 'Business email' input field and a blue 'Register' button. A note at the bottom states: 'Note: By registering, you agree to be emailed information about this event recording and related product-level information'. On the right is a screenshot of the 'Grafana Incident' dashboard. The title of the dashboard is 'Grafana ML: high CPU usage and latency in prod-us-central'. The dashboard displays several panels: a line chart showing CPU usage and latency over time, a text panel with log entries, and a sidebar with team members and incident details. The sidebar includes links for 'Open Google Docs', 'Join Google Meet', and a list of tasks such as 'Machine Learning latency dashboard', 'Incident 772 dashboard', and 'Google Docs - create and edit documents online, for free'. At the bottom of the sidebar, there are buttons for 'Assign COMMANDER role', 'Assign INVESTIGATOR role', and 'Assign incident severity'.

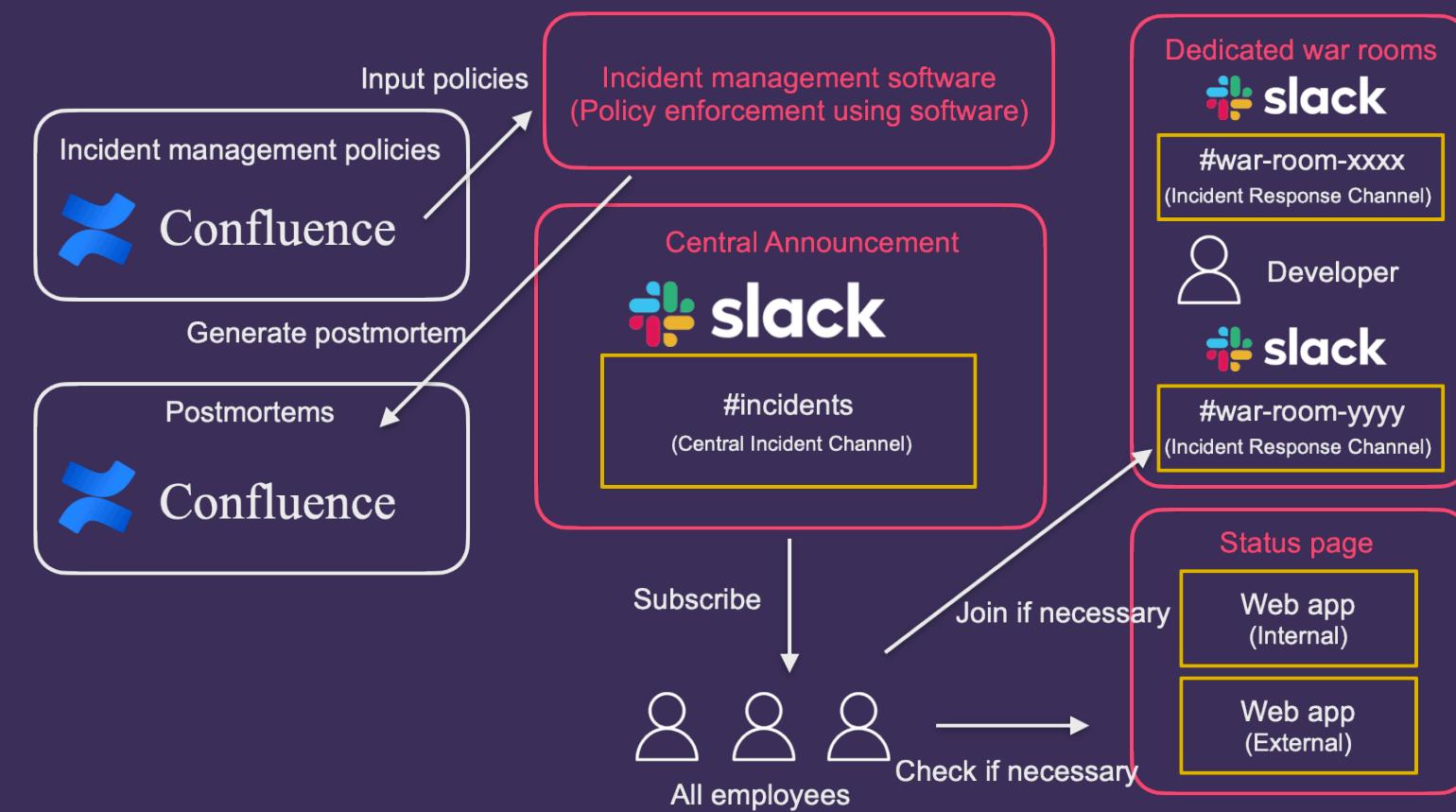
OSS: monzo/response

OSS version of incident.io

- <https://github.com/monzo/response>
- <https://monzo.com/blog/2019/07/08/how-we-respond-to-incidents/>

