Finding what's important: Actionable, Automated and Accurate Alerting

DATADOG

The Problem

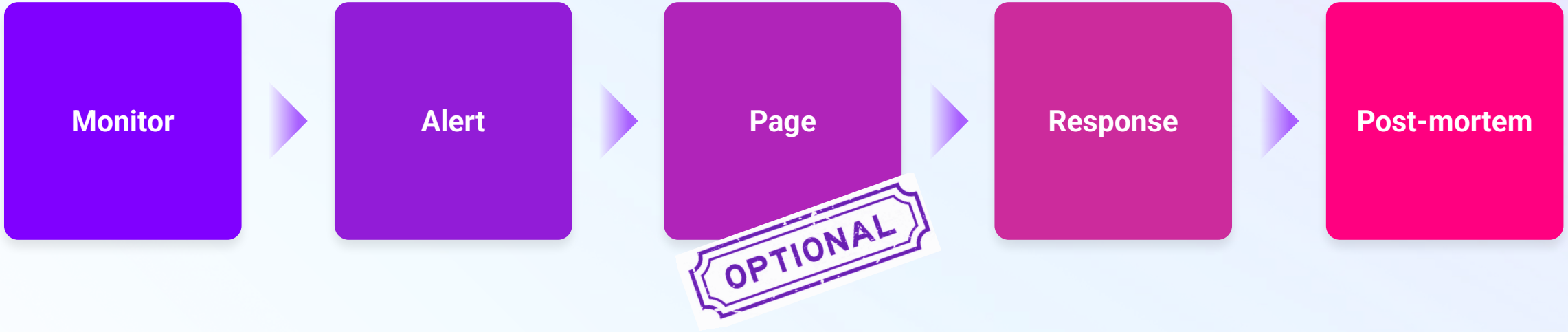
Kennedy Toomey

Application Security Researcher & Advocate @ Datadog

Max Saltonstall

Technical Storytelling Advocate @ Datadog

Life cycle of an issue



Today’s solutions (?)

Ignore them

Waste time fixing

Add to the backlog

Four stages of alerting grief

Toil

Frustration

Boredom

Burnout

Types of alerts

Page

Urgent & needs immediate attention

- high trust alerts
- annoying but necessary
- use sparingly

Slack/Teams message

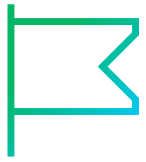
Important but is not drop everything urgent

- should be first priority during working hours

Email/Ticket

Investigative or informational

- Best if grouped as a summary



No Alert

Low priority or no action needed

- visible in monitoring/security platform

Making the RIGHT alerting decisions

Is the issue **real?**

Is the issue **urgent?**

Does the issue **require attention**?

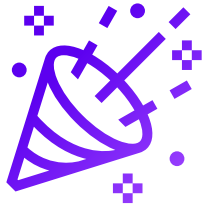
Is the fix something that is **NOT automated**?

