EBOOK

3 ways to improve your incident management program in 2023

Regardless of where you are on your incident management maturity journey, there's a right next step you can take. In this ebook, we'll discuss three areas of focus and the foundational, intermediate, and advanced steps you can take to improve your program in each.

Introduction

Incidents are costly. You have hard financial costs, like lost sales and SLA breach paybacks, to start, but when you mismanage those incidents, the costs really start piling up. We're talking opportunity costs like lost productivity among engineers, and cultural costs like a degraded reputation among employees and customers. Poor reliability, perceived or not, is never a good look — but in the post-pandemic Wild West of the tech world, it can be tragic.

> **47% of companies** that experience downtime say they contribute to a loss in productivity
>
> The Real Costs Of Planned And Unplanned Downtime, Forrester, 2019

The flipside though is that when you manage your incidents well, you have the opportunity to not only respond faster, but also to make real improvements in reliability. To do that though, you need to have two things — a streamlined and documented response process and the commitment to learning from your incidents. Of course, that's all easier said than done, but where do you start, and how?

When in doubt, look to the numbers. We analyzed data from more than 53,000 incidents declared and resolved using the FireHydrant platform between 2019 and 2022 for our Incident Benchmark Report. This is the largest analysis of incidents from date, and from it, we were able to identify some activities that made a positive reduction in mean time to resolve. This is by no means a perfect measurement (what is?), but it's a place to start.

We identified three key figures through that analysis that can help you define goals for your 2023 program:

1. **Assigning roles during an incident decreases MTTR by 42%.**
2. **Using a service catalog during an incident decreases MTTR by 36%.**
3. **The average number of retros per month by company increased 236% between 2021 and 2022.**

Regardless of where you are on your incident management maturity journey, there's a right next step you can take. If you're just starting with formalized incident response, focus on assigning primary roles, building processes, and kickstarting learning from incidents.

If you've already established a consistent incident management program, now might be the time to start thinking about more robust roles, building out your service catalog's utility, and right-sizing your retro process to be more consistent.

And even if you already have a robust and reliable practice, there's always room for fine-tuning your roles, processes, and retrospectives. Use metrics to drill down on areas of your

practice that need the most attention and to show maximum ROI.

In this book, we'll present ideas for how to take the next formative step on roles, service catalogs, and retros, no matter where you are today. Let's get started!

Chapter 1: Incident roles

The fastest way to spring into action is by having a well-defined team in place ahead of time. But it's not a matter of just throwing all hands on deck for every incident.

We found in the Incident Benchmark Report that the incidents with the lowest MTTR had a magic number of responders — 6. MTTR increased by 18% when even one more responder was added. Likewise though, it's not just having the right number of responders, it's making sure they clearly understand their roles during the incident that makes the difference.

> **42% lower MTTR** when roles are assigned

Like everything we'll talk about in this book, there are incremental steps you can take to mature how you handle roles. In this chapter, we'll talk about three approaches, each based on maturity level.

For the foundational team

If you're just getting started with a formal incident management program, or don't have many dedicated resources to put toward incident response, start by building a foundational team that includes the most critical roles for executing an incident response.

Incident manager

Sometimes called the incident lead, the incident manager owns the entire incident response process. The incident manager is often a team lead or someone with a decent amount of institutional knowledge. At many companies, the person who declared the incident adopts this role by default, but the incident manager could be anyone on the team with sufficient knowledge to manage the task.

Incident manager primary duties may include:

- Assemble the responding team, assign tasks, and track progress
- Provide direction to mitigate the incident at hand
- Communicate updates at large and work with others to make sure that the right stakeholders are brought into the loop
- Determine how to handle anything else that comes up during the incident

Incident responders

This is an umbrella term for every responder who contributes to mitigating incidents. Often, the responding team is composed of members of whatever team the affected service sits under. However, some companies default to senior or experienced team members or a centralized group of responders that are deployed during an incident, often as part of an on-call rotation, to ensure someone is always available.

