Pager Rotation Duties in DevOps

Best Practices for Pager Rotation Duties

Bella Apo

CSD380

February 10, 2025

What is Pager Rotation?

- A DevOps practice where developers take turns being on-call to address software and system issues
- Helps teams prevent recurring issues that could last from days to years
- Provides valuable feedback on how upstream coding and design decisions impact production
- Ensures operations teams do not bear the full burden of production-related code issues

Importance of Pager Rotation

- Developers gain first-hand insight into real-world application issues
- Encourages collaboration between development and operations teams
- Balances fixing problems with time allocation for new feature development
- Market-oriented teams benefit by ensuring a seamless production experience

Responsibilities During Pager Rotation

On-call engineers must respond immediately to software/system issues

If an issue cannot be resolved, it should be escalated to a more experienced team member

Ensuring 24/7 availability helps maintain system reliability and user satisfaction

Developers learn how customers interact with their product and can improve user experience

Challenges

Alert Fatigue - Overwhelming notifications can lead to burnout

Work-Life Balance Issues - Being on-call outside regular hours can be exhausting

Lack of Real-Time Progress Visibility - Management may not immediately see feature completion as teams focus on stability first

Handback Process - Some companies (e.g., Google) have structured ways to return unresolved issues to developers while maintaining fairness.

Best Practices

PagerDuty Recommendations

- Rotate on-call duties across the team to avoid fatigue
- Escalate issues when necessary to ensure quick resolution
- Ensure reliable coverage so critical issues are always addressed

Incident.io Recommendations

- Automate scheduling with software tools
- Clearly define team responsibilities
- Set up escalation policies with defined time limits
- Allow shift coverage flexibility to maintain work-life balance

Preventing Burnout and Improving Efficiency

- **Distribute on-call responsibilities** among multiple engineers.
- **Ensure transparent communication** within the team.
- **Use automation** for scheduling and alerts.
- **Maintain proactive monitoring** to prevent issues before they escalate
- **Encourage post-incident analysis** to improve future responses

You Build It, You Maintain It"

Emphasize accountability

Developers should follow their work into production

- Gain direct insights into user issues
- Improve coding and architecture decisions
- Ensure a seamless experience for end users

Help create a culture of shared responsibility between development and operations teams

Conclusion

Pager rotation is essential for system reliability and developer accountability

Best practices help prevent burnout and maintain efficiency

Organizations must balance reliability, transparency, and work-life harmony to create sustainable on-call policies

Investing in structured rotation policies ensures long-term DevOps success

References

- Incident.Io. (2024, February 26). Best practices for creating a reliable on-call rotation. *incident.io*. https://incident.io/hubs/on-call/on-call-rotation-best-practices
- Kim, G., Debois, P., Willis, J., & Humble, J. (2016b). *The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations*. https://dl.acm.org/citation.cfm?id=3044729
- *On-Call rotations and schedules | Articles | PagerDuty*. (2023, January 17). PagerDuty. https://www.pagerduty.com/resources/learn/call-rotations-schedules/