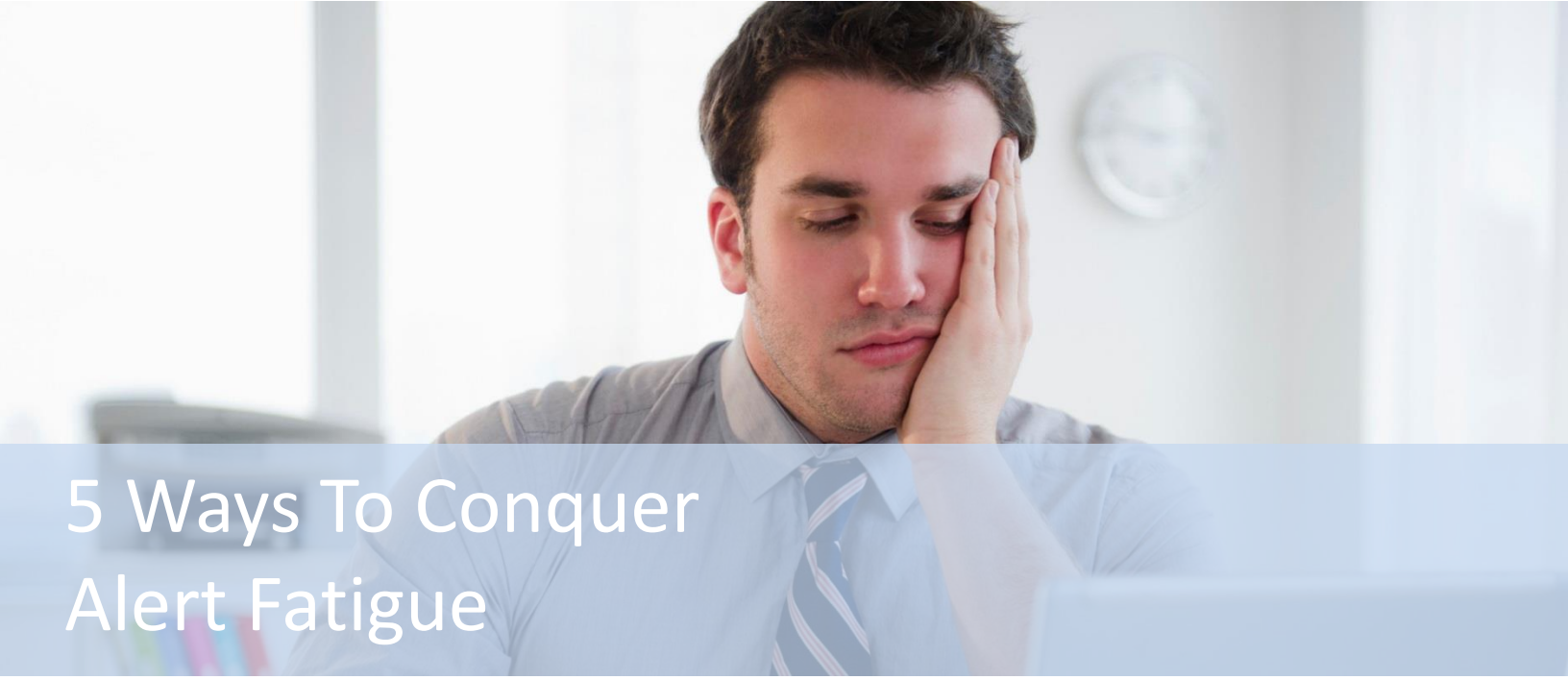
WHITEPAPER

5 Ways To Conquer Alert Fatigue



5 Ways To Conquer Alert Fatigue

The problem of alert fatigue is considered to be the #1 pain point for both traditional IT teams as well as modern DevOps engineers. Especially for those who provide operational support for their applications and production infrastructure. *(1)*

Alerting is a key component of today's IT environment.