Response teams can be tailored to the type of incident. For instance, if the incident is purely technical or declared regarding an issue in the staging environment, the responders would likely only include engineers and product managers. More significant incidents could also have customer-facing roles and other team members who have context around how the incident impacts specific customers or features (we'll discuss some of those roles later on).

Incident responder primary duties may include:

- Work on mitigating and resolving the incident (as opposed to organizing the team, like the incident manager does)
- Communicate what they are doing and why to the incident manager and other members of the responding team
- Complete other duties as assigned by their incident manager

Internal stakeholders

Stakeholder is a general term for people invested in the outcome of an incident who need to be kept informed. Your internal stakeholders will vary based on your company, the severity of the incident, and how much it impacts customers. However, common internal stakeholders include leadership team members, product managers, customer-facing teams, and engineering leaders.

The incident manager should facilitate communication with these stakeholders or appoint someone to own those communications. As appealing as it might be to focus solely on mitigation and skip the updates, don't. Good communication builds trust and buys grace during an incident.

Internal stakeholder primary duties may include:

- Receive updates about the response efforts on a timely basis
- Provide resources or assistance as requested
- Provide relevant information to external stakeholders, like customers, in some cases

For the growing team

As your process matures, you'll gain a better understanding of your organization's specific needs for responding to incidents. That clarity will help you tailor your response roles based on the incident at hand versus taking a one-size-fits-all approach. Here are some roles you might consider adding as your approach to incident management matures.

Incident commander

The incident commander is a more robust version of the incident manager with more expansive duties. An incident commander is accountable for the entire process and duration of the incident. In contrast, an incident manager is primarily focused on mitigating the incident.

Incident commanders can either be volunteers from within the organization or they can be appointed to their role. Depending on your organization's process, this could be a domain expert or a general project/process manager. Regardless of how you choose, this is typically an ongoing additional responsibility that high-performers take on to increase their exposure to stakeholders with the hopes of going into leadership.

Incident commander primary duties may include:

- Get the right people in the room together and appoint other roles as needed
- Remove blockers to mitigation and help the team work toward resolving the incident
- Organize communication with stakeholders
- Minimize risk across the company
- Support incident resolution tasks, including documentation, post-incident review, and working with others accountable for process improvements

Incident management lead

While the incident commander owns real-time incident response, the incident management lead is responsible for preparing the organization for incidents ahead of time. They may designate roles and train individuals in those roles, set incident milestones, and document all processes.

The incident management lead might be a domain owner, like someone in engineering or product leadership. Or if your company has a designated risk mitigation team, platform engineer team, or site reliability engineering (SRE) team, a representative may act as incident management lead.

Incident commander primary duties may include:

- Work across departments to ensure all relevant people receive training on the incident management and response process, roles and responsibilities, and expectations
- Make sure SLAs are understood and communicated to ensure the team knows how to best respond to an incident and/or any process updates
- Coordinate training and gamedays to run through practice incidents

Customer success lead

The customer success lead is a liaison between customer-facing teams and the incident response team roles. They act as a shield to keep the response team focused and a resource to provide information to customer-facing stakeholders.

Typically, this person is a customer success team member with a good working relationship with engineering. In some cases, it might also be a member of the engineering team.

Incident commander primary duties may include:

- Advise customer-facing teams on the situation and timeline for resolution
- Pass relevant details from customers and customer-facing teams back to the response team
- Handle communications with high-profile or heavily impacted customers

For the robust team

An experienced and well-provisioned incident response program may have room for even more specialized and advanced team roles that may only be needed for some incidents but can be particularly useful for high-profile or public incidents. But beware of scope bloat; build your response team to include only the most relevant people to the task.

Communication lead

The communications lead disseminates information to internal and external stakeholders. This role becomes increasingly essential with major incidents that require public communication.

The communication lead is typically someone with extensive experience in written communications, possibly from customer support or, if the company has one, the public relations department.

