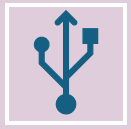
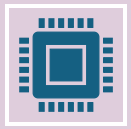
Pager Duty in DevOps

Presented by: Candice Garcia

What is it?



Coined after the popular 90s device, pagers were a prominent way of getting a hold of people. The term was used early on at Amazon and stuck as a visual and conceptual name for on-call systems.



This idea helps make sure the right person in operations or development gets notified when it’s their turn to handle an important message or issue.

Developers who worked on the code are expected to be available when something needs attention, especially when the issue is related to their specific work.

How does it work?

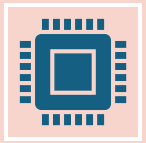
The teams have on-call rotations scheduled, and they must be available to respond to any issues.

This is assisted by automated tools like PagerDuty, which notify the developer if an incident occurs, requiring them to respond.

There is assistance with tools like monitoring alerts, which notify the right person when an incident is detected.

They must work on the problem to fix it and write a review to analyze what happened, where the issue was, and how to prevent it in the future.

What are the benefits?



Developers are expected to take accountability for their code, which helps them stay mindful of how it performs in production.

This can help improve their work because they see how it acts as it's implemented in the real world. Their code follows them through the whole process.

A lot of this is automated, which helps with scheduling notifications where critical alerts are priority an customers can be protected as issues are resolved more quickly.

This aids collaboration by sharing responsibilities across the team and improving communication throughout the lifecycle. Switching roles also helps balance the workload.

Best Practices

Ensure on-call duties are balanced and fair to prevent burnout by distributing the workload evenly.

Automation streamlines processes by prioritizing critical alerts and reduces unnecessary interruptions.

Post-incident reviews help improve future responses and drive continuous improvement.

Having training and support in place helps each team member be aware of their responsibilities and feel well prepared

How Pager Duty relates to the DevOps model

Developers share responsibilities by being accountable for the code they work on.

Collaboration is vital, with development and operations working together as a team.

Progress happens continuously thanks to post-incident reviews and real-time feedback.

Incident responses are built into the pipeline to improve development flows.

The goal is to make systems reliable, stable, and ready for production.

Companies already using this

Zoom

Glovo

Chicago Trading Company

Wiley

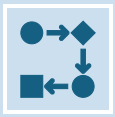
Conclusion

Accountability is important among developers to make sure they take responsibility for their code.

Reports from incidents can help find areas to improve the software over time.

Collaboration and ongoing improvements are essential to keep everything working smoothly.

Following best practices is key to making pager duties successful.



The focus is on bringing development and operations teams together for better results.

Sources

- *What is PagerDuty in DevOps? - Matoffo*. (2017). Matoffo.com. https://matoffo.com/what-is-pagerduty-in-devops/
- Solomon, A. (2019, February 21). *A Decade of Duty: A Brief Look at PagerDuty's History*. PagerDuty. https://www.pagerduty.com/blog/decade-of-duty/
- Kim, G., Humble, J., Debois, P., Willis, J., & Forsgren, N. (2021). *The DevOps Handbook, Second Edition*. IT Revolution.
- *PagerDuty for DevOps*. (n.d.). PagerDuty. https://www.pagerduty.com/resources/solutions-brief/pagerduty-for-devops/