




Best Practices for Pager Rotation Duties in DevOps



Sarah Ewing
11/28/2024
Module 7.2 Assignment



BENEFITS OF EFFECTIVE PAGER ROTATION

ENHANCED TEAM TRANSPARENCY AND ACCOUNTABILITY

Ensures that all team members are clearly responsible for addressing issues.

IMPROVED CUSTOMER SATISFACTION

Customers have confidence that urgent issues will be quickly addressed by on-call staff.

INCREASED SERVICE RELIABILITY

Swiftly responding to and resolving alerts leads to more reliable service.

TIME EFFICIENCY

Reduces delays in engaging on-call staff to handle issues promptly.

CHALLENGES OF PAGER ROTATION

ALERT FATIGUE

Excessive alerts may cause engineers to ignore important notifications, delaying critical issue resolution.

INADEQUATE KNOWLEDGE

Unfamiliar on-call staff may struggle with issue resolution causing longer resolution times or escalations.

BURNOUT AND STRESS

Continuous on-call shifts can cause stress and burnout, affecting personal life and job performance.

LACK OF OWNERSHIP

Lack of on-call responsibility for developers can result in lower code quality and more production issues.

POOR ROTATION DESIGN

Uneven on-call rotations cause resentment and unfair workload distribution.

On-Call Scheduling Examples

<u>Location-Based (Follow the Sun)</u>	This model leverages team members across various time zones to ensure continuous coverage and preventing disruptions during local nighttime hours.
<u>Responsibility-Based</u>	This model assigns pager based on expertise and responsibilities, ensuring that the right personnel are alerted for relevant issues.
<u>Frequency-Based</u>	This model organizes on-call duties based on a rotating schedule, such as weekly, semimonthly, or week/weekend.
<u>For Smaller Teams</u>	Creating on-call schedules for teams of five or fewer that avoid burnout and alert fatigue can be challenging. Alternating days or weeks is a best practice. Rotating backup schedules and third-party monitoring can help balance workloads.

PAGER ROTATION BEST PRACTICES

- Consider software for automation to save time and minimize manual overhead.
- Set up teams with on-call responsibilities and appropriate monitoring.
- Define escalation policies to manage incidents effectively.
- Establish time limits for responses to ensure issues are addressed promptly.
- Enable easy overrides for shift swaps due to unexpected events.
- Ensure 24×7 coverage with proper shift planning.
- Maintain transparency and communication about schedule changes.
- Keep team members aware of their on-call hours for better planning.



Source: <https://www.pagerduty.com/resources/learn/call-rotations-schedules/>

MORE PAGER ROTATION BEST PRACTICES

Implement DevOps practices: Remove communication barriers; delegate issues to the right team; developers support their own code, ensuring better software.

Assign teams and team roles: Create teams with on-call duties for each service; on-call engineers can quickly contact teammates if needed.

Automate menial on-call tasks: Use scheduling software to notify engineers of issues automatically, reducing overhead and human error.

Set up escalation policies: Define a clear structure for who is contacted first and their actions to address the issue.

Be proactive about discovering incidents: Identify and address incidents beforehand to boost customer satisfaction.

Monitor on-call performance metrics: Track metrics like the number of pages per engineer and non-business hours paged to find areas of improvement and reduce overwhelm.

Source: <https://www.cortex.io/post/best-practices-for-on-call-rotations>

RESOURCES

- <https://www.blameless.com/blog/on-call-rotation>
- <https://www.pagerduty.com/resources/learn/call-rotations-schedules>
- <https://incident.io/hubs/on-call/on-call-rotation-best-practices>
- <https://www.squadcast.com/sre-best-practices/on-call-rotation>
- <https://www.cortex.io/post/best-practices-for-on-call-rotations>