In some organizations, the role of the communications lead role may be split into two if a large incident requiring complex communication occurs: an internal communication lead and a public company spokesperson.

Communication lead primary duties may include:

- Post regular updates (externally and internally) about the incident
- Create public-facing communications about the incident in conjunction with customer success, legal, etc.
- Liaise with stakeholders and facilitate flow of information in both directions

Investigative lead

The investigative lead collects and analyzes information from past incidents and makes recommendations to help prevent future ones. In some companies, they're also called the problem manager or root cause analysis manager.

Your investigative lead should have a firm understanding of operations related to the incidents at hand so they can ask the right questions and make knowledgeable recommendations. Sometimes the incident commander doubles in this role.

Communication lead primary duties may include:

- Manage risk
- Collect and analyze data to find the root cause of the incident
- Make recommendations for future incident prevention

Legal advisor

Occasionally, incidents affect sensitive information that could lead to potential legal repercussions for the company. The legal advisor is a legal expert who can advise on responsibilities or liabilities related to an incident.

Usually, this is someone on the company's legal team who has at least a basic understanding of the technical aspects of your software.

Legal advisor primary duties may include:

- Ensure legal compliance during incidents and surface any violations to the legal team
- De-escalate events to prevent contractual or legal breaches

- Work closely with the communications lead to ensure a proper amount of information is shared at the right time with the public, especially during a large incident

Chapter 2: Service catalogs

Efficient incident response isn't just about getting the right people in the room; it's about making sure everyone has the same knowledge base to work from. Service catalogs help you do both.

At its most basic, a service catalog is a detailed list of technical services (enterprise applications, task-specific tools, microservices, and so on) used by your organization, both internally and externally.

A service catalog helps knock down knowledge silos and ensure everyone has the information they need to move forward confidently — a big deal when you've just been paged at 1 a.m. When teams use a service catalog, they can more quickly identify root problems and bring in the subject matter expert or owner of the affected service during an incident.

> **36% decrease in MTTR** when services are attached

Like everything we'll talk about in this book, there are incremental steps you can take to implement and then mature your approach to service-based incident response. In this chapter, we'll talk about three approaches, each based on maturity level.

Get started

At its most basic, a service catalog is simply a list of internal and external technical services (enterprise applications, task-specific tools, microservices, APIs, and so on) used by your organization, and relevant details like owner, code location, and operational dashboards. By documenting this information, you help knock down knowledge silos and ensure everyone has the information they need to respond to incidents confidently — a big deal when you've just been paged at 1 a.m.

Even if you haven't written anything down yet, you probably have a framework of service dependencies living in the heads of your team. Moving this service graph into your service catalog will further help you determine who should be in the room when things go down.

If you're building a service catalog from scratch, though, start simple.

Start by listing all the services with their owning and responding teams, contact details, repositories, documentation, and monitoring dashboards. If you're managing a monolith instead of microservices, you can still use a service catalog. Break down any monoliths by module, components, or product surface area. Each product area should have an engineering team associated with it, and those teams should be trained on your incident response process.

Once you've got the simple service details and dependencies out of everyone's head, you can start to add valuable layers to your catalog. Add functionalities, like login or checkout, on top of the services that power them. Why by functionality? Because that's how your customers think. They're not concerned with what service is broken, they're concerned that they can't log in. This has the added benefit that more people in your organization can be involved in incidents without knowing the technical details of your system.

You don't necessarily need a dedicated tool if you're just starting out — you can store your service catalog in a company wiki or a shared document that you can reference. However, if you're new to documenting a formal incident response process altogether, taking the time at the beginning to center it around your services will deliver better results when you start to mature your program. So consider setting up your process so that when an incident is declared and you find out what's broken, it's clear which team member needs to be alerted (and maybe that's even done automatically).

Level up

When they're most valuable, service catalogs are treated as more than a directory; they are valuable tools for aggregating institutional knowledge. The ultimate goal is a fully fleshed-out service catalog that includes dependencies, owners, and links to operational documentation. And once you have the basics in place, there are several steps you can take to level up.

