Incident Management Metrics that Matter

Jamie Luck (they/them)
delucks@datadoghq.com
Laura de Vesine (she/her)
silverrose@datadoghq.com

DATADOG

Who We Are

Jamie

Document read/writer
Blameless everything
As old as my eyes and a bit older than my teeth
Has a million polite ways to say no
Owns more floppy disks than you'd expect

Laura

General incident management busybody
Full of opinions on "good" incidents
Routinely joins incidents holding a cat
Thinks dumpster fires are warm and cozy
Has not seen it all; is jaded anyway
Big fan of "people over process"

[Jamie] Since it's relevant to this talk, let's take a minute to introduce ourselves in a bit more detail. Hi, I'm Jamie. I'm a senior engineer at Datadog and I just finished a few months as the interim manager of Datadog's internal incident management team, responsible for maintaining and improving our tooling, process, and data collection for incidents internally. I'm also a huge vintage technology nerd, please come talk to me about my unix workstations.

[Laura] And I'm Laura – I'm a senior staff engineer at Datadog with a broad scope around "we should be reliable and resilient", which obviously includes making sure that our incident management process and tooling are working well and keeping customer impact from unforeseen events to an absolute minimum

But for this talk, I'll will be playing the role a new engineering manager at Datadog who's getting ready to make an impact in the incident management space.

[Images on slide: a candid photo of Jamie wearing a brightly colored shirt, and a photo of Laura holding a threatening pose with a throwing axe]

[jamie onboarding new busybody] Let's also give some context on what kind of company Datadog is: while we're not FAANG-sized, it's a pretty large company, with a bottom-up engineering culture. We practice distributed incident management, so any engineer might be involved in any incident, including needing to write the postmortem. Every engineer eventually commands or responds to an incident. We have a dedicated team staffed to build sustainable incident processes and re-evaluate our posture for incident management and on-call.

[laura drawing conclusions] So how do executives know if it's worth employing the team? We obviously need to understand how we're doing at making sure our incidents are under control, and that the work the team is doing to make them better is business dollars well-spent. After all, we definitely want to see if there's a company-level need for a major correction, ideally before it gets really bad (and also, I want execs to give me fat bonuses). How do we prove we're doing a good job, in executive speak? [dramatic pause for thought, then next slide]

Image made by AI; honestly shocking how hard it was to get it to make this one. Try getting a "the fish was this big" gesture out of one yourself!

[Image on slide: a white fisherman wearing an orange jacket, making a “it was this big” gesture]
[Zebrafish image from wikipedia: https://commons.wikimedia.org/wiki/File:202101_Zebrafish.png]

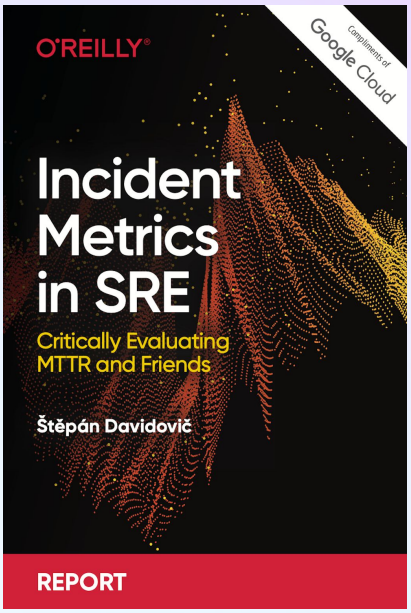
[laura] I know! we should measure Mean time to recovery from incidents! After all, if our incident response is effective, we'll recover from them faster. So MTTR going down means we have good incident response.

fireworks image from https://pngimg.com/image/15671
[Images on slide: "MTTR" surrounded by brightly colored fireworks]

How about not?

MTTR is the easy path

But like the Dark Side, it will lead you astray



[Jamie]: *Sigh* That's an easy statistic to measure, but what kind of value does it actually bring?

- Generates a picture of reliability that's inaccurate and overly simplistic - incidents are complex and one measure isn't enough to capture that complexity
- It's not a robust summary statistic (mean!)
    - Stepan Davidovic (approx pronunciation: "shtay-pahn dah-weed-oh-wich") has actually done this statistical analysis. For incidents specifically, the number of incidents you're calculating a mean for, and the size of the standard deviation, mean that changes you measure in MTTR are almost deterministically noise in your data, not meaningful changes.
- Creating perverse incentives (so if we resolve incidents faster we have better outcomes)
    - The easiest way to drive down MTTR is therefore to have *the same* incident over and over, so you get really fast at fixing it. And that doesn't seem like what we want…

And before you say it, no, MTTM and the various cousins aren't any better – they have the same statistical problems, and the same perverse incentives, plus, you can get into fun arguments about when the incident is really "mitigated" or what "impact" directly means.