in-house tool: Slack App + Web App

- It's not so difficult to implement Slack App and Web App for this purpose.
- But... I want to use my time for other stuff.



SaaS vs OSS vs in-house tool

- We want to maximize developers' disposal time for product developments.
- We don't want to increase cognitive loads.
 - OSS and in-house tool needs code and document maintenance.
 - OSS and in-house tool needs evangelical activities for this type of tools.
- **Use SaaS if money allows (Buy, Not Build)**
 - Salaries for software engineers are way more expensive than SaaS cost.
 - SaaS improves their features as their business.
 - SaaS maintains documents as product features.

incident.io covers wide improvement targets

| Improvement target | Actions from phase 0 to 1 | Actions from phase 1 to 2 |
|---------------------------|--|---|
| Transparency | - Encourage push communication | - Encourage pull communication - Create war rooms - Share status pages |
| Tangibility | - Automate parts of incident response flow | - Automate entire incident response flow - Introduce incident lead role |
| Trust | - Introduce blameless culture | - Split lead and operation roles for complex incidents |
| Time to Engagement (TTE) | - Automate incident announcements | - Automate entire incident response flow - Introduce on-call rotations - Expand follow-the-sun coverages |
| Time to Fix (TTF) | - Introduce observability | - Improve observability |
| Time to Triage (TTT) | - Introduce observability | - Improve observability |
| Time to Learn (TTL) | - Introduce postmortem template | - Generate postmortem |
| Time to Preparation (TTP) | - Create incident management policies | - Enforce incident management policies - Self-service incident response trainings |

Central channel for all incidents

incident.io can share all incidents in the specified Slack channel.

incidents incident.io incident.io announcements channel. Every time someone kicks off an incident, we'll announce it here, and make sure the post is always up to date.

incident APP 14:46

✓ NullPointerException when loading a model (Minor)

Wednesday, February 2nd

After deploy production, some tenants(?) happened failed load model

Severity: Minor

Status: Closed

Incident Lead: @ [redacted]

Reporter: @ [redacted]

Channel: #inc-2022-02-02-nullpointerexception-when-loading-a-model

Incident homepage

Last updated at 11:05, Wednesday, February 9th ([\[Main\] custom announcement rule to send announcement to #incidents](#))

2 replies Last reply 2 months ago

Dedicated war rooms (Slack channel)

incident.io handles all tasks we want to complete for incident response initializations.

inc-2022-02-02-nullpointerexception-when-loading-a-model ▾ Lead: @i [REDACTED] | Severity: Minor | Status: Closed | [Homepage](#)

5 Pinned

inc-2022-02-02-nullpointerexception-when-loading-a-model (archived)

@incident created this channel on February 2nd. This is the very beginning of the inc-2022-02-02-nullpointerexception-when-loading-a-incident.io response channel for INC-6

Wednesday, February 2nd

incident APP 14:46 joined #inc-2022-02-02-nullpointerexception-when-loading-a-model.

incident APP 14:46 set the channel topic: Severity: Minor | Status: Investigating | [Homepage](#)

