5 Questions for DevOps Teams Who Want to Move Faster

A white paper for DevOps teams who want to move faster, automate for higher productivity, and get early warning of service-affecting problems. Five questions are examined, leading to a better way for responding to incidents and alerts across both pre-production and production application service environments.

WHITE PAPER

For information about Moogsoft visit www.moogsoft.com.

Moogsoft®

Introduction

This white paper explains how DevOps team can move faster and achieve greater productivity with automation in three areas:

- Dev / quality assurance (QA) and automated pre-production testing
- Alert and incident notification
- Correlation, diagnosis and collaborative remediation
- To frame the discussion, we pose five questions for DevOps teams to consider as they do their work.

Questions:

1. What's your delivery lifecycle today, and what do you want it to be?
2. How can you detect failures earlier?
3. How can you manage the volume of alerts and distribute notification better?
4. How can you increase collaboration to diagnose and remediate faster?
5. How can you best pursue continual experimentation and learning?

Our ideas are guided by these principles:

- Understand system relationships
- Automate aggressively
- Collaboratepervasively

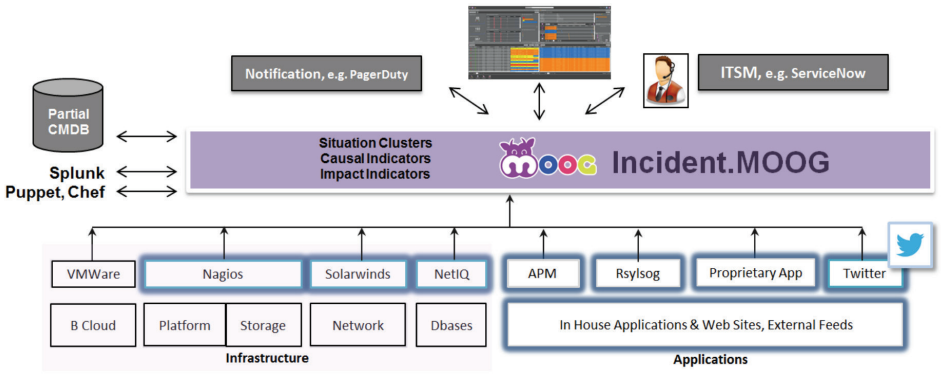
Moogsoft's perspective

Moogsoft works side by side with DevOps teams to eliminate roadblocks that might prevent them from achieving rapid delivery of innovation and business value.

Moogsoft AIOps is our product.

When you see the cheerful Moo Head, you'll know we're talking about Moogsoft AIOps.

Figure 1. Moogsoft AIOps's place in a DevOps monitoring architecture.



This white paper will show:

- How you can benefit from the convergence of monitoring across pre-production Dev/QA and production Ops
- What you can gain by converging visibility across your separate Dev and Ops tools – getting "real" data at the Dev stage**
- No matter how much progress you have made, there is still yet, a better way

**In regulated industries, getting real data at the Dev stage may be in conflict with "Controlled Access to Production Data" policies.

Question #1

What's your delivery (development to automated QA to production) lifecycle today, and what do you want it to be?

No matter how much you have streamlined your software delivery lifecycle, it is likely that you still want it to be faster and more efficient so you can scale up more easily. You have likely focused on automating code release and deployment. The next phase should focus on automating incident detection and response.

How can this be done?

Embrace experimentation. We will examine this in more detail in Question 5. But for now, here are several suggestions for removing roadblocks to embracing experimentation, and finding new ways to manage incidents:

- Detect failures earlier and notify the right people immediately
- Make it easier for teams to collaborate
- Immediately see the impact of builds/changes on application health and stability
- Converge Dev/QA and Ops with new-era monitoring; merge visibility across Dev and Ops tools into one

Before we can detect failures earlier, we must first:

UNDERSTAND ALL PARTS OF THE FACTORY

How can this be done?

- Ingest pertinent behavioral data across the entire application-service stack***
- Remove the noise so you can clearly see what's important
- Identify and cluster the data that is related

*** The entire application-service stack spans the applications to the infrastructure, and everything in between, such as new software infrastructures, clouds, and tools for automated release deployment and domain-specific monitoring.

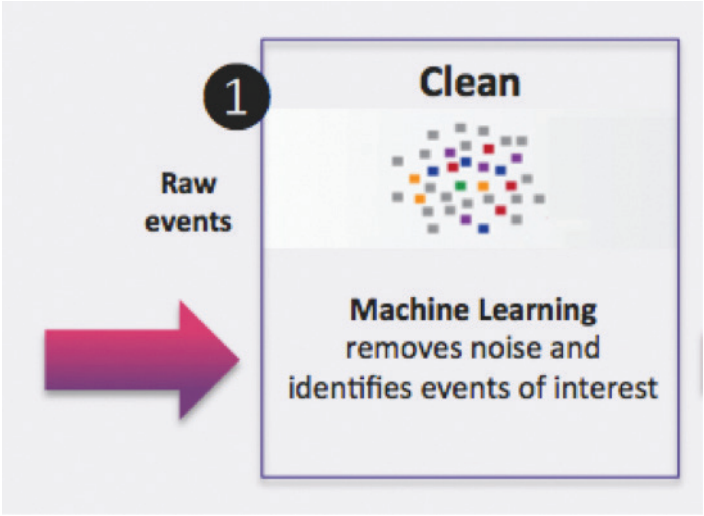
It may seem irrelevant to capture data that is external to an application. However, it is necessary if you want rapidly determine if a failure is caused by the software code or the underlying environment. (See Figure 2.)

