

# DevOps In An Unplanned World

*Perfecting Communication During Critical Incidents*

 everbridge®  
Assured Critical Communications

 DevOps.com  
Custom Publishing

Communications is the glue that binds Dev and Ops, especially when things go wrong — here's how to build the strongest bond.



For many years now, traditional IT shops have operated with a natural divide present between developers and operations. Developers work on the code, collaborate with QAs to get it cleaned up and then wash their hands of it once it is shipped over to operations staff to ready it for release and make sure it actually runs in production. In the days of waterfall development, this method worked decently. But as Agile methods have required faster release cycles, this divide introduces friction every time work must be handed over across the dev and ops divide.

## Thus the DevOps movement was born.

### DEVOPS ORGANIZATIONS ARE:

**2x** 2x as likely to exceed profitability, market share, and productivity goals

**50%** Have 50 percent higher market capitalization and growth than rival businesses

The idea behind DevOps is to destroy that divide and encourage more collaboration and accountability between both groups so that everyone feels responsible for the code no matter where it is in the software development lifecycle.



# *Picking up the pace...*

When done right, DevOps practices can rapidly increase the speed of release cycles. With automated tools within the continuous delivery pipeline and a solid culture that promotes everyday collaboration between developers and operations staff, IT organizations can drastically release delays that hold up the time it takes to ship code into production.



*In order to achieve this,  
though, solid lines of  
communication are  
essential.*



Communication is the glue that binds developers and operations staff, plus any additional stakeholders in the process. Without a strong communication layer shared across the organization, DevOps becomes nearly impossible to achieve.

Many teams believe they have got communication covered with a handful of tools. The combination of SMS, email, Skype and ChatOps tools—homegrown or commercial—helps them get in touch throughout the normal release lifecycle.



The background of the slide features two flashlights lying on a reflective surface. The flashlight on the left is black, and the one on the right is silver. Both have their white power cords frayed and broken at the ends, with the frayed ends pointing towards the center of the image. The text is overlaid on this scene.

But what about  
when things  
*break?*

# What happens when critical incidents occur that are outside the bounds of the typical release schedule?

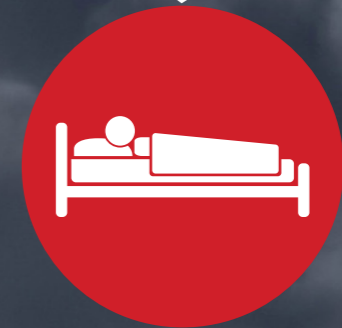
Unfortunately, that's when those lines of communication and processes break down.



The people who are needed to resolve the issue are away from their desks.



They're on vacation and no one on other teams know about it.



Or, it's invariably 2am on a Sunday and they're all asleep.

Simply sending an email invite to a conference call does not wake people up in the middle of the night.

Sending an SMS or even ChatOps message is no better. Waiting on a response is going to result in valuable minutes lost. And even before this process is started, whoever initially finds the issue must be able to identify who the 'right people' are to contact. Just figuring out which teams need to be contacted and who within those teams can start making a fix, plus who among those is actually on call is a process in and of itself. That alone adds more minutes to the equation.

***...and every minute of outage can mean tens or hundreds of thousands of dollars — sometimes even millions — lost when mission critical applications are involved.***

A simple downtime cost equation:

$$LR = (GR/TH) \times I \times H$$

Lost Revenue = (Gross Yearly Revenue / Total yearly business hours) x % impact x hours of outage

Unplanned downtime costs organizations an average of  
\$5,600/minute  
\$46,000,000/year

The point is that most people understand that DevOps is the art of speeding IT up through automation and strong communication. But the really effective DevOps organizations understand that in order to account for when things go wrong, they also need to automate the communication process itself.