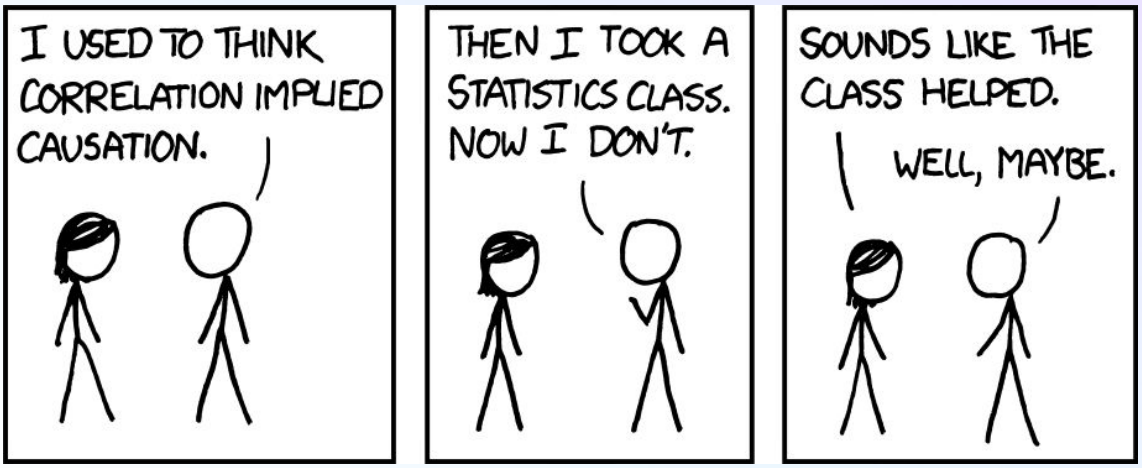
[Laura] Okay fine, so you're telling me that incidents can be really variable in length and we shouldn't be measuring how long they are. I get that, your argument makes sense. Hmmmm. Okay, what if we measured [click]

"Come to the dark side we have cookies" image generated from AI
Incident metrics in SRE image from
https://sre.google/resources/practices-and-processes/incident-metrics-in-sre/
[Images on slide: Darth Vader holding a cookie, labeled "MTTR", and the cover of "Incident Metrics in SRE" by Stepan Davidovic]

[Laura] Incident count! Surely if we are doing our jobs right we'll have fewer incidents over time – it only makes sense. We want things to break less, we see that by having fewer incidents. Q.E.D Jamie!

fireworks image from https://pngimg.com/image/15671
[Images on slide: "Incident count" surrounded by brightly colored fireworks]

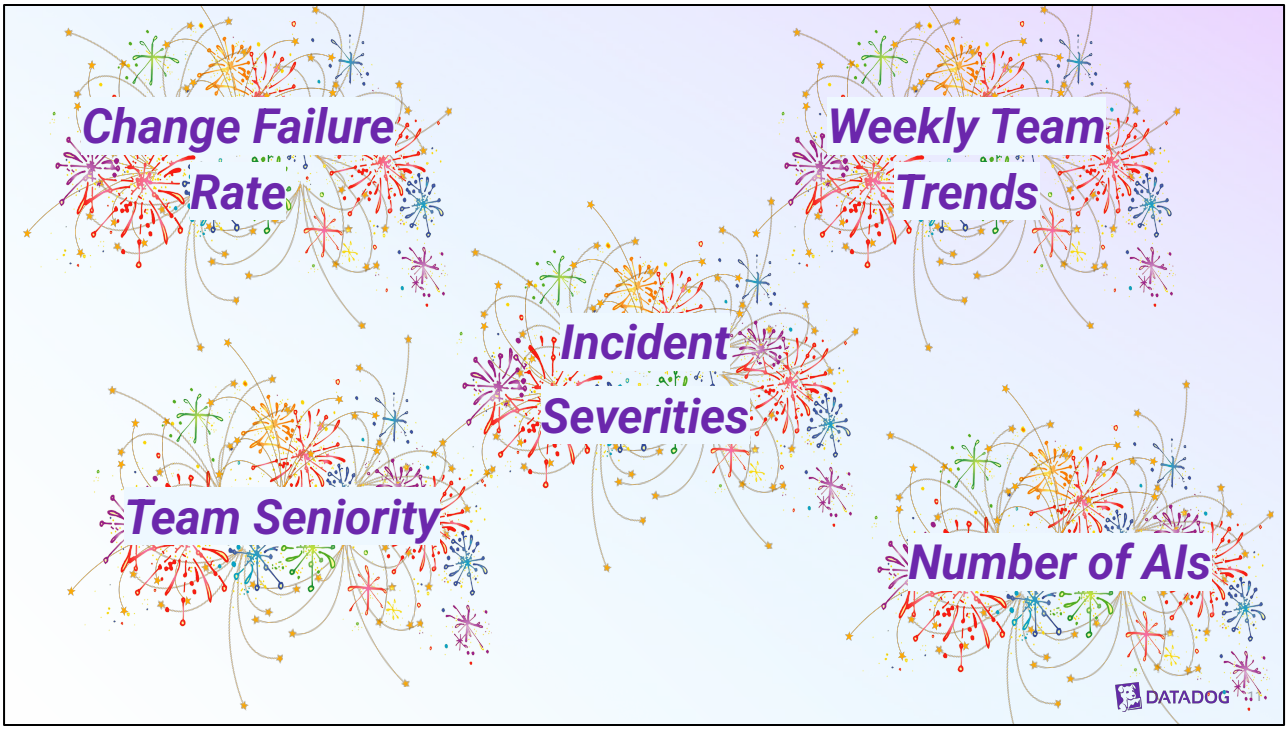
Ehhhhhh



Jamie: *Sigh* Okay. That's also an easy statistic to measure, but what does an increase or decrease in incident count actually mean?