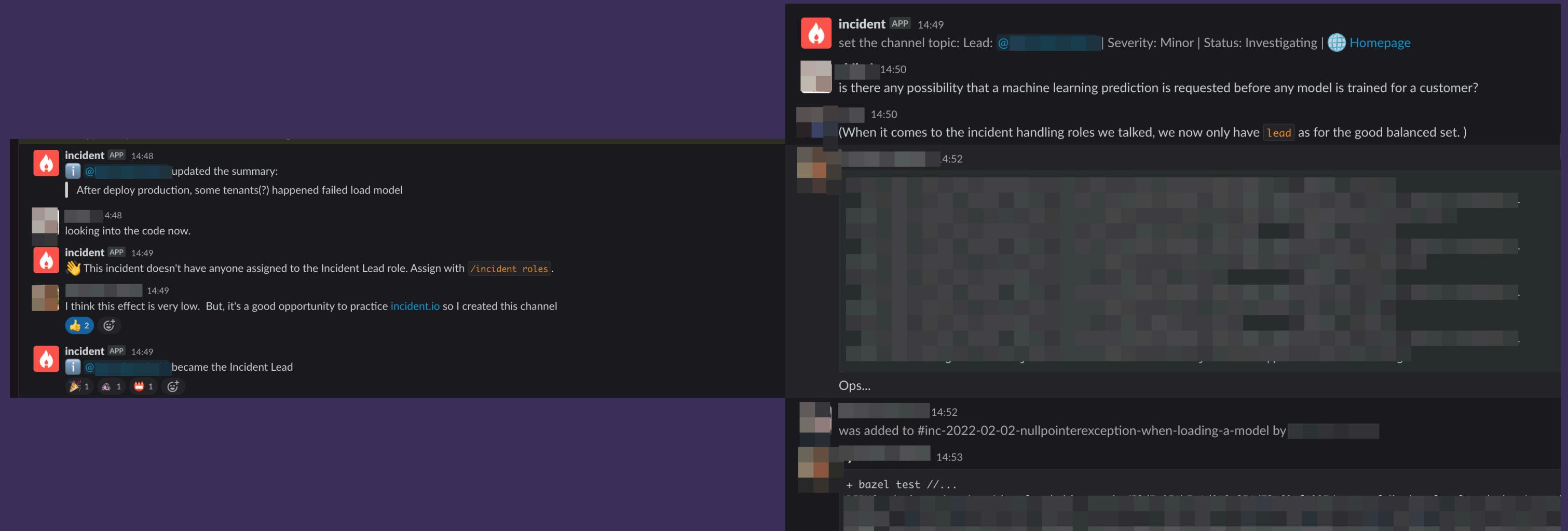
incident APP 14:46 set the channel description: 🔥 An incident.io response channel for INC-6

[REDACTED] 1:46 was added to #inc-2022-02-02-nullpointerexception-when-loading-a-model by incident. Also, [REDACTED] and [REDACTED] joined.