Alerting is what enables IT professionals to know when, for example, their company's servers are down, apps aren't functioning or the website is down. However with the advent of more technologies to monitor cloud, infrastructure and microservices, alerts are often poorly calibrated and come in too frequently. As a result, alert fatigue ensues and your organization suffers.
The goal of this white paper is to:

- Define alert fatigue
- Highlight the costs to the organization of alert fatigue
- Offer solutions to alert fatigue

(1) https://medium.com/@BotmetricHQ/square-away-the-alert-fatigue-mess-with-devops-intelligence-f1ab5f408333

460 Totten Pond Road,
Waltham, MA 02451
781-916-0040
sales@onpagecorp.com
www.OnPage.com

Alert fatigue - a definition

Alert fatigue is the feeling of exhaustion and dissatisfaction resulting from receiving excessive numbers of alerts or alarms. The alerts, usually arriving in the form of emails or SMS messages, are often the result of non-actionable alerts, false alarms or improper configurations which cause too many event notifications. Engineers can often express symptoms akin to depression when suffering from alert fatigue.

Most IT shops rely on multiple monitoring and log management systems. The same alert may appear in different monitoring systems. Without correlation and calibration, these multiple points of alerting lead to engineers suffering from alert fatigue.

According to one Google engineer (2) , alert fatigue sets in when he receives alerts more than a few times a day:

> *Every time my pager goes off, I should be able to react with a sense of urgency. I can only do this a few times a day before I get fatigued.*

Alert fatigue can thus be seen as resulting from both *frequency* and a *lack of coordination* in the alerts an engineer receives.

The costs of alert fatigue

The cost of alert fatigue is seen in both the financial and personal realms.

When engineers become numb to the overload of alerts, they are unable to respond to critical issues. Issues like latency in service or customer data being unavailable should cause a quick reaction from the DevOps or IT team. However, if these issues are not handled swiftly, customers might go elsewhere with their business.

Businesses might also have contractual obligations with customers to provide IT services. If these obligations are violated, the business could suffer financial consequences as a result.

The cost the engineer feels personally is also significant. If the engineer suffers from alert fatigue, they are usually tired, grumpy and irritable. They are not as productive and don't work well with the team.

460 Totten Pond Road,
Waltham, MA 02451
781-916-0040
sales@onpagecorp.com
www.OnPage.com

(2)https://docs.google.com/document/d/199PqyG3UsyXlwieHaqbGiWVa8eMWi8zzAn0YfcApr8Q/edit

Additionally, the engineer will likely look for greener pastures by getting another job and the employer will have to look for new recruits. The cycle of hiring new engineers can become expensive, with estimates ranging from 1.25 to 1.4 times base salary (3).

Solutions to alert fatigue

1) Decrease alert to noise ratio: calibrate systems

Often, engineers are alerted for non-critical events or events for which no action is required. While it might seem like it's better to get a false positive than to miss an alert, there is definitely a cost to receiving too many false positives. In the IT world, this is compared to crying wolf. Like the boy who cried wolf too many times, alerts that are not actionable get ignored after a while.

As Twitter engineer Caitie McCaffrey noted at the 2016 Monitorama Conference,

"when alerts are more often false than true, the on-call's sense of urgency in responding to alerts is diminished...the simple burden of alerts desensitizes the on-call to alerts".(4)

The solution is to review the monitoring tools used by the IT team and calibrate them so that noise is dampened while real alerts rise to the surface.

Alerting thresholds	Recommendations
Set threshold levels	Avoid setting thresholds too low. For example, if you set an alert to notify you when a storage bucket in the cloud hits 50%, this is probably too low and will increase the likelihood of un-actionable alerts or false positives.
Adjust settings	Adjust your conditions over time. • As you use monitoring tools to help you optimize your team's performance, tighten your alerts policy conditions to keep pace with your improved performance. • If you are rolling out something that you know will negatively impact your performance for a period of time, loosen your threshold conditions to allow for this.
Disable settings	Audit and disable any superfluous alert condition. This is useful, for example, if you want to continue using other alert conditions for the policy while you experiment with other metrics or thresholds.