- Correlation instead of causation
  - Were there holidays? Code freezes? Is your business periodic?
- Bad incentives
  - Encourages people to not file incidents for things which are incidents
  - Lower severity incidents just become "spicy bugs"
  - Loss of visibility into what's actually broken

Image from xkcd: https://xkcd.com/552/
[Image description: cueball says "I used to think correlation implied causation. Then I took a statistics class. Now I don't." Longhair responds: "sounds like the class helped". Cueball: "Well, maybe"]



[Rapid fire]

- Laura: Change failure rate: it's like incident count, but normalized to how big our systems are and how fast they change!
- Jamie: what is a change, what is failure? This gives a bad representation of how problems can build up over time and how changes are interrelated
- Laura: Fine, but we know people are following up diligently if their postmortems have *lots* of action items!
- Jamie: encourages the creation of low-value AIs, especially because this one is usually paired with a measure of AI completeness
- Laura: How about week over week trends for individual teams! We can see at the team level if things are getting better
- Jamie: Why not just make it hour over hour? There's way too much random variation in the data for this level of granularity to be valuable
- Laura: Okay, okay, but like… we want our incidents to be less bad, right? How about if we measure how many of our incidents are severe, it'll help us reduce the number of severe events!
- Jamie: Once again, this sets up poor incentives to set the correct

- severity for an incident, discourages updating the severity as conditions change, and ultimately decreases the value of severity as a coordination concept
- Laura: I dunno, like… why not team seniority levels? We know more senior engineers are better so they should only break things when it was actually hard, right?
- Jamie: Uh. This seems like a great way to break a bunch of incentives, create a blameful culture, and set all your engineers against each other? – this is obv bad

[Images on slide: each suggested metric is surrounded by fireworks and flies in as it is suggested]

Okay but like, what do we actually measure?

Laura: okay look, I get that it’s hard to measure incident success – you’ve made very clear that incidents are complicated and variable and there’s a lot of human interactions going on so we have to think hard about the incentives we’re building. But meet me halfway here – the business wants to know if we're being effective (and we should want that for ourselves as well!), and it's not okay to just say "trust us" or have no insight into whether our practices are working or making things better

Jamie: So why don’t we try breaking down the problem, and doing a little separation of “incident response” as a goal in itself. We can agree that no matter how good our engineers are, sometimes things *will* break, right? So if we’re effective at responding to that – if we have an effective incident response program – what things would be true? And how do we show we're getting there?

Crying cat image generated by AI
[Image description: a cartoon cat looks sad and defeated, as its eyes drip tears]

What's "success" for oncall?

Engineers treat oncall seriously

Engineers respond to pages promptly

The tooling for "being oncall" works well

[Laura] Hey that's a really good question! So… what *are* my goals as an Official Incident Busybody around people being oncall? Well, I really want that to mean that our engineers are "good at being oncall" – that is, I want to know that engineers who are oncall show up to incidents, know how to use their tools, and that their tools work well. I guess what I'm really after is just… knowing that engineers are responsible about *being* oncall, and that that process is doing what we ask it to ("getting a person with the power and knowledge to fix it alerted and ready to act quickly"). I want to know that incidents are getting treated as the absolute, interrupt-level priority that they are; that people are able to join and respond quickly; no one spends their oncall shifts being unreachable or unable to deal with incidents that arise, thinks like that.

Silly pager image generated by AI
[Image on slide: a calculator (doesn't really look like a proper pager) says "U UP?" on the screen, and is surrounded by word bubbles saying "BHUP" and "WO?!"]

Let's measure that

We trust our engineering culture to be responsible

Oncall tooling: is it visible?

[Jamie; click to stop graph loop]
> Engineers taking oncall seriously / being responsible

- At Datadog, we're confident that this is culturally true globally, and genuinely don't feel the need to measure it for success. If we wanted to, we could look at things like "time to answer pages" or similar, but let's not measure things we don't think need watching. We know our managers and engineers' peers will share expectations when there's an issue, and those issues are not systemic for us
- Collecting data isn't free – it's costs effort to collect, validate, keep clean, analyze, etc. If you don't expect to learn anything from it (or do anything different based on it), don't spend that cost!

> Oncall tooling: it should be invisible. Toil and confusion in oncall tooling is poison; the tools you don't notice are the ones that are working well!

- Tracking oncall shifts for compensation and reporting - do people have to do the work themselves?
- How intuitive is it? Measure peoples' honest mistakes- using the

- wrong way to page someone of the various available ways, tracking pages using right/wrong tools, number of pages going to slack or never ack’d

toolbox image from AI
[Images on slide: a classic “up and to the right” graph, and a wooden toolbox containing wrenches, magic wands, and a crystal ball]

Burning people out is bad too

It sucks a minimum amount to be oncall

- Minimal suck when actually getting paged
- Minimal impact on personal life

No one is oncall who doesn't need to be

Rotations are “fair”

[Laura] I also want to know that we’re not burning out our engineers with oncall work – the more we page people, the less time and energy they have to write new features, which is really what I get those fat bonus checks for (and as a manager I really do want to do right by my people). So I want to know that the demands we make of engineers to be oncall are the minimum reasonable demands, that the business really needs them, and that those are within the bounds of what humans can realistically sustain over time. Being oncall is a continuous cost to a team; we should know that we’re minimizing that cost.
If I’m thinking about preventing burnout… I also want to know that we’re not letting that burden fall on people in an unfair way – that when you ask a team “is your oncall rotation fair”, they’ll answer “yes”. People who feel that work is unjust burn out a lot faster :P

Photo from Ketut Subiyanto
https://www.pexels.com/photo/unrecognizable-person-sleeping-under-blanket-4546117/
[Image description: a person of color is mostly hidden under a sheet in a bed, with only eyebrows and hair visible]

