INCIDENT POSTMORTEMS: TIPS, TEMPLATES AND THE #1 SUCCESS FACTOR

Stuff happens! Our IT systems are incredibly complex. Inevitably, things will break and customers will experience the consequences of these failures. You will feel stressed, angry, frustrated, and pressured to get it fixed ASAP. Once you have the problem fixed and major systems restored, you probably want to forget the whole thing ever happened. Don't.

When we have experienced a major loss or degradation of our IT services, it is essential that we learn from what happened. A learning approach ensures either that the incident doesn't happen again, or that we can remedy the situation more expediently than the first time around.

Of course, our learning is no use if we can't remember what we learned. Thanks to how our brains work, we tend to forget the specific highs and lows of a project, especially when trying to recall them months or years later. And that's why we must document our lessons learned—in a document often known as a postmortem.

Major incident reviews, or incident postmortems, form an important part of any continual improvement program. These reviews are opportunities to improve both our IT infrastructure and, possibly more importantly, our processes for dealing with these events. A mature organization will see these events as valuable learning opportunities, rather than apportioning blame for errors.

Let's explore incident postmortems, including the #1 factor for their success. Then, we'll cover the benefits, rules, and best practices for creating your incident reviews.

What is a postmortem?

Performing a postmortem may sound a bit dark and depressing—it literally translates to "after death"—but it's actually meant to shed light on a significant problem. A postmortem process comes at the end of a project and helps you both determine and analyze successes, non-successes, and failures. The outcome of this process is a document or report that aims to inform best practices and mitigate risks in the future.

Postmortems, or lessons learned reports, can be performed after anything:

- The completion of a project
- The end of an event
- The close of a sales season
- The end of a sprint

In IT, most postmortems tackle incidents: a severe problem, downtime, or outage that has an immediate impact on users. The postmortem should document detailed information regarding every aspect of the incident: from the root cause to the successful resolution, and all the lessons you might glean from the whole thing.

Postmortems that fail

Perhaps you've been involved in an incident postmortem, but decided to scrap it for more "important" work. Maybe you filed the report but, now that it's hidden away, the recommendations therein haven't been adopted.

These are the two biggest problems with creating IT postmortems: people dismiss them as non-essential, so the reports aren't always read, let alone adopted, by the people who can affect change. Because of this, many people immediately see postmortems as an unworthy investment of time and resources.

A few reasons point to why we might dismiss documenting these lessons learned:

- We think our memory is better than it really is, that we'll remember what happened and change your actions accordingly.
- It feels like a blame game: determining who did something incorrectly at a moment of significance. Better to avoid that game altogether.

For a postmortem to be useful, it must provide specific recommendations for changes, such as policy or processes. If it's just documenting for documenting sake, it's a waste of everyone's time.

#1 factor for incident postmortems to succeed

In my opinion, the most critical success factor for incident reviews is that they are **blameless**.

To use a popular phrase: do not make your incident postmortem a witch hunt. 'Blamestorming' sessions do not benefit anyone. If your company culture seeks out the person who may have caused, through error or omission, a major outage, it is extremely unlikely that you will get truthful answers during the review. (Besides, most incidents are more nuanced than one person failing at their duties.) In this culture, no smart person would be willing to raise their hand and admit a mistake. When that happens, your postmortem has failed before its begun.

Consider a company culture that **rewards honesty** rather than demonizing mistakes. People will put up their hand willingly to flag an error they may have made. Then, real and useful changes can be made to prevent it being made again in the future.

Benefits of incident reviews

A successful postmortem goes well beyond reviewing how you handled its resolution—the best ones indicate unknown system problems and highlight areas you can improve or automate to reduce risk. A well-run postmortem allows your team to come together in a less stressful environment to achieve several goals:

- Work through what happened
- Discover previously unknown system vulnerabilities
- Mitigate the possibility of repeat incidents
- Uncover any potential process improvements that could speed up resolution of the next major incident

Of course, incident reviews aren't just for internal stakeholders. Ultimately, your incident reviews show your customers two important characteristics about your company, which provides invaluable benefits:

- **Your willingness to learn from mistakes.** Reporting the findings of your review back to your key customers, owning any errors or omissions and giving them a roadmap showing what you are doing to improve the reliability of your services will build your reputation as a valued business partner.
- **Your ability to create new processes that ensure your customers aren't impacted by the same issue again.** Showing that you understand and take seriously the impact of IT outages on the wider business is essential to growing a relationship based on mutual respect.