ONPAGE
SOME MESSAGES CANNOT WAIT

460 Totten Pond Road,
Waltham, MA 02451
781-916-0040
sales@onpagecorp.com
www.OnPage.com

(3) http://www.investopedia.com/financial-edge/0711/the-cost-of-hiring-a-new-employee.aspx
(4) https://vimeo.com/173704290

2) Decrease the sending of emails for alerts

Monitoring tools are used to identifying critical events in the monitoring life-cycle. However, if they are configured to send emails to the group, they are creating a further level of anxiety among engineers.

Engineers fear that they are missing alerts and as a result feel anxiety. By calibrating alerts, as described above, and by ensuring alerts are sent through tools other than email, engineers can diminish their feelings of alert fatigue.

3) Make sure alerts are actionable

Engineers want alerts to be actionable and require their intelligence to fix the problem. No engineer wants a 2am wake-up call that simply tells them "core infrastructure is down". This note is not actionable. Instead, alerts should indicate as much specificity as possible.

Moreover, engineers should keep in mind that they need to monitor and create alerts for their users(5). Users don't care if a MySQL server is down. They care if their queries are failing. It is important to keep this perspective in mind when setting up and creating alerts.

4) Enable high versus low priority alerts

As any engineer who has been on-call knows, not every alert they receive is critical. To mitigate alert fatigue, IT should classify each type of alert as high, medium or low priority(6). High priority alerts are anything that is absolutely critical and must be handled to ensure business continuity. These high priority alerts are also ones that require alerting.

Of course, severity of alerts is in the eye of the beholder. What is high priority for one organization, might only be a medium priority alert for another. There is no clear, right answer on what is a high versus a medium alert. What is important however is to identify what alerts are which.

Furthermore, the mindset should be that high priority alerts are critical notifications that require immediate alerting. Medium and low priority alerts can wait until the next day.

460 Totten Pond Road,
Waltham, MA 02451
781-916-0040
sales@onpagecorp.com
www.OnPage.com

(5)https://docs.google.com/document/d/199PqyG3UsyXlwieHaqbGiWVa8eMWi8zzAn0YfcApr8Q/edit
(6)http://searchitoperations.techtarget.com/tip/Prioritizing-alerts-with-server-monitoring-tools

5) Decide who is notified and how by managing alert structure

The nature of after-hours service is volatile and one cannot predict the kinds of emergencies that might pop up. Consequently, schedules need to be flexible. While you can create triggers based on the conditions set in your monitoring system, it becomes cumbersome when you are unaware of who needs to get notified.

The on-call schedule should be the responsibility of one person on your team who modifies the schedule on a weekly basis according to their working times and availability. Alerts should not get sent to the whole team. Managers should also make sure that the person receiving the alert is someone who can respond to the issue. The alert should not go to someone who will need to foist it off onto another colleague for the issue to get resolved.

Conclusion

Peter Drucker is often quoted as having said that *"You can't manage what you can't measure."*

This is equally true in managing after-hours critical alerts. After the alert has been responded to and remediated, you need to look back at the situation and reconsider if the alert was triggered and responded to appropriately. Engineers should ask if:

- ❑ The event was triggered by a real issue or by an issue that could have waited until the following day or by an issue that didn't exist in the first place
- ❑ Was the alert delivered to the appropriate person? If not, was this because the alert was missed or was the person who received the alert unable to handle the issue?
- ❑ Was enough information provided to the on-call engineer so that they could handle the issue quickly? If not, the alerting information should to be updated so it is more robust.

Integrating the steps mentioned above into an IT team's alerting policy will make a significant impact on the team's morale and level of alert fatigue.

If you have further questions on afterhours alerting best practices, please contact OnPage Corporation at

http://onpage.com/contact-us OR call us at 781-916-0040

460 Totten Pond Road,
Waltham, MA 02451
781-916-0040
sales@onpagecorp.com
www.OnPage.com