Sustainability metrics for pagers

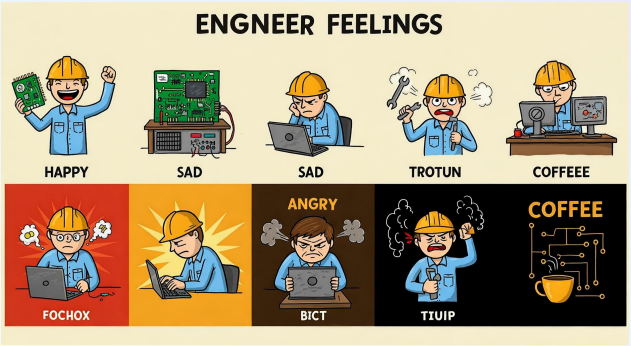
When and how are engineers getting paged?

Does it affect their lives?

Engineer sentiment: check in throughout the process!

Measure “fairness” in rotations

Not everything is numbers



[Jamie]
> It sucks a minimum amount to be oncall, in terms of both the actual getting-paged, and the impact on an individual's personal life

How would we measure how much oncall sucks, especially in terms of whether it’s going to burn people out?

- Being woken up sucks, how many overnight oncall shifts (raw and consecutive) does each person get? How many overnight pages did they get?
- Actually getting paged: How many pages do they get in the average shift?
- Impact on someone’s personal life: how many consecutive days are they oncall? How long are their shifts? What’s the total amount of time they’re oncall?

> Fair rotations and not burning out our engineers with oncall work

Not everything is numeric, especially how things *feel*, but sentiment can be quantified.

- Fairness can also be quantified in terms of the distribution of shifts within a rotation and the distribution of weekend/overnight shifts within that rotation

“engneer feelings” chart masterpiece AI generated (obviously)
[Image description: a chart of “engneer feelings” showing the moods happy, sad, sad, trotun, coffeee, fochox, angry, bict, tiuip, and coffee. Each “mood” is depicted by a small cartoon, most of which seem “angry” in some form]

What's "good incident management" mean?

Incident coordination, tooling, and process are

- Effective
- Invisible

Incident activities focus on mitigation

[Laura] Alright, I think we can work with that – I can bring that data to execs and point to lines getting "better" over time (let me just pause to have dollar signs in my eyes). So next up we have what you might even think of as the central team mission: after engineers get paged, sometimes they have a real incident. What makes that incident "good"? I mean, obviously no incident is "good", but what makes one "better" than another?
I actually saw this really neat talk at SREcon in Dublin on "Incident Groundhog Day", doing studies of incident response with AI (so obviously as a manager I love it) – and the takeaway there was that the best responders are the ones who are good at coordination and communication. And building on your point about the best oncall tooling being invisible – our incident process and tooling should be invisible, too. Can we measure that Jamie? Measure whether our engineers are communicating effectively, pulling in the people they need, and then maintaining coordination among those people?

Photo from Tiarra Sorte https://www.pexels.com/photo/firefighter-gear-on-red-truck-in-tultepec-31100671/
[Image description: old shoes and clothing stacked on a red step]

Measuring incidents

Tooling effectiveness and coverage

“Gap filling” automation

Sentiment and qualitative analysis

[Jamie]

We can measure that!

- Communicating effectively: adoption of the newest features/processes for incident management (e.g. incident workstreams)
- Pulling in people they need: measure how often specialized rotations are paged into high-sev/complex incidents (e.g. Core Incident Commanders, security lead)

For incident tooling, as long as you have one team staffed to build the paved path (like we do), that’s pretty straightforward.

- Coverage: are folks making their own incident automation to cover gaps?
- Invisibility: do people know the name of your service that provides automation? or are they surprised that it’s built by your company at all because it’s so straightforward?

There’s also a significant sentiment dimension to this too: even if your tooling and process are super effective, there may be other factors

frustrating people who are performing incident response. We all know that people who are angry are more likely to fill out a survey, so give your incident responders an opportunity to give you feedback in their own words and a process for collecting and acting on that feedback. You can also try to do fancy LLM sentiment analysis on the chat happening during the incident, although your success may vary since people may like to joke around in an incident.

Picture from Dakota Edwards https://www.pexels.com/photo/man-holding-axe-1459943/
[Image description: a man wearing a yellow jacket, red helmet, and red backback holding a large axe, shown from behind]

And are we srs bznss?

Incidents are fully managed through to resolution

- Regular stakeholder updates
- “These are important” vibes

Incidents only used for “real” incidents

[Laura] And I know you told me that folks treat their pager seriously… Do teams treat *incidents* with appropriate seriousness, as defined by me? Or do we see things like teams getting to the point where the problem is mitigated and then just walk away, or updates not making it out to stakeholders as often as they should, or maybe uses of the incident process for things that aren’t actually emergencies? Maybe teams using incidents because they can’t get their problem on some other team’s OKRs?

image: AI generated, via following the rabbit hole down from “noodle incident”
[Image description: a ferret on its back, on and covered in a pile of spaghetti, looking joyful]

We can measure that!

Teams moving their incident through the whole state machine

Metadata gets recorded

Teams *choose* to write postmortems

Incidents with few teams responding

[Jamie]
Yeah we can measure incident srssness! In the process, we can look at incidents as a state machine which begins with the initial state of impact beginning and the terminal state of an incident being fully followed-up on, all action items closed etc. Given that mindset, we can think about measuring where incidents linger.