What to alert on

Alert hierarchy

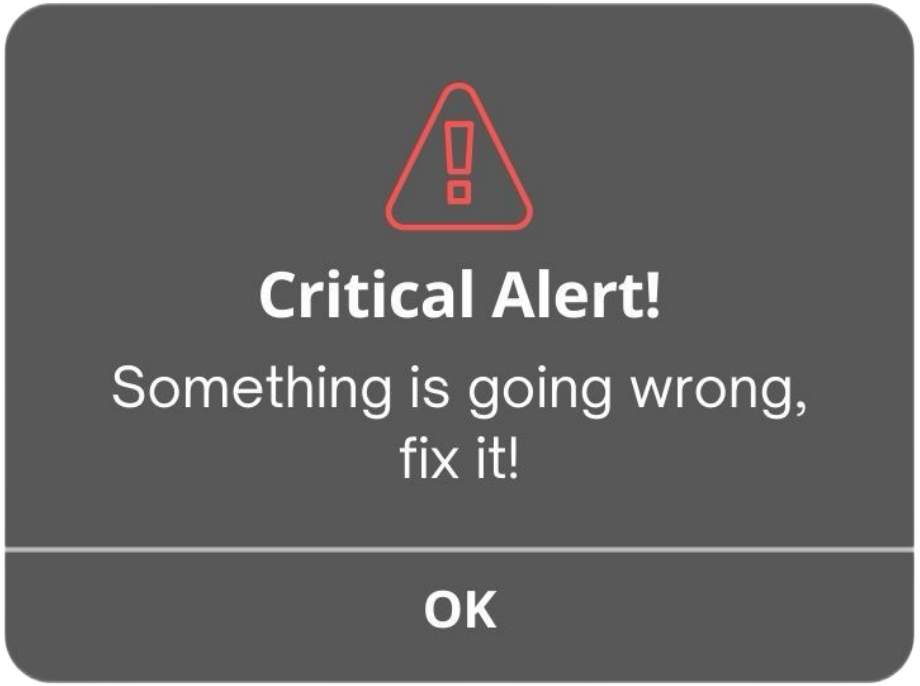
Bad vs Good

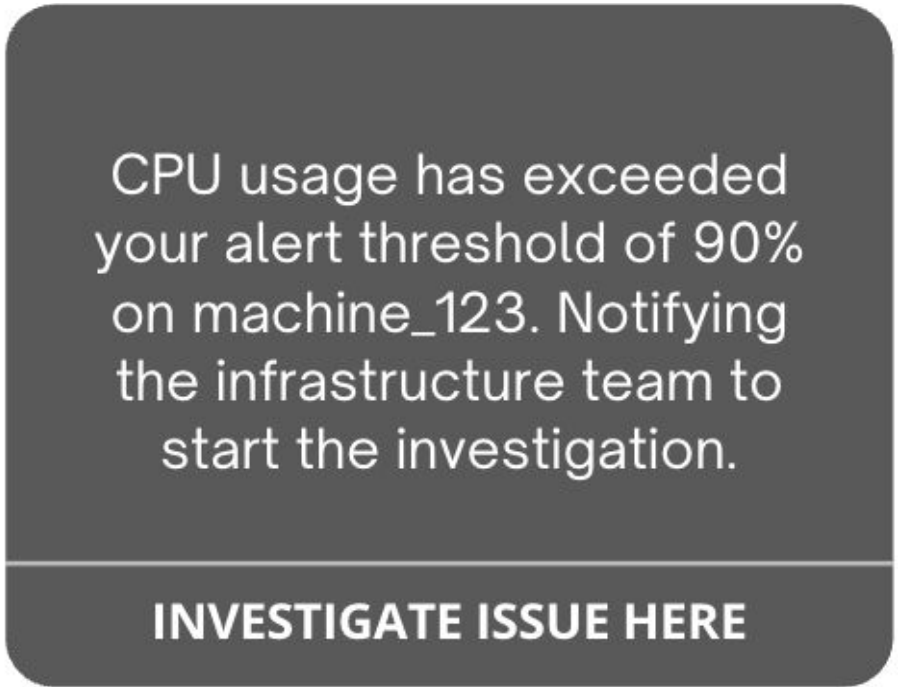
Too late
Wrong person or team
Lack context
Nothing for a person to do