[REDACTED] 4:47 Hello

Dedicated war rooms (Slack channel)

incident.io can assist incident responses.



Dedicated Slack channel (closing incident)

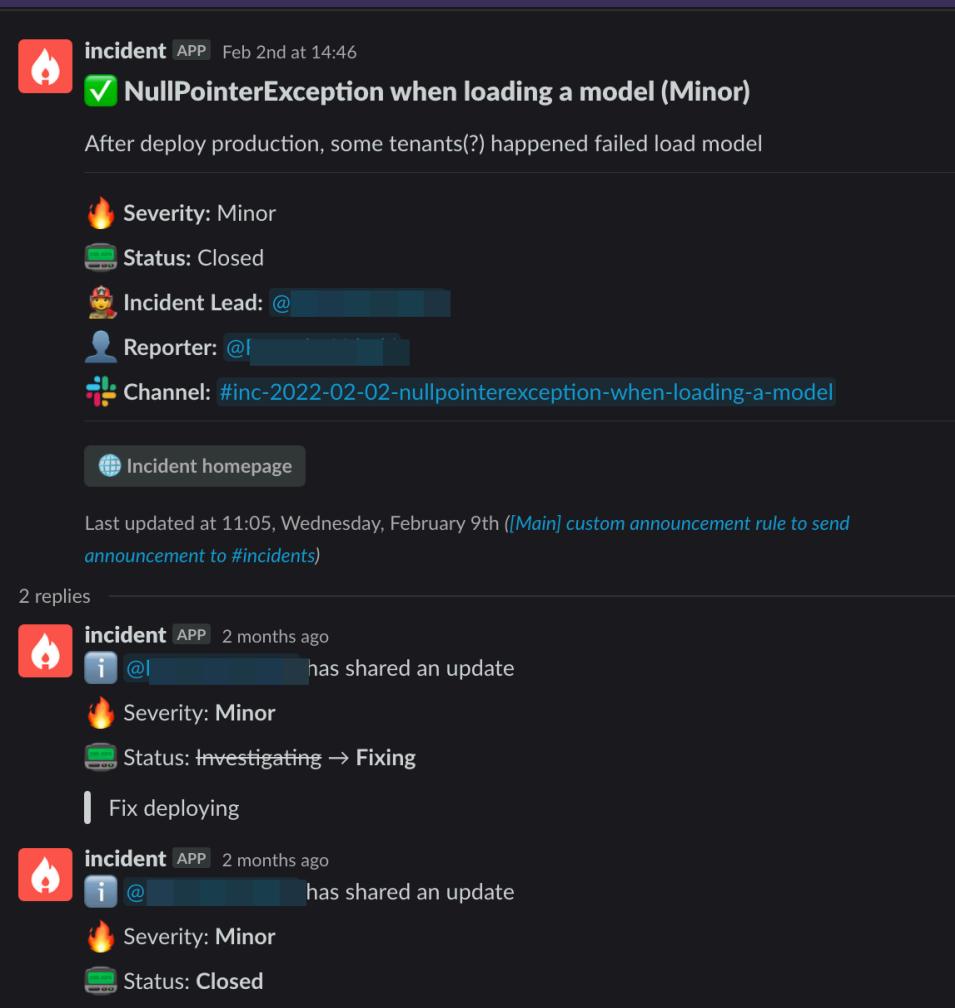
At the end of incident responses, incident.io tells us what we need to be done next.