How to conduct incident postmortems

Like many things in IT, incident postmortems run much more smoothly (and take significantly less time) if you have a process and some basic rules in place. So, let's set a few:

1. **Have a template.** Create a template that you will work off for each review. This ensures you don't miss anything. A template also provides the basis for the reporting, that goes to your management team, and the communications that goes out to affected customers and stakeholders.
2. **Define roles and owners.** The owner of the review is responsible for managing the meeting and producing the subsequent report. The owner(s) should be someone who has sufficient understanding of the technical details, familiarity with the incident, and an understanding of the business impact.
3. **Set rules around which incidents need reviews.** You must have clear, well defined rules about which incidents will trigger the postmortem process. A good rule of thumb is any incident that has been given a severity one rating. There may be other incidents where a review may be useful. Consider establishing a process whereby service owners can request reviews of incidents that do not meet the severity criteria but that may have severely impacted their services and customers

4. **Act timely.** A critical incident will almost always require some downtime for your team; do not delay any longer than necessary. Procrastinating too long means that important details are forgotten. So, when a critical incident occurs, convene within 24-48 hours, and certainly do not delay more than a week.

Create a postmortem template

The responsibility to research, write, and publish a postmortem report lies with the project manager or the person most responsible for a particular outage or data loss. (By responsible for, we mean the person who immediately begins fixing it, not the person who caused it—as many times, these outages occur without human interference.)

An IT postmortem report need not be complicated. In fact, its simplicity encourages us to complete them and others to actually read them. Include specific information that focuses on the key factors of the incident without bogging the reader down with unnecessary details. Here are the core components of a successful post-mortem report:

Summary

First, create a brief summary of the incident. This part of the document should be short, just 1-2 sentences that answers the question "What happened?" This lets readers determine if this report applies to them. Also include details like a relevant, easy to understand title; authors and date; most recent status.

Background

Next, include any supporting information that's necessary for understanding the incident should be provided immediately after the brief summary. This information offers supplementary (but still concise!) details to help the reader understand the context of the incident.

Incident

Now you're into the body of the postmortem report. Include a description of the events that's detailed enough so that someone who wasn't involved in the incident can understand what occurred. Use timestamps to provide insight into how and when everything unfolded. Use these questions to guide your writing:

- What was the problem?
    - How incident was detected
    - Size and time of event
    - Software used
    - Impact
- Why it happened? How was it resolved?
    - Identify major events
    - Isolate root causes, if possible
    - Look at technical pieces: Were design, process, poor maintenance the underlying cause that lead to a technical failure?
    - Looking at non-technical pieces: How did organization, management, and team

    environment improve or detract from the problem and its resolution?
    - What about the effect of things like culture, time crunches, and budget pressures?
    - Include who worked on the incident
- How did the team respond?

Detail any decisions that were made and the steps that were tried, both successfully and unsuccessful, towards incident resolution—and timestamp these, too. This is important for informing the resolution of future similar incidents as well as tracking important metrics like response times and service outage times.

Timeline

This section should provide readers with a bullet point-style reference for every event and action during the incident and its resolution. A simple graphic and short descriptions is plenty. If your timeline is too long, move it to the end of your report so it doesn't bog down the reader.

Takeaways

This section can be broken into three parts:

- **What went well.** Detail where your systems or actors performed well and helped to reduce the impact of the incident. Don't be modest! It's important to understand what systems and decisions had a positive effect.
- **What went poorly**. Admit where systems broke down or decisions were wrong. Discovering areas for improvement is essential for ensuring that future incidents can be mitigated or avoided better in the future.
- **What to do in the future.** Analyze the incident details and outline your learnings from the experience. Then—and this is essential—make recommendations for changes that should be made, both in the short- and longer-term. Acknowledging your learnings without suggesting change is a failed opportunity.