- **Create space for documentation**: Users should be able to quickly find out what a service does, who created it, what recently changed, and all other information associated with it. Dedicate a space in your service catalog for documentation of past incidents and events.
- **Map services to runbooks**: Seriously speed things up by connecting your services to runbooks. For example, Avalara mapped a runbook to each service the company monitors, which helps the team get the right people in the right place at the right time faster, as well as document service-specific nuances and processes for the many applications monitored. When a service goes down, the corresponding runbook is triggered, and everyone jumps into action.
- **Consolidate information**: Update your service catalog as your organization grows (there are tools that make this easier) to ensure a streamlined, focused incident response process. Information consolidation ensures that new hires and unfamiliar teams can access the same valuable historical data and context as your most experienced engineers.

Make the most

Cataloging and tracking changes to all of the services within your system can become increasingly complex as more users and processes are added. Tools like ours or Backstage (which integrates with ours) can make maintaining your service catalog significantly easier. And ultimately, having all of that information interacting with your incident response process is where the big payoff on time savings happens.

Using a dedicated tool can also help you improve your incident management process by:

Automating incident kickoff
When an incident is declared, your tool can use the service catalog to automatically pull in all relevant parties into a shared Slack channel.

Automating record keeping
Automatically add incident reports and historical data to the service catalog, so it's ready for the next time it's needed.

Creating production readiness checklists
Evaluate and maintain the production readiness of the services your users rely on every day: spot risks in your service dependencies before they cause incidents, and respond quickly if they do.

Generating user service dependency graphs
This is a visual way to quickly surface dependencies, understand the relationship between services, and determine the scope or impact of an incident.

A robust service catalog is an essential tool in the overall incident management ecosystem and can significantly enhance your team's productivity. It's not surprising that our Benchmark report found a 1640% increase in services created throughout 2022.

Chapter 3: Retrospectives

The retro is an essential step in the incident response process usually taken after resolution with the goal of giving responders a space to process their experience, understand the incident's causes, and improve the response effort itself, as well as the software we build.

Despite their usefulness though, we've found that not all teams hold retros after every incident, perhaps thinking they're not worth the time or effort, especially on lower-severity incidents. In fact, in our Incident Benchmark Report, an analysis of 50,000 incidents resolved on the FireHydrant platform, we found that on average, retros are performed after about 29% of low-severity incidents and 42% of high-severity incidents. The interest is growing though — we also saw a 236% increase in the monthly average number of retros per company, between 2021 and 2022.

236% year-over-year increase in monthly average of retros per company in 2022

So what if we just made retros easier to complete? If your team skips retros, reframe your thinking and consider right-sizing them so the retro effort level is commensurate with the severity of the incident. Ditch the one-size-fits-all process to ensure that this important step is held at the end of every incident. Here's how to make it happen.

Get started

So where do you start? Ideally, a retro should include all key responders. Start small and grow as needed. Commit to doing a 20-minute retro with an SRE retro format for a 3-month time commitment and see how it goes for any SEV1 or SEV2 incident.

During your retro, reflect on a preset set of questions. These might include:

- What factors contributed to this incident?
- What lessons did we learn?
- What did we get right? What did we get wrong?
- What do we need to improve?(this question combined with the first question are drivers for follow up work).

Another popular tactic is the five whys interrogation technique. In this exercise, you ask why something happened, then ask why that happened, and so on, five levels deep. This will help you determine the root cause of a problem.

Whatever format you ultimately adopt, the goal is to develop and document recommendations for maturing your incident response process and improving reliability of your systems. And to safely discuss issues related to those goals, it's important to remember that retros are not for assigning blame to any one person or team. Blame is counterproductive to learning because it leads to stress and defensiveness.