Moogsoft AIOps ingests syslog and other log data, log4*, JSON, SNMP, XML, IMAP and other event data from all parts (all layers of the "stack") of the factory.
Once you have access to all the data you need to understand (and visualize) the entire ecosystem, you must clean it up so you can spot failures easily.

Figure 2. Moogsoft AIOps ingests data across the entire stack and the surrounding ecosystem so you can understand how all parts influence one another within the whole.

ELMINATE NOISE – FOCUS ON WHAT MATTERS

Figure 3. Moogsoft AIOps uses analytics to clean up (remove) the noise in real-time and reveal what's important.



When you are able to capture all pertinent behavioral data and eliminate the noise, you can begin to consider Question 2.

Question #2

How can you detect failures earlier?

When the application-service environment is small and not too complex, relying just on tribal knowledge works OK. As the environment grows large and complex, this ad-hoc approach becomes more difficult. When there are more moving parts and more dependencies (adding new layers, e.g. Docker containers) more things can go wrong.

This is where new-era adaptive monitoring comes into play.

Moogsoft AIOps monitors differently. It continually "learns" about the entire environment and adapts to change algorithmically. Moogsoft AIOps updates its learning in real-time.

LEARN FROM THE PAST BUT DO NOT BE BOUND BY IT

Today, when failures happen, you try to diagnosis by "drilling down" by running queries on log files. Drilling down is quite useful, assuming you know where to look and what you're looking for. But this is an after-the-fact approach that can be well after the problem is occurred. There is still yet, a better way.

DETECT FAILURES ACCURATELY AND IMMEDIATELY

Moogsoft AIOps monitors events in real-time and uses sophisticated machine learning to group related events into clusters (Situations). A Situation indicates that an incident has occurred (or is occurring), so you can make corrections immediately.

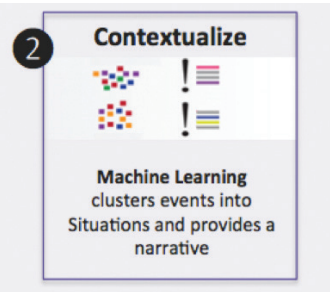


Figure 4. Moogsoft AIOps does much of the "drilling down" for you in real-time, by contextualizing data from across the entire environment to reveal all components involved in the service-affecting situation.

There is more that can be improved. That brings us to Question 3.

Question #3

How can you manage the volume of alarms and distribute notification better?

When you embrace experimentation, failures happen… and you receive alarms.

- When you get an alarm, do you know if it is truly important?
- If it is, are others downstream also affected, and do they send alarms?
- If other teams are also receiving alarms, do you know which alarms are related?
- If multiple teams are affected, which team(s) should be responding to the issue?

HOW DO YOU CORRELATE A FLOOD OF ALERTS?

Using rules and filters to sort alarms in the form of emails only goes so far. Alarm on-call escalation processes and tools (e.g. xMatters, PagerDuty) can help determine who should receive an alert but they lack any smarts to correlate across alarms.

There is still yet, a better way. Launch new-era collaboration when you are paged.

Moogsoft AIOps inserts easily into your current alert notification process and makes it work BETTER by adding a layer of intelligence BEFORE alarms are distributed. Moogsoft AIOps distributes sets of related alerts (Situations).

Moogsoft AIOps reduces the number of alarms, clustering related alerts into a single Situation. Moogsoft AIOps reduces notification down to 1 situation, not 100s of related alerts each generating its own notification. Moogsoft AIOps correlates event and alert data FROM ALL COMPONENTS IN THE STACK, either on-premise or in the cloud:

- Application services
- Application infrastructure (e.g. containers and virtualization services)
- DevOps tools (e.g. automated release deployment tools)
- Domain-specific monitoring tools (e.g. APM, log analyzers)
- Cloud services (public, private, hybrid)
- Customer sentiment (e.g. Twitter handles)

Correlation helps you diagnose probable cause of failures faster by exposing relationships between event data in all parts of the stack. This is going in the right direction, but to speed up remediation, you need to collaborate in real-time. This brings us to Question 4.