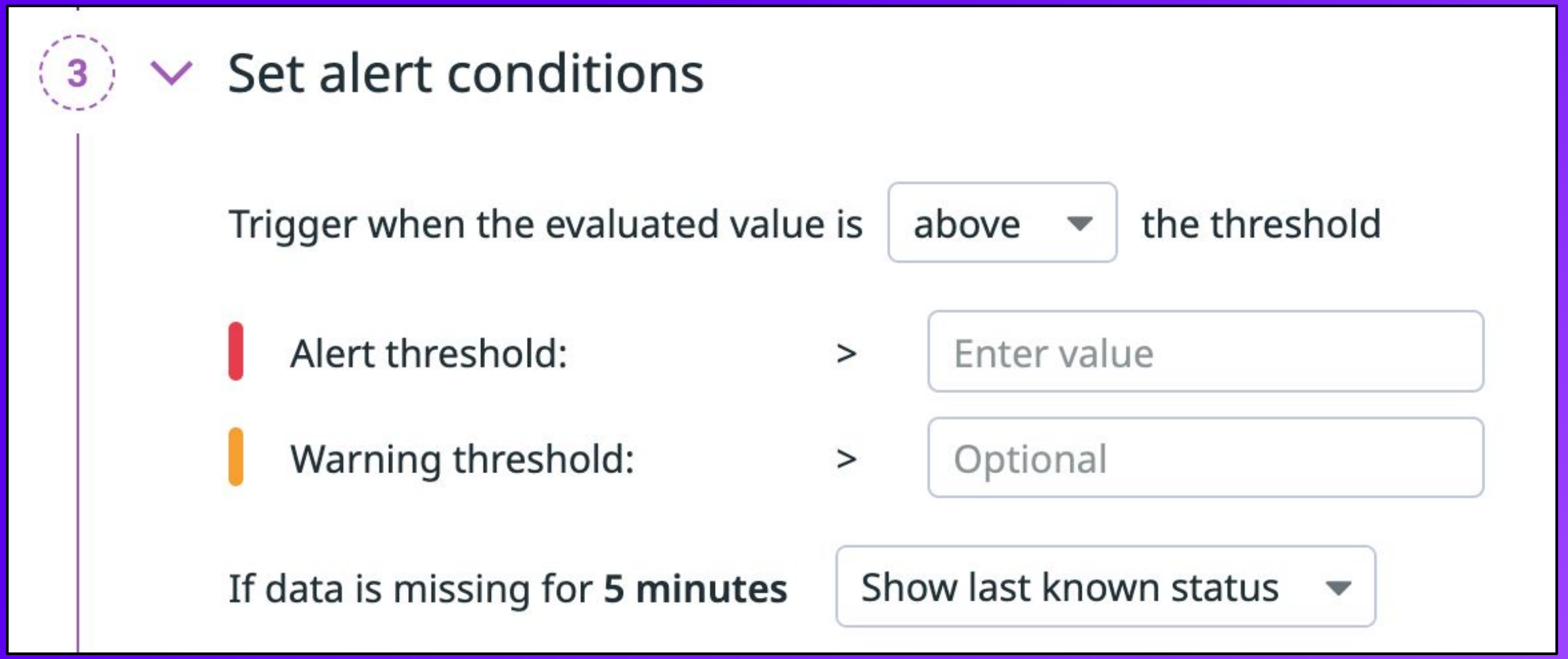


Timely
Delivered correctly
Documented properly
Actionable

Bad vs Good


Critical Alert!
Something is going wrong, fix it!
OK


CPU usage has exceeded your alert threshold of 90% on machine_123. Notifying the infrastructure team to start the investigation.
INVESTIGATE ISSUE HERE


3
Set alert conditions
Trigger when the evaluated value is
above
the threshold
Alert threshold:
>
Enter value
Warning threshold:
>
Optional
If data is missing for 5 minutes
Show last known status

1 For every new...

Finding

A security weakness in an asset that can be exploited by a threat.

Signal

Potential suspicious activity (active threat targeting the infrastructure)

