Pager Rotation

Presented by Jaci Brown
For CSD 380 Module 7.2
09/21/2024

Agenda

- Introduction
- Pitfalls
- DevOps & Minimizing pager calls
- Implement rotation well
- Learn from the rotation

We all know that sometimes things go wrong. In critical situations, business hours are out the window. But can we make it better? Less frequent?

Pitfalls

We all know that pager rotation can be rough with a high-profile project. Initially, there are many errors while things are fine tuned. Automation can help, but it is also tricky, leading to unneeded alerts. Perhaps the scheduling is slimmed, leading to the available rotation members not being capable of resolving the alert. Many of these pitfalls lead to unhappy development and ops teams, yet these things are not surprises to anyone. Can we reduce these negatives?

A DevOps perspective can minimize alerts.

Control the flow

A project in development can have many errors as new features are blended with existing. Before developers pass control to operations, they should manage the code failures until it can be considered stable.

After a project is mature, members from development and operations should be on call to handle different types of situations. This also includes separate teams/specialties within departments so that each alert is directed to the appropriate on-call team, minimizing overall alerts.

Use automation and escalation effectively

With a DevOps mindset, we can use our analysis tools to reduce over alerting by carefully setting thresholds for problems. Additionally, the severity of the alert auto alerts could indicate a waiting period, or to pull the Andon Cord.

After our alerts are streamlined, effective scheduling and on-call mapping can help the first one alerted know who to contact to resolve the issue. This streamlines the flow, reducing time of project unavailability and undue stress for ill-assigned alerts.

Retrospective

Learning from experience

Implementing all of these rotation practices will help reduce alerts alone. But we can view the tickes and alert metrics to see how to improve further. Patterns may emerge in time or frequency of calls that can lead to more stable projects. Common occurrences can receive bounties/priorities to increase speed of repair or workaround.

Final tips & takeaways

- Critical systems require critical alerts
  - This is a necessary ‘evil’ or unpleasantry.
- Set up scheduling and communication for success
  - Streamlining the process reduces those awake to only critically necessary.
- Learn from internal patterns
  - Use metrics to identify potential weaknesses and improve overall stability, reducing total calls.

References

[1] https://www.cortex.io/post/best-practices-for-on-call-rotations

[2] https://alertops.com/on-call-rotation/

[3] https://blog.symops.com/post/your-on-call-rotation-is-painful#:~:text=You%20should%20set%20an%20incredibly,the%20night%20for%20no%20reason.

Add my notes?