The screenshot shows a Slack channel interface with the following details:

- Channel Name:** #inc-2022-02-02-nullpointerexception-when-loading-a-model
- Lead:** @I [redacted]
- Severity:** Minor | Status: Closed
- Date:** Wednesday, February 2nd
- Activity:** The channel was renamed from "inc-2022-02-02-nullpointerexception-when-load-a-model" to "inc-2022-02-02-nullpointerexception-when-loading-a-model".
- Topic Set:** The channel topic was set to Lead: @I [redacted] Severity: Minor | Status: Closed | [Homepage](#).
- Closure Message:** An incident app message at 15:24 stated: "The incident is now closed. If you need to re-open, you can do so with [/incident status](#).
- Next Steps:** A section titled "Next steps" lists:
 - Add, update and export any unfinished actions on the incident [Homepage](#).
 - Assign an owner to write up the post-mortem, and arrange a follow up meeting if you feel it'd be helpful.

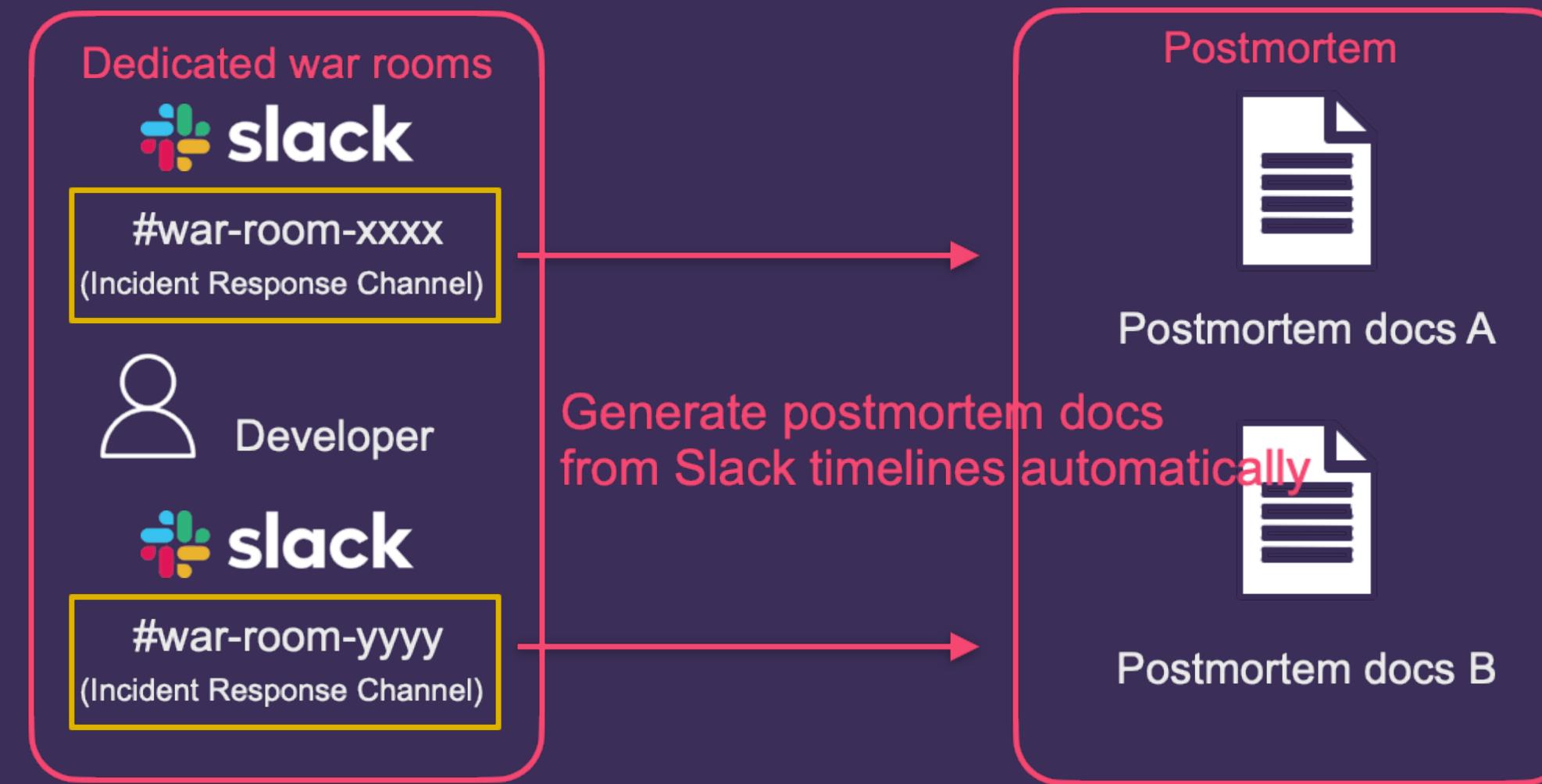
Status updates at central channel

incident.io automatically syncs the latest status of incidents at the central channel.



Postmortem generation

incident.io can collect timelines from war rooms and generates postmortems.



Postmortem generation

We can generate a postmortem documents using incident.io.

2022-02-02 Null pointer exception on prediction vectorizer

Created by [REDACTED] +2
Last updated: Mar 30, 2022 by Yoshiori Shoji • 4 min read • 9 people viewed

| | |
|-----------------|--|
| Status | FINISHED |
| Incident Date | Feb 2, 2022 |
| Completion Date | |
| Incident Level | Sev 2 |
| Incident.io |  incident.io https://[REDACTED] [REDACTED] Connect to preview |

INC-6: NullPointerException when loading a model
Generated Wed, 09 Feb 2022 02:53:06 GMT, by [REDACTED]

Key Information

- Severity: Minor
- Slack Channel: #inc-2022-02-02-nullpointerexception-when-loading-a-model
- Reported: Wed, 02 Feb 2022 05:46:03 GMT
- Identified: Wed, 02 Feb 2022 06:16:56 GMT (+30m 52s)
- Closed: Wed, 02 Feb 2022 06:24:12 GMT (+ 38m 8s)

Team

- Incident Lead: [REDACTED]
- Active participants: [REDACTED]
- Observers: [REDACTED]