App & API Protection

2 Which has...

Any of these severities

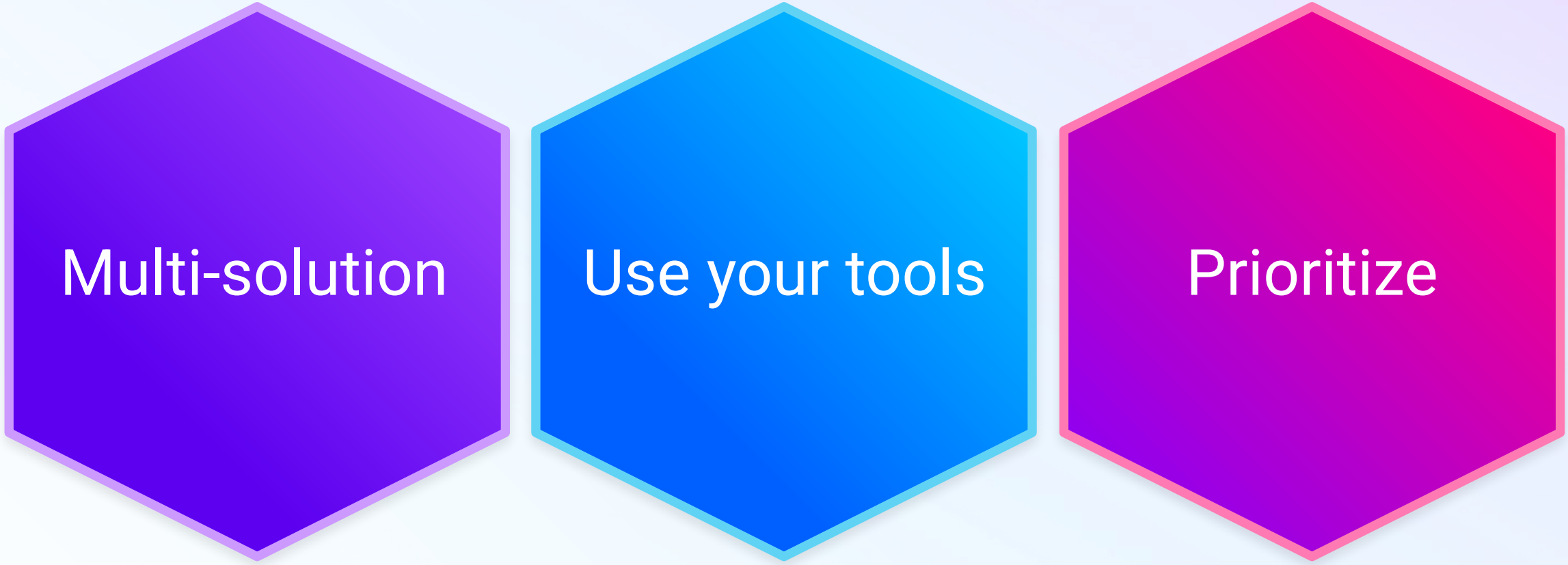
CRITICAL

Any of these tags or attributes (e.g., env, service, account_id, ...)

Start typing a tag or an attribute

Tip: Reduce noise in your alerts by excluding certain attributes using a hyphen, e.g.: `-@resource_type:kubernetes*`

Customization is key



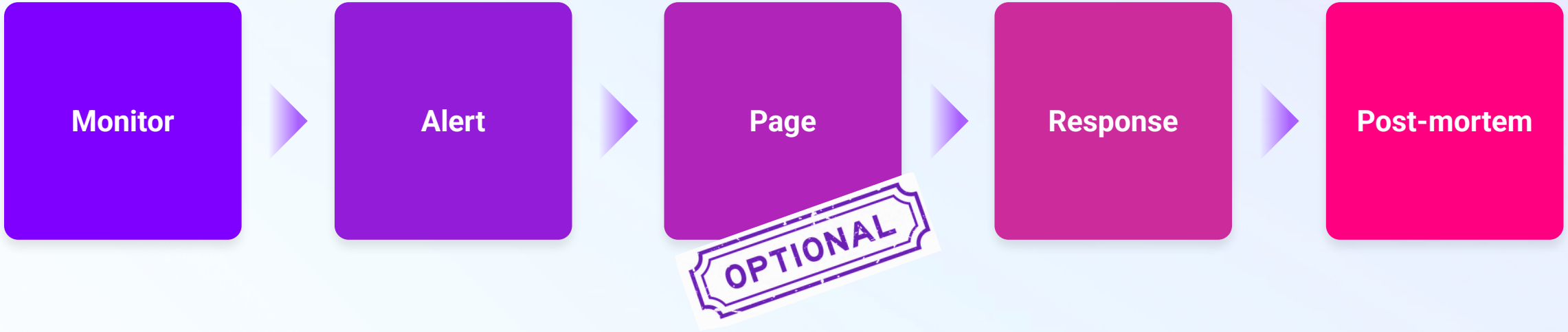
Priority is situational

What's important to the business?