- Incidents do not (typically) linger in "mitigated" state without resolution, when they do, they remain active (e.g. responders giving regular status in slack)

When you get to the later stages of the state machine, consider measuring the completeness of the data you need to reflect on incidents as an organization.

- Tracked metadata is recorded (and more consistently for more severe incidents).
- Teams choose to write postmortems even when not required / even for lower severity incidents

Teams filling out the metadata is a signal that teams think incidents are important

In terms of using incidents for things which don't require that kind of response, there's a couple things to quantify and plenty to qualify.

- Watch for (and stop) "robots create incidents" team workflows - incidents are declared by humans for a reason!
- Follow up on incidents with "no incident commander" - someone was in charge or everyone was confused.

However, there's often a significant organizational fault line behind that pattern of making incidents- like you were saying about "getting their problem on some other team's OKRs". This can get fuzzy and difficult to quantify because those organizational patterns are different for each group of people experiencing them, but doing reviews of your postmortems can help to point out some of those patterns.

One interesting measure we've been piloting to check on the "appropriateness" of incidents being filed is measuring the percentage of incidents that involve very few or very many teams. A reduction in incidents with few teams may indicate a hesitance to file incidents for less severe/more local problems. This is still experimental and a bit fuzzy, but tends to get at "are we good at incidents" on a more local scale than looking at the state machine holistically.

Image by Pat Whelen on pexels:
https://www.pexels.com/photo/photograph-of-people-riding-an-amusement-park-ride-7118579/
[Image description: a distance shot of part of a rotating carnival ride]

Are we… good at them?

Can’t measure time

Remember, it’s bad statistics and bad incentives

Do we *feel* competent and prepared?

Focus on the problem

Not scared about attention

[Laura]
And I do want us to be *good* at incidents – able to solve customer problems as quickly and competently as possible. Since you’ve told me that measuring time doesn’t tell us that no matter how much I wish it so, I guess we need a proxy that does work. <thinking face> Well… I guess I could probably sell the assumption that “engineers feel prepared and good about their response” is a good proxy for “we solve our incidents well”. That’s an assumption, but it’s one I’d be comfortable standing up for in front of higher management. So… for those engineers responding to incidents, *do* they feel comfortable, prepared, and ready to respond? After the fact, how is their self-critique? Do they mention having a hard time following or focusing on the technical challenges, either for skill/preparedness reasons or because they’re worried about the fact that things are down/they might be blamed?

[Images on slide: a thinking emoji, an “I have an idea” emoji]

Measure those vibes

Usage of roles in your process

Including incidents with no one in charge

Sentiment and qualitative analysis

[Jamie]
> Comfortable, prepared, and ready to respond?

- Engineer sentiment: Qualitative measurements before and after training, and after first N incidents, through surveys. Direct LLM sentiment analysis on conversation during an incident
- Usage of roles in your process: you should have a way to measure this (for instance, our own incident app lets you do that :))
  - Are there incidents with no incident commander in the driver seat?
- Read postmortems for red flags of confusion
- Again this is fuzzy - do qualitative sentiment analysis on the incident chat and postmortem

After the fact, how is their self-critique? Do they mention having a hard time following or focusing on the technical challenges, either for skill/preparedness reasons or because they're worried about

the fact that things are down/they might be blamed?

- You touched on a great point saying that responders may have a hard time focusing on the technical challenges when they're afraid they might be blamed. Blamelessness starts at the policy level but really requires constant checks throughout the organization to ensure there's not a pocket of blameful practices under one leader.
- Staff up a team who can keep the pulse of incidents, either directly or by conversation with people involved in them. If something doesn't smell right, follow up on it. <click> In the infinite wisdom of PECO, smell us tell us.

[Image description: Jim Carrey excitedly and very quickly answering emails, not pausing for a sip of coffee]

Communication builds trust

We communicate quickly and effectively with customers

In language that makes sense to them

[Laura]
How's our communication with customers? Usually having an incident does less harm to your reputation and customer trust than having one and *not* communicating about it well. So do we inform customers about problems in a timely way, in language that makes sense to them?

trust fall failure image generated by AI
[Image description: a man and woman executing a trust fall, except both arms are thrown akimbo and the faller is not being caught]

An easy one!

Timeliness of customer communications

Measure customer feedback

[Jamie]
This one is actually really easy to measure directly! All your comms with customers are happening in known public or private places, after all.

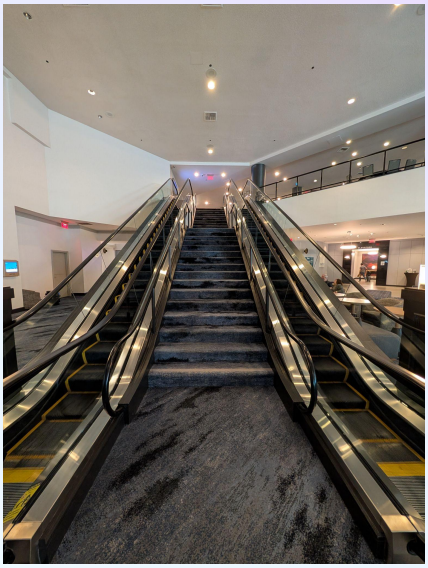
- Time to post communications for high-sev incidents
- Customer feedback about what's been posted
- Qualitative feedback from dedicated customer comms team, if your company has one

Photo by Yaroslav Shuraev on Pexels:
https://www.pexels.com/photo/megaphone-with-flowers-in-hand-7697265/
[Image description: a hand holding a megaphone which has been stuffed with flowers]

Does escalation work?