Summary

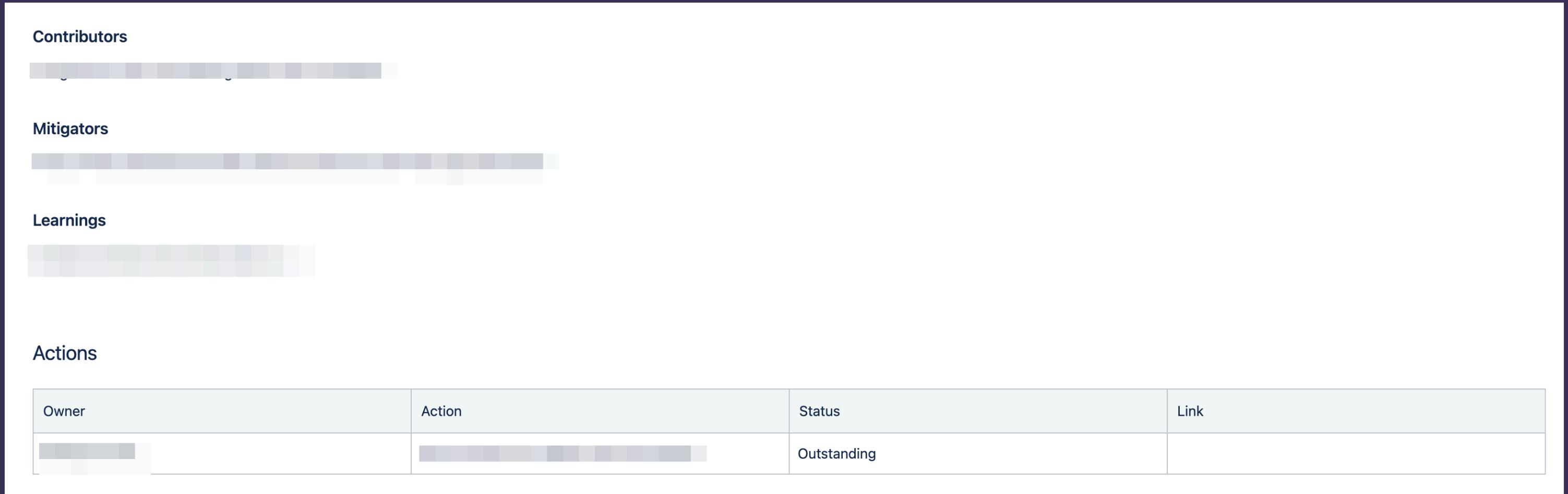
[REDACTED]
After deploy production, some tenants(?) happened failed load model

Postmortem generation

We can collect timelines from dedicated Slack channels.

| Timeline | |
|-------------------------|---|
| Time | Details |
| 2022-02-02 14:46 +09:00 | updated the state to Investigating |
| 2022-02-02 14:47 +09:00 | shared a Sentry Link: https://sentry.io/organizations/ https://sentry.io/organizations/ |
| 2022-02-02 14:48 +09:00 | Takayuki Watanabe pinned message: it's apparently at the code I release this morning. |
| 2022-02-02 14:48 +09:00 | updated the summary: After deploy production, some tenants(?) happened failed load model |
| 2022-02-02 14:49 +09:00 | became the Incident Lead. |
| 2022-02-02 15:02 +09:00 | shared a GitHub Link: 1 2 3 4 5 6 7 8 9 10 |

Postmortem generation



Self-training mode

incident.io has a mode to walk though dummy incident responses on Slack.

The image consists of two side-by-side screenshots of a Slack conversation. Both screenshots feature a dark theme.

Left Screenshot (Initial Setup):

- A user was added to the channel at 16:04.
- An incident app message at 16:04: "incident APP 16:04 was added to #inc-2022-01-25-tutorial-hazelwood-horse by incident."
- A user sends a message: "incident APP 16:04 🎉 incident.io Tutorial"
- The incident app responds: "👉 Introduction (Step 1 of 7)"
- The message continues: "Welcome to the incident.io tutorial! Here, we'll show you the basics of running an incident. In this scenario, the office cat 🐱 has gone missing, and you're in charge of coordinating the rescue. Usually, we'd post all of the relevant information as an **incident announcement** in the #incidents channel to let everyone know what's happening. We'll automatically keep it up to date for you as things progress. For this tutorial we'll post it in here."
- A button: ">Show me the announcement"
- An incident app message at 16:11: "incident APP 16:11 🎁 tutorial-hazelwood-horse (Critical)"
- The message continues: "The office cat is stuck up a tree"
- Information about the incident: "Severity: Critical", "Status: Closed", "Incident Lead: @[redacted]", "Reporter: @[redacted]", "Channel: #inc-2022-01-25-tutorial-hazelwood-horse".
- A button: "Incident homepage"

Right Screenshot (Final Step):