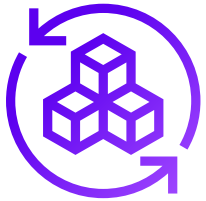
What does your team need?

What are the key metrics?

Life cycle of an issue



Prepare for future incidents



Telemetry data from past incidents

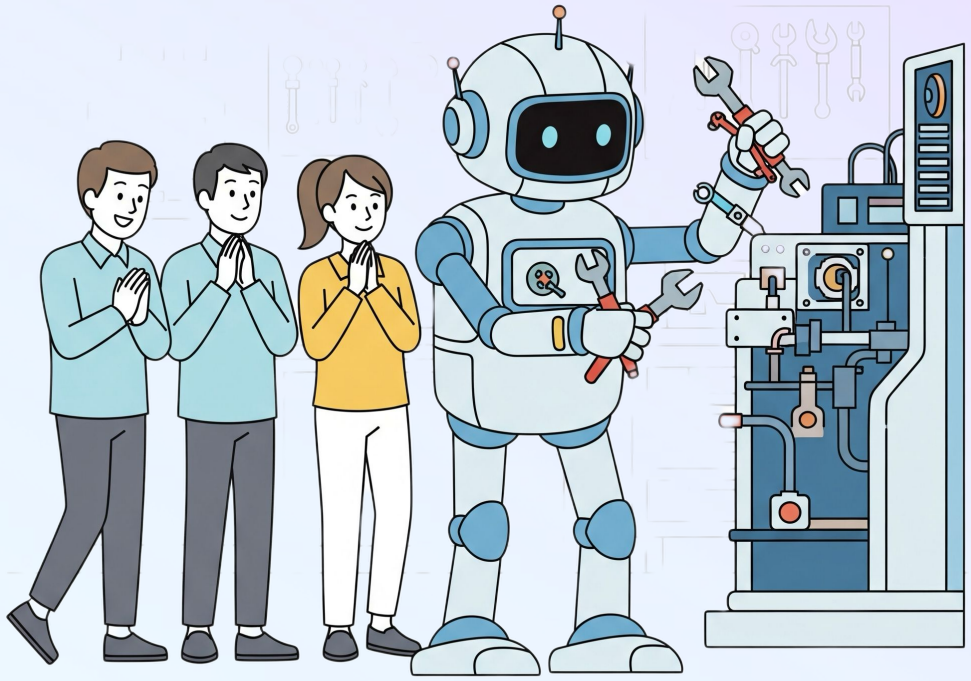
Earlier detection & diagnosis

Create new alerts

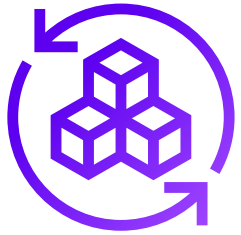
Automate where possible

Does the alert require human interaction?

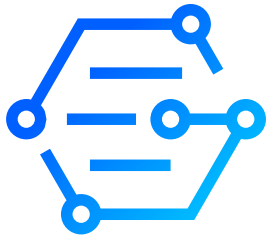
Can a robot do it for me?



Using AI to help



Determine priority & forecast

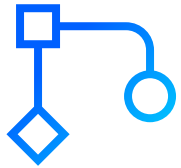


Create automations

Fix the issue

Ideal outcomes

More time on the right problems



More space to automate

More proactive, less reactive

Faster root cause analysis

So what?

Do not wake people up with alerts

So what?

Do not wake people up with alerts *unless* it is urgent

So what?

Do not wake people up with alerts

unless it is urgent

and

So what?

Do not wake people up with alerts
unless it is urgent
and
requires human interaction

A final reminder:

not everything is urgent!

Thank you - Questions?