Our central volunteer IC rotation remains well staffed

- Prepared for our most severe and complex incidents
- Viewed as reliable and knowledgeable



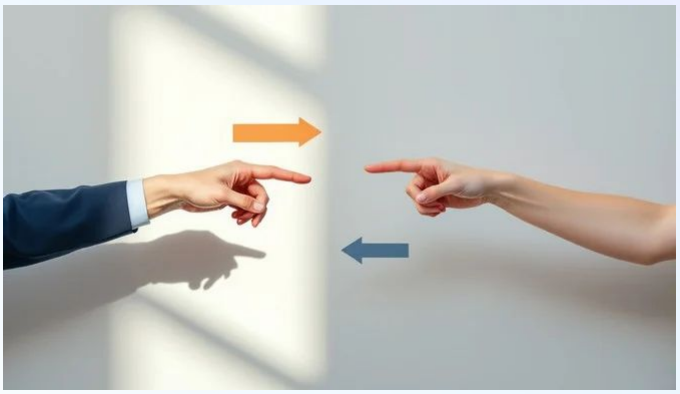
[Laura]
Okay last one on our actual in-incident response – at Datadog we have a volunteer “escalation” rotation for our more severe incidents, to help coordinate them and drive resolution. Is that rotation well staffed? Or is it struggling to find enough volunteers to keep up with attrition? Do the people on *that* rotation feel ready to respond, comfortable with our tooling and tech, and able to step into that incident command role? Conversely, does that rotation have trust from the rest of the company – when they show up to help, is that perceived as a good thing or a bad one?

image credits: Photograph by Jamie Luck on Monday. May look familiar.
[Image description: the escalator in the convention hotel lobby, seen from below]

Moar metrics!

Tenure and team size for critical rotations

Direct feedback and perception of others



[Jamie]

- Count of number of members of the rotation and average tenure, with ~12-18 months "sweet spot"
- Consistency of handoffs within this rotation
- Direct engineer feedback on value of coordination rotation (since it’s a small size this can be manual)
- Perception of escalation positioned as experts is very important; it’s harder to build trust during an emergency. You can reinforce trust that’s already there but when engineers are stressed out it’s not a great time to build those bridges.

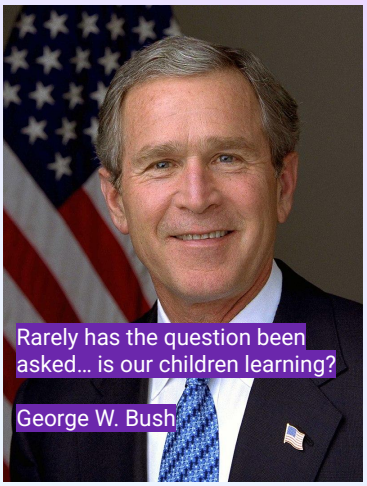
Photo by Stockcake
[Image description: two arms pointing fingers at each other, one on each side of the image]

Is our engineers learning?

Analysis of our incidents is adequate:

- Prompt but complete
- Finds real fixes

We do the fixes identified



[Laura]
How about after our incidents? If I think about an incident as an investment where we involuntarily pay most of the cost up front, I want to know we're getting the maximum return from that investment. Do we properly analyze what happened, and fix the things we find? Do we do that quickly enough to prevent repeats? That's going to mean we have real engineering investment in doing that analysis (not just ticking a box), and that we see progress in our actions instead of churn (like an action to upscale in one incident, then downscale in the next one)!

[image: George W Bush official photo]

How do we measure learning

Postmortem length and completeness

Review and reading groups

Track action completeness

But no “target” – allow for deep fixes

Track repeat incidents

DATADOG 34

[Jamie]

- How do we measure that? Well hopefully it doesn’t require writing structural engineering equations on an orange but hey what works for your company right?

- Analysis of our incidents is adequate, prompt but complete, and embraces complexity and full understanding of our systems
  - Postmortem completeness and length – and make sure this analysis is in your career ladder!
  - Postmortems are written for all severe incidents
  - Postmortems are consistently started quickly even if not completed (we have a standard internally on starting postmortems very soon)
  - Groups of teams have use an in-team postmortem review process and report that this is valuable.
- Real engineering investment in analysis and fixes, with low churn
  - We do not measure number of actions on a postmortem, but we do track how quickly actions are finished. There’s no

- static target there because some fixes are very deep within our systems.
- We look for teams to define both "short" and "long" term follow up actions.
- Monitor for repeat incidents and track contributing factors over time
  - Repeat incidents ~has to be manual – engineers looking for “patterns I’ve seen before”
  - LLMs are not good at this

Image credits: cottonbro studios on pexels:
https://www.pexels.com/photo/person-holding-orange-and-white-round-ornament-4778677/
[Image description: a hand writing equations on an orange ball with a sharpie]

A wild learning appeared!

Real fixes

more complex incidents

[Laura] Alright, we're getting there. I can see how I can use those metrics to tell our execs a story that our incident management is good and effective. But I do want to be measuring that the things we build for customers are getting better as we work on them, and we're not just running in place. You know… based on what you've been telling me Jamie, it seems like if we're doing it *right*, MTTR should go *up* over time because our incidents get more complex and more things have to fail, in new and surprising ways, to have an incident at all. I know, I know, we can't actually measure a "mean time"... but can we measure incident complexity?