# ***Critical Incidents Challenge DevOps Processes***

In theory, DevOps helps put IT processes on rails. The automated pipeline ensures code sails through the development and release cycles. Monitoring in place gives organizations metrics to incrementally tweak processes and improve things along the way. And regularly scheduled calls, meetings and lines of digital communication enable developers and operations staff to work out minor kinks as they come up. But the truth is that this utopia depends on everything operating normally. And as anyone with even a little experience in IT will ask, when do things ever go as planned in an IT department?



# ***IT problems happen all of the time.***

These could be failures in hardware or patches in third-party commercial systems that don't play well with internal code. Or it could be a release deployment issue that requires an emergency roll-back. Regardless, when these incidents occur and when massive outages present themselves, the question to ask is how quickly do those regular lines of communication really inform the people who need to know about it to resolve an issue?

In order to solve big problems in IT, it usually takes a team of people from both dev and ops to understand all of the details of the situation. Even ITIL doesn't recommend best practices on these occasions--it usually leaves it up to the organization to wing it with their own process.

Whatever that process is, it will usually require a group call with people across multiple teams. It will also require that all of these people are looking at the same set of diagnostic information from relevant monitoring tools and the application in question.

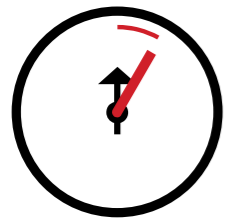
Getting to that point can eat up a tremendous amount of time. One SaaS provider explained with DevOps and a normal set of communication tools, it'd take about 45 minutes to get the right people apprised of the details of a critical incident

# ***What's a 'Big Problem' in IT?***

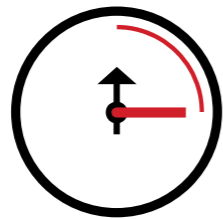
Measuring your biggest issue is all about the impact on users and urgency. Even a seemingly small technical problem can have disastrous effects if not handled swiftly.



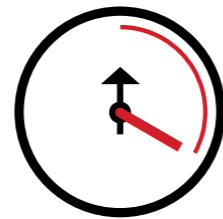
# In a typical response scenario, little delays add up quickly. Here's how things can break down for a typical organization:



The incident is flagged and someone establishes that there is a problem.



Service desk personnel or an incident manager looks up the right people to inform. *Even in the case that an organization has a critical incident response team (CIRT) in place, someone usually needs to look up contact information and find out who is on call.*



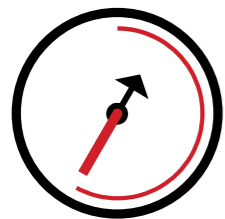
An email is sent to people who already receive hundreds of emails a day.



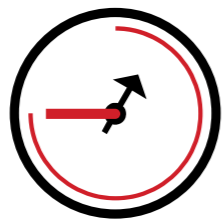
The incident manager gets no response from the majority of people. *Some people are not at their desk. Others are asleep.*



## Typical Response Process



An SMS is sent. A few additional people respond.



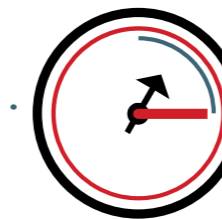
The incident responder still needs to get in touch with more critical people and spends time looking up emergency phone numbers.



The manager makes these calls and wakes up several people, some of whom are angry because they're not on call. *Two people are on vacation, another is out sick and someone else is on maternity leave.*



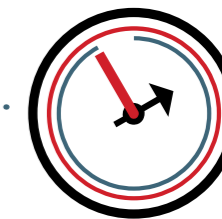
The manager makes these calls and wakes up several people, some of whom are angry because they're not on call. *Two people are on vacation, another is out sick and someone else is on maternity leave.*



Conference call is set.



Time is spent troubleshooting international numbers so everyone can join call.



Everyone joins, then at least 15 minutes are spent explaining where team can find relevant information about problem in tools like AppDynamics and Solarwinds.

Clearly, the process of getting everyone aware of the problem and on a course for resolution requires multimodal communication. But to really do it effectively, it also needs to be automated. An incident manager doing it him or herself manually will need to try multiple types of communication tools to find the one that connects, will need to discuss issues on the phone and need to share information from multiple IT tools. There's a lot of delay and lag that can occur as this happens manually.

