Pager Rotation best practices

How to handle on call rotations with DevOps.

Brett Fuller

CSD-380 - Assignment 7.2

7/5/2025

Create clear rotation schedules.

It is imperative to establish a clear responsibility schedule to evenly distribute responsibilities and avoid burnout. There are several different scheduling methodologies you could utilize from: Frequency based scheduling, to a responsibility based methodology, to utilizing a location based follow the sun kind of rotation.

Establish on-call responsibilities and incident playbooks.

Have a clearly drawn up matrix of responsibilities to ensure that issues are routed to the correct team along with well documented runbooks and incident response guidelines. According to blameless.com "You don't want team members to feel adrift while on-call, especially when it's odd hours and they can't just ping team members with questions. So it's essential to set up expectations and responsibilities from the start and save documentation so team members can refer back to it."

Balance workloads

Be accommodating to staff on the pager rotation. The afterhours responsibilities of pager duty can get overwhelming when paired with the employee's regular responsibilities. Providing the ability to offload some of their daytime work after late nights can help prevent burnout.

Distribute responsibilities across multiple teams and encourage cross team communication

If the on call rotation is focussed on a single team the workload can be overwhelming for that team. Additionally it can become routine to treat the system and not the disease. An example might be that when a website drops an infrastructure team may restart the server to restore service. With this action the team treated the symptom and the website was once again functioning. However the cause of the crash was not discovered and it could very likely happen again and since a solution has been discovered of restarting the server that becomes protocol for downed website. If instead there were communication between the teams and the developers reviewed the logs to discover that issue x had caused the problem they could prioritize resolving that issue to avoid future occurrences.

Regularly review the types of incidents that occur

By regularly reviewing the incidents you can find trends and work to attack the root cause of the issue. Think of these reviews as a sort of Andon cord where you prioritize the issues that are regularly causing pages to occur and address them to reduce call volume.

Avoid Alert Fatigue

When a monitoring system produces too many alerts that are not urgent it can cause the on call personnel to filter out messages from that system and as a result it can be easy to miss the signal through the noise. By ensuring that your system is only alerting on critical actionable issues you can improve response time because your employees will pay more attention to those alerts.

Embrace Automation

If you combine the learnings of the last several practices in recognizing regular issues and firing off actionable notifications a knowledgeable team can leverage automation on alerts to resolve issues long before a person could ever respond to a page. This not only increases efficiency but can also prevent the fatigue that comes from performing simple actions in response to a call, especially if that call is outside of work hours.

References

Cortex. (2021, February 13). *Best practices for your team's on-call rotations*. cortex.io. Retrieved July 5, 2025, from https://www.cortex.io/post/best-practices-for-on-call-rotations

Kim, G., Humble, J., Debois, P., Willis, J., & Forsgren, N. (2021). *The DevOps Handbook, 2nd Edition: How to Create World-Class Agility, Reliability, & Security in Technology Organizations*. IT Revolution.

Nizami, M. (2022, September 28). *Best Practices for Creating On-Call Rotations and Schedules*. blameless.com. Retrieved July 5, 2025, from https://www.blameless.com/blog/on-call-rotation