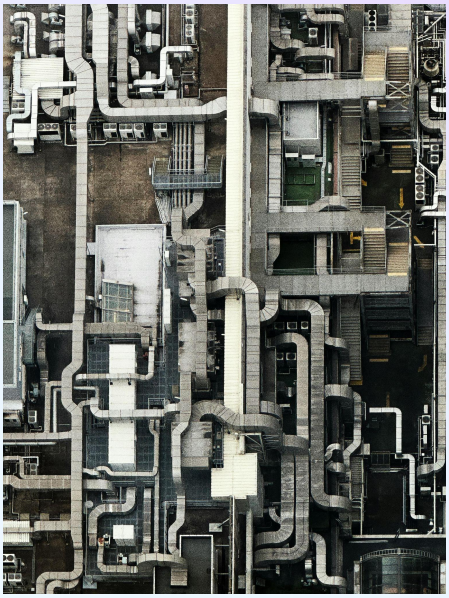
Light Bulb or Idea Flat Icon Vector.svg from Wikimedia Commons by Videoplasty.com, CC-BY-SA 4.0
[Image description: an original pokemon screen capture with a "Wild LEARNING appeared!" message and a light bulb as the "pokemon" appearing]

Complexity indicators

Length of postmortems

Number of people and roles

LLM sentiment analysis



[Jamie]

- Yeah – we've covered some of this already, but there are definitely some metrics we can use to detect incident complexity
  - Length of postmortem
  - Number of engineers and which roles were involved in an incident
  - Incidents requiring more escalation members
  - Sentiment analysis- LLMs can do okay here!

ductwork image by 도연 김 on pexels:
https://www.pexels.com/photo/aerial-view-of-complex-industrial-pipes-31230742/
string image via pxhere: https://pxhere.com/en/photo/854856
[Image descriptions: a colorful tangle of string; a complex pipe infrastructure]

Execs as customers: are they happy?

The stakeholder we need to please is execs

Can they see the reliability in their org

[Laura] It's already implied by the questions I'm asking here, but to a significant degree the customer for these metrics we're building are our internal execs. So… do they have the info they need from us? Presumably that's visibility into the the reliability state of our systems and places where problems are emerging, so they can shift investment if they need to. Are we giving that to them?
Executives have the visibility they need into the reliability state of our systems and where problem areas are emerging, to allow for shifting investment in reliability

photo credit: Tima Miroshnichenko on pexels:
https://www.pexels.com/photo/businessman-man-woman-laptop-6694918/
[Image description: a standing white man, sitting woman of color, and standing South Asian man in business attire holding tablets and clipboards gaze sternly toward the camera]

Execs are customers too

Are summaries being read?

Leverage existing feedback processes

[Jamie]
Executives have the visibility they need into the reliability state of our systems and where problem areas are emerging, to allow for shifting investment in reliability

- Publish summaries of the monthly incidents and check view counts for those summaries
- Feedback processes:
  - Quarterly reviews
  - Qualitative feedback from execs in incident meetings

Photo by RDNE on Pexels: https://www.pexels.com/photo/people-talking-to-each-other-7580751/
[Image description: a group of people in an office environment engaged in conversation]

And the success North Star: Customers!

Customers are customers: If they stop paying, we stop working!

[Laura] All right – so that gives us *lots* of metrics to show to executives to tell them that our incident management process is successful and working well. But what am I going to tell any execs who notice that that's not the same thing as measuring customer reliability? Like, our satisfaction with our incident management process is a proxy for "we fix customer problems effectively", but ceci n'est pas une pipe (yes my French is terrible). We've covered a lot of ways to measure if the *incident process* is working, based on our belief that the process lets us recover well when things inevitably go wrong, but that's not the same as "is the product reliable for customers".

[Image description: "Ceci n'est pas une pipe" by Magritte, with "pipe" overwritten by "Customer Experience"]

How do you measure customer success?

… say it with me

[Jamie] Say it with me everyone! SLOs. see other talks in this conference, yo: if you want to measure reliability for customers, *do that directly*, don't use incidents (your own internal process for handling outliers) as a proxy for the customer experience.

Is it working?

Feedback on measured oncall metrics

Resistance to some measures

Keep the conversation open

Visibility into trends we need to correct

Metadata isn't a substitute for analysis

[drop kayfabe, Jamie] Okay, so we've actually done this – we had this conversation, as a team and with our management, about both the need to measure success and what we can and think we should measure. How's it been going?

- Feedback on metrics: get this early and often.
  - Once you've defined success metrics for a team it's harder to move and change, so make sure you have strong alignment before making changes official.
  - Do you have a structured feedback process? Maybe you should, it may be helpful

[Laura]

- Resistance to some measures
  - Start by trying to get initial alignment on values and metrics with anyone who has decision-making power
  - Some people will still resist: it's helpful to have an understanding conversation with them about what they're trying to do, and why the implementation of it doesn't meet their goals
  - There's no simple solution here; it's an ongoing conversation – and you can keep the conversation open

    - from your end even in cases where you are measuring the “wrong” things right now. Think marathon and broad education, not a sprint and “winning the argument”

[Jamie]

- Visibility into trends that need to be corrected
    - Tricky. We haven’t figured this out fully yet beyond regularly checking the dashboards and getting a sense for what “normal” looks like
    - Building a mental model is valuable before taking action on any metrics