What's more, there are additional curveballs that can introduce additional friction to the process:



### **CAB**

Once everyone has decided on a resolution, they may need to go through this process of once again to get approval from change approval board (CAB), as CAB members will need to be contacted swiftly



### **INTERNATIONAL CONCERNS**

Team members may be spread internationally across different time zones and different international calling codes will be needed for group calls. What's more, language could be a barrier. Incident managers will need to make sure communication is crystal clear, especially for non-native English speakers. And when translations are done, it is crucial nothing is lost in translation.



### **VENDOR/THIRD-PARTY CONCERNS**

Relevant contractors or vendors often need to be brought into the loop to resolve certain issues. This eats up more time to find the right contact at the third-party organization, to get in touch and connect with them



### **NOTIFYING CUSTOMERS**

In customer-facing applications, there may be a need to send out mass messages so they're apprised of the situation. And more importantly, so they stop clogging up the service desk with their calls.



# Automating Communication: A Case Study

## With the right tools in place, here's how a critical event communication chain CAN look...



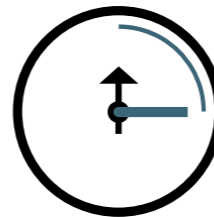
An event is flagged by automated tools, an end user or network operations center (NOC) staff

An operator identifies issue and triggers the process to reach available CIRT members.

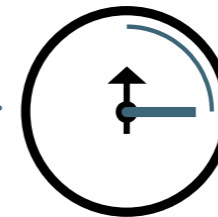
IT service alerting tool already has relevant people's contact information, scheduling and geography in its system.

The right on-call team members dev and ops CIRT team are contacted using the mode of communication most likely to receive an immediate response based on the time of day for that person and their schedule.

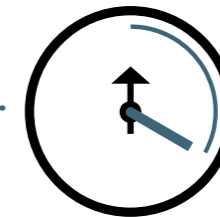
The IT service alerting tool also automatically notes when people are unavailable due to vacation, travel or illness and rolls communication to the person covering for them.



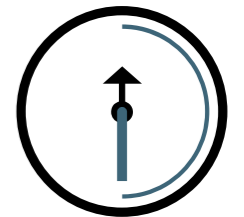
The system automatically sets up a group call and completely bypasses the nightmare of setting up international friendly calls and complicated dial-in numbers, shaving more minutes off the process.



And, most importantly, the system seamlessly integrates with tools to give people a view in relevant dashboards of the monitoring data and diagnostic information to understand what is going wrong.



When the group call is on, participants can be looking at that data in real time and discuss resolution steps immediately.



Once the course of resolution is decided, system can also automatically send alerts and request for approvals to CAB.

Remember the customer that took 45 minutes to even get on a conference call and the right people informed?

When implementing automated communication tools to get people informed and initiate a group call, this time was slashed in half to 22 minutes.

Once IT tools and dashboards were integrated into the communication platform, this was cut down to 7 minutes.



***Over the course of numerous critical incident, this could mean a significant savings to the organization.***

## Everbridge and DevOps

Everbridge's IT service alerting tool can be the glue that firmly connects developers and operators even in the toughest outage situations. Just as automated tools within the continuous deployment pipeline reduce the delays in delivering code to production, the automation of communication ensures that CIRT members are quickly synced to resolve outages as quickly as possible.

Everbridge is a global provider of SaaS-based unified critical communications solutions for over 2,600 corporations and communities. Our enterprise scale platform and communication applications for Mass Notification, Incident Management, IT Alerting, Secure Messaging, Community Engagement and the Internet of Things (IoT) enable customers to quickly and reliably deliver the right message and reach the right people, to more than 100 different communication devices, in over 200 countries and territories – all simultaneously. The Everbridge IT Alerting application specifically helps quickly connect the right on-call IT personnel, with the right information, to resolve critical incidents faster and mitigate their negative business impact.