Right-size retros

So you know you need to have retros to invest in the resilience of your process, people, and products, but how do you walk the line between enough and too much? Retros can get costly — and maybe that's okay for SEV1 incidents. In those situations, you might want to have a wide swath of your engineering team and leaders in the room (especially if you're having high-severity incidents frequently). But when you're taking high-cost employees away from other high-value tasks, you want to make sure it's worth it.

Think about minimizing costs and the demands on people's time by optimizing your retro process to better fit your team's needs.

Have them when customers are impacted

If your customers felt pain, you should seek to understand the cause and impact.

Institute asynchronous retros

Some retros — like ones for severe incidents — might necessitate a big sit-down meeting, but on lower-severity incidents, maybe you can get the same input through a shared document. In a world where "it could've been an email," think about whether or not a live conversation is necessary to get the input of your team.

Don't bloat the invite list

The people who responded to the incident are the most important people in the retro — consider everyone else optional. Track who participated in an incident (or let a tool like FireHydrant do it for you) and use this information to build your retro invite list.

Don't cancel the retro if someone can't attend

Hard to get everyone's schedules aligned for a retro? Work around them. If a critical member of the team can't make it, ask them to share their input directly with you or async in the retro report doc. You can deliver their answers to the rest of the group.

Share your findings

Finally, make sure you share the retrospective report after the retro has occurred. Share the report in your incident response channel, and share it with the wider company as well. That

way, everyone — including those who may have wanted to attend but could not make it — has access to the incident summary. This is a great way to help senior stakeholders stay in the loop without including them in every retro.

Make the most of retros

The goal of holding retros is to ensure your team captures the important takeaways that come out of every incident — not checking a box just to say you had a meeting. Retros are a designated time to allow responders to decompress, share lessons learned, and ultimately, create action items to optimize your incident response plans and system health.

Once you have your retro structure and cadence in place, start measuring data around incidents. You can't determine how to improve if you don't know your starting point, right? And retros are all about improvement. Measurement gives your team a benchmark for how well they are performing and can highlight areas of improvement and growth.

Again, this is an area where you can start small and mature over time. Even if you're not ready to collect extensive analytics, measuring anything can be useful. For example, for our Incident Benchmark Report, we chose mean time to resolve (MTTR), which measures the length of time between when the incident was declared and when it was resolved, as our starting point metric.

Another one to consider is mean time to respond (also MTTR 😅), the length of time it takes from when the problem is identified to when the mitigation effort begins. This number can tell you a lot about whether you're burning time and money on rote tasks and toil. And, since it's an area that's ripe for automation, it's a great place to start eliminating costs.

Whatever metrics you choose to focus on, have your team ask themselves, how can we improve this metric? Discuss frequently, particularly during retros, and track progress over time. This focus will ensure that your team is constantly demonstrating growth and improvement.

You can also consider opening up the learnings to folks outside the engineering organization. Retros give your teammates an opportunity to see how your team approaches incidents, which can be a learning experience for everyone. In fact, Snyk holds a monthly meeting of their Incident Response Guild where they review incidents; the meeting often brings upwards of 100 attendees.

Take the next right step

We've covered a few different ways you can improve your incident management program, but ultimately improvement depends on the priorities of your program and organization. The most important thing is to start somewhere: whether your focus is on having fewer incidents overall, improving a specific functionality where you see a lot of incidents, or improving external communication. Start making incremental change, and just as importantly, start measuring that change so that you can showcase your progress.

And no matter where you are in the process, it helps to have a community of peers to bounce ideas off and learn from. The Better Incidents Slack group, sponsored by FireHydrant, is just that place. Join now to learn and share more ideas on better managing, responding to, and learning from your incidents.

FireHydrant

From ring to retro

Start for free

Developers

- Docs
- Quickstart
- Integrations
- Changelog
- System Status

Platform

- 4 Minute Demo
- Alerting & On-call
- Service Catalog
- Runbooks
- Incident Retrospectives
- Incident Analytics
- Pricing

Community

- Better Incidents
- Customer Stories
- Blog

Company

- About Us
- Careers
- Security