[Laura]

    - In reality a lot of our ongoing tracking/awareness is done by “keeping our hands in” and doing periodic check-ins as a team for “emerging concerning patterns” that we may need to follow up on – if we do that, and stop seeing the pattern, no further action needed
    - It’s important to remember that metadata and summary statistics, especially if you’re collecting it cheaply (by, for instance, asking every team to fill in a bunch of checkboxes) is not a substitute for analysis – digging in and understanding your actual patterns and problems

[Image description: an upside down cat lying on a cable, captioned “I plugs it in… but it still no work”]

Summary of things we're measuring

Mistakes and time spent using paging, oncall scheduling, and comp tools

Pages per person, especially overnight

Length and frequency of oncall shifts

Perceived fairness of oncall schedules

Adoption of incident tooling or creation of team-level support tools

Incidents left open

Incidents with no incident commander

Postmortem completion, voluntary PMs

Team counts in incidents

Use of full spectrum of incident roles

Time to customer messaging

Customer messaging feedback

Escalation rotation size and turnover

Sentiment analysis on postmortems, incident channels, etc.

Postmortem length and complexity

Postmortem reading groups

Action item tracking *without* "deadlines"

Repeat incidents

Number of people & roles in incidents

Executive attention

Executive satisfaction

SLOs! (as a whole distinct project)

[Jamie] so, we've covered a lot. While your org *probably shouldn't* be measuring exactly the same things that we are since it's undoubtedly different, here's a summary of some of the things we keep tabs on internally to measure the success of our incident management process. There's of course a whole RFC behind this slide with waaaaay more detail than we could possibly fit in 35 minutes :)

Messy desk background generated by AI

Takeaways

"Do we manage incidents well" *can* be measured directly

Define what "good" looks like

Incidents are unique and complex event

Extreme loss of fidelity for any metrics that are "simple"

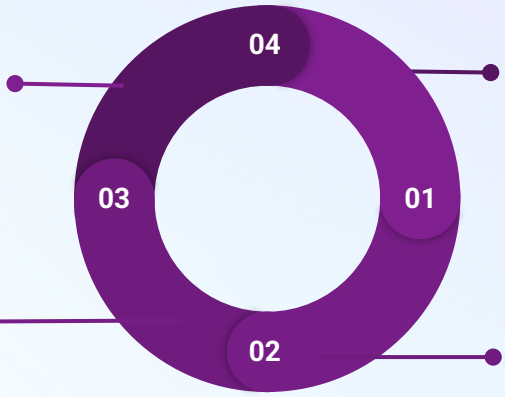
Working with stakeholders is key to create the right incentives

DATADOG 34

[Laura] But please don't take away that list as "this is the answers". Like most things engineering, the right thing to build is going to depend on your specific needs and environment. But we want to emphasize that it makes sense to think of your incidents as an *internal* coordination process, and to measure directly whether that coordination is succeeding. Measuring whether we handle incidents "well" involves both defining what "well" means, and understanding that incidents are unique and complex events -- so simple averages of "good" things won't help. Working with stakeholders, especially executives, to build a shared vision around both the complexity and goals of an incident management process is key to building metrics that really make things better, instead of generating shallow optimization behaviors and meaningless work across your organization.

Takeaway image by Jakub Kapusnak via
https://www.rawpixel.com/image/447666/free-photo-image-takeaway-foam-box-fries
[Image description: fried food in a styrofoam box on a picnic table]

Executive-friendly flow chart!



Define and Expand
Define corrective actions for existing metrics and a feedback process to identify new metrics

Alignment
Stakeholders agree that goal-based metrics are worthwhile

Identifying Metrics
Try to get multiple metrics to support each goal, from different perspectives.

Identifying Goals
What's important to your organization? How do you want to create incentives?

[Jamie]
So if you, like Laura's character, are looking for something to give your executives to show them that we are doing the right things and continuously improving, here's an exec-friendly flowchart of how to build measurements for your own internal incident process. Keep in mind it's a cycle because you should be constantly re-evaluating your posture here.

Questions?

DORA metrics

If your execs have nothing…

Do you routinely optimize to all your compliance frameworks?

Compliance frameworks can also be a proxy for “maturity”

[Laura] I’m so glad you asked!
We’ve focused in this talk on what we find useful for engineering management and really understanding our incident management process so we can improve it. If a team doesn’t have any executive-level visibility already on offer, then DORA metrics are going to fill that gap, whether they’re any good or not.
One thing to remember is that organizations are typically subject to a *lot* of compliance frameworks – just because the organization is measuring something and reporting it externally *doesn’t mean you have to use those numbers as an optimization target internally* (or even report them internally!)
One thing a colleague told me about compliance frameworks in general – they’re not always about the specific metrics you’re reporting. Sometimes, they’re just a measure of whether your organization is mature enough to *have* those metrics, and have them be within certain reasonable parameters.
Also – I learned actually just this week that the latest version of DORA *doesn’t* include “MTTR” – instead, the measurement is something like “time to remediate a bad rollout” – that is “how long does it take us to roll back”, which certainly *is* something you probably want to do some level

of optimization for!

So: my stance at least is that DORA metrics are interesting to external parties because that's how bureaucracy works, and that internally we should not be trying to report or optimize them as any kind of health measurement – and that giving engineering leadership data you can and do get behind instead is how to succeed there.

Elephant in wrapping paper generated by AI
[Image description: an elephant wrapped in colorful paper]