Best practices for incident reviews

Even with rules in place, an incident postmortem can go all over the place. Consider these best practices as you embark on your next incident review, and then revisit them with each postmortem iteration.

Conduct a review for every incident classified as 'major'. Every major incident! Even if it's too hard. Even if you already know the root cause or you've developed a permanent fix. Don't skip any major incident review. Remember that not everyone is aware of the final resolution or the steps that were taken. The review is as much about reviewing how well your process performs as it is about finding the technical or true root cause.

Do it right away! The time for a postmortem is immediately after you've wrapped the project or as soon after the triggering incident as possible, especially if it had an immediate impact on users, such as an outage, downtime, or data loss. The postmortem process should be built into your scheduling. If not, you lose precious recall around exactly what happened and how good or bad something was. We tend to remember really bad things, gloss over other things, and forget our successes

Choose a moderator. Ensure that one person controls the room, so that it stays on track and doesn't

become a “blamestorming” session. Typically, the moderator is the owner of the incident review, whom you've already designated. If not, perhaps rely on a person who can command a room. The moderator is responsible for maintaining order and giving every participant the chance to speak.

Involve many people. Most major incidents involve many players from internal and vendor teams. The review gives everyone a chance to contribute their views and learn from the experience. Beyond this specific incident, being inclusive helps build trust and resiliency in the team, creating relationships that will help the next major incident war room run more smoothly.

Lay the ground rules at the start of your meeting. No finger pointing, no dismissing anyone's ideas. Treat everyone with respect.

Single out no one. Successful postmortems are blameless postmortems. Do not single out any individuals as being responsible for the incident: it's negative and it wastes time. Instead, you must concentrate on actions, results, and impact.

Use “The 5 Whys” technique. I like this technique and promote it often. First, make sure everyone is on the same page about the original problem and its details. Then, ask why that happened. As you get that answer, ask why again. Keep asking “Why?” at least five times. This ensures you uncover **all** the underlying factors that contributed to the incident. The information obtained from this exercise will also form the basis for the ongoing problem investigation.

Don't let participants shy away from uncomfortable truths. In group settings, it's easy for participants to choose the truth of least resistant, or come to an easy or convenient consensus on cause. The owner/moderator should prevent this from happening.

Do not skimp on time. Your incident review is all about detail—things that did not seem important during the heat of the incident may provide valuable insights that could help with understanding the root cause. Give everyone a chance to contribute, and consider each and every one of those contributions, no matter how far-fetched they may seem.

Use a tried and true template. You're not writing award-winning stuff here, it's the recommendations that matter. A good template means you don't have to worry about how well you write—and that you don't waste hours or days on the effort. (A quick online search turns up dozens of templates; experiment to find what works best for your team.)

Track positives and negatives. Not all postmortems have to be gloom and doom – some can highlight positives in a process that you may not have been aware of. In that case, perhaps your recommendation is to rollout these positives more widely.

Publish the report. Postmortems don't have to lurk in a basement storage area, among old files. In fact, you don't even have to print it out – simply share the findings with the team, the department, or the company and decision makers as whole, whatever makes sense for your work environment. A bonus: publishing will help you keep things short and concise, too!

Review your postmortems. The last thing I will leave you with: reviewing your incident reviews encourages you to do better next time, and there will be a next time. For continual improvement, *everything* we do contributes.

Good incident management reinforces problem management

The outcome of (and attitude around) IT postmortems won't improve if you continue to minimize the

importance of IT postmortems. Next time you create a postmortem, consider following a reliable template and commit to implementing the changes.

Of course, postmortems should be seen for all their positives: finding good processes that can apply to other teams and functions, improving processes iteratively so it's easier to implement and maintain, and supporting problem management.

Additional resources

For more on incident management, see the BMC Service Management Blog, or check out these articles:

- Incident Management vs Problem Management for 2020
- Incident Management: A Beginner's Guide
- Incident Management in ITIL 4
- How To Map the Incident Management Process
- 5 Ways to Improve Incident Management with Decision Support
- How and Why to Run Incident Drills