- An incident app message at 16:18: "incident APP 16:18 set the channel topic: Lead: @[redacted] | Severity: Critical | Status: Closed | 🌐 Homepage"
- The incident app responds: "The incident is now closed. If you need to re-open, you can do so with /incident status."
- A section titled "Next steps":
 - Add, update and export any unfinished actions on the incident [Homepage](#).
 - Assign an owner to write up the post-mortem, and arrange a follow up meeting if you feel it'd be helpful.
- A user sends a message: "incident APP 16:18 🎉 incident.io Tutorial"
- The incident app lists completed steps:
 - ✓ Introduction
 - ✓ Set the incident lead
 - ✓ Send an incident update
 - ✓ Create an action
 - ✓ Pin a message to the timeline
 - ✓ Close the incident
- A button: "View the incident in the web app (Step 7 of 7)"
- The message continues: "The incident may be over, but we can still learn from it. Let's head to the incident homepage to check out the incident in more detail and continue the tutorial on the web."
- A button: "Open Incident Homepage"
- A timestamp: "16:18 🏆 Tutorial complete"

Introduction of lead role

- We need communication leads when incidents are complex
- However, for most of incident, a single person can be responsible for operations and communications.
- So, adding a lead role only is prudent so we don't make incident managements overly complex.

We have more rooms to improve!

Recap

- Incident management has a life cycle.
 - Preparation -> Detection -> Recovery -> Post-incident actions -> Preparation
- Incident response roles and structures exist to embody 3T.
 - Transparency
 - Tangibility
 - Trust
- Choosing strategy and tools makes incident managements at startups sensible.

Thanks

References

Incident management

1. Atlassian: Understanding incident response roles and responsibilities, <https://www.atlassian.com/incident-management/incident-response/roles-responsibilities>
2. PagerDuty Incident Response Training, <https://response.pagerduty.com/training/overview/>.
3. Anatomy of an Incident, Ayelet Sachto, Adrienne Walcer, and Jessie Yang, 2022.
4. US Federal Emergency Management Agency, Emergency Management Institute ICS Resource Center, <https://training.fema.gov/emiweb/is/icsresource/>.
5. The National Institute of Standards and Technology SP 800-61, Computer Security Incident Handling Guide, <http://dx.doi.org/10.6028/NIST.SP.800-61r2>.
6. Introduction: Incident Response overview, Gov UK National Cyber Security Centre, <https://www.ncsc.gov.uk/collection/incident-management/incident-response>
7. Incident Review and Postmortem Best Practices, <https://newsletter.pragmaticengineer.com/p/incident-review-best-practices>
8. Incident Review Practices [The Pragmatic Engineer Newsletter], <https://docs.google.com/spreadsheets/d/1GPINipdf-I2H05iKOUpkrqwIz61ZCJDnwY5iE8LtRM/edit#gid=0>

SRE

1. Google SRE book Chapter 14 - Managing Incidents, <https://sre.google/sre-book/managing-incidents/>
2. Postmortem Action Items: Plan the Work and Work the Plan, Sue Lueder and Betsy Beyer (Google), USENIX SRECon 2017, <https://www.usenix.org/conference/srecon17americas/program/presentation/lueder>.
3. Google SRE book Chapter 15 - Postmortem Culture: Learning from Failure, <https://sre.google/sre-book/postmortem-culture/>.
4. Postmortem Metadata Index, <https://postmortems.app/>.
5. The Art of SLOs, Google Site Reliability Engineering, <https://sre.google/resources/practices-and-processes/art-of-slos/>
6. danluu/post-mortems: A collection of postmortems, <https://github.com/danluu/post-mortems>.
7. Great Incident Review Examples, The Pragmatic Engineer, <https://blog.pragmaticengineer.com/postmortem-best-practices/#great-incident-review-examples>

DevOps performance metrics

1. Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations, 2018.
2. GoogleCloudPlatform/fourkeys, <https://github.com/GoogleCloudPlatform/fourkeys>
3. Are you an Elite DevOps performer? Find out with the Four Keys Project, Google Cloud, <https://cloud.google.com/blog/products/devops-sre/using-the-four-keys-to-measure-your-devops-performance>
4. DORA DevOps Quick Check., <https://www.devops-research.com/quickcheck.html>

SaaS and OSS

1. Datadog, <https://www.datadoghq.com/blog/incident-response-with-datadog/>
2. incident.io, <https://incident.io/>
3. jeli, <https://www.jeli.io/>
4. monzo/response, <https://monzo.com/blog/2019/07/08/how-we-respond-to-incidents>
5. Etsy/morgue, <https://github.com/etsy/morgue>