Question #4

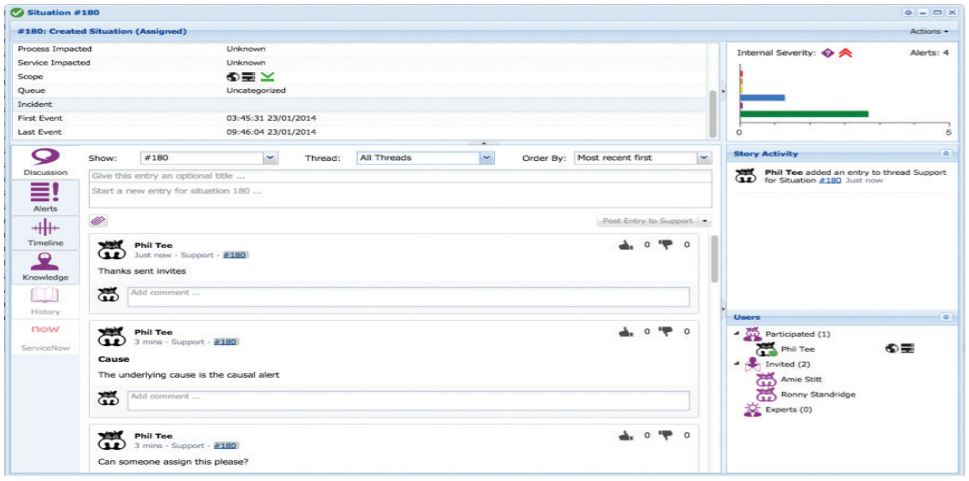
How can you collaborate to diagnose and remediate faster?

When failures happen, the DevOps team springs into action. Perhaps you use Slack, HipChat, or another IRC channel to get updates on what's going on. There is still yet, a better way.

THE SITUATION ROOM: A SMARTER PLACE TO MEET

When a failure unfold, Moogsoft AIOps notifies the right team to come together in an automatically created Situation Room, and arms them all with context and a shared workbench to remediate problems fast. A Situation

Figure 5. The Discussion Tab in a Situation Room is similar to a Facebook Wall.



Room is created for each failure.

A VIRTUAL WAR ROOM: BUILT-IN TRIBAL KNOWLEDGE

The Situation Room makes it easier to collaborate. Your team can:

- Share diagnostics and knowledge
- Understand what other groups are doing
- Access pre-existing knowledge about this type of anomaly
- Collaborate to restore service faster and initiate remediation

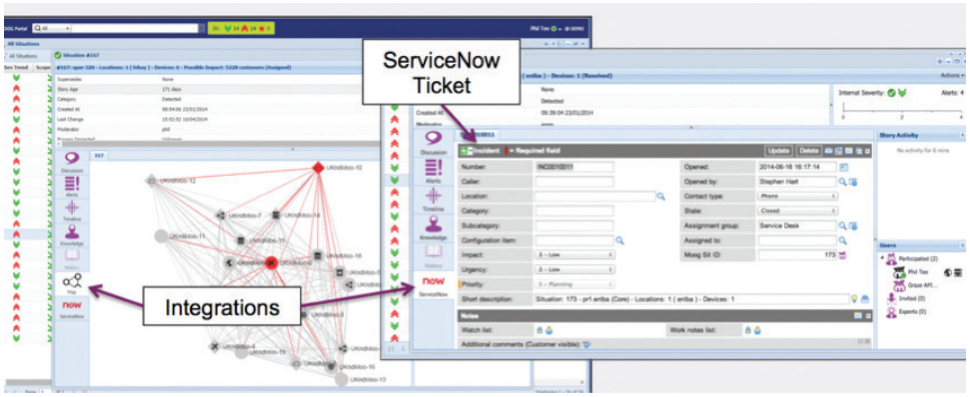
Moogsoft AIOps can send you a notification directly or act as a "smart" front-end to an on-call notification tool – you get a single page with a link to the Situation Room – eliminating the "Alert Fatigue" of receiving multiples notifications for the same Situation. But there's more.

"HOW DID THIS FAILURE HAPPEN?" SEE THE ANSWER

Figure 6. The Timeline Tab in the Situation Room shows the sequence of events to speed diagnosis and help you answer, "how did this happen?" Each color bar represents an event from the source data at a specific point in time. Moogsoft AIOps also has a Knowledge Tab that gives you immediate access to tribal knowledge that is archived automatically from online discussions in both previously closed and active Situations.

SPEED UP REMEDIATION: BRING YOUR FAVORITE TOOLS WITH YOU

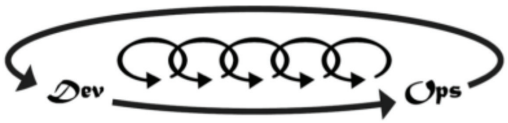
Figure 7. You can dynamically add links to third-party support tools (e.g. Jira, StackStorm, ServiceNow, etc.) and use the tools you love – all within the context of active Situations.



To bring all of these fresh takes on DevOps practices into full circle, let's take a look at Question 5.

Question #5

How Can You Best Pursue Continual Experimentation and Learning?



Fail boldly… fail in smaller doses… detect failures earlier… use new-era monitoring.

ACCELERATE PRODUCTION ROLL OUT: Merge Visibility Across Separate Dev and Ops Tools

In the beginning: There was the **HANDCRAFTED** Dev and Ops approach:

STEP 1: Dev delivered code to QA teams and Performance Labs who tested and blessed.

STEP 2: Ops, a completely different team with a completely different toolset, formally (and highly manually) deploys the app.

Two separate teams, two separate toolsets – each watching the same applications!

Now: Enter the **Cloud-Aware Application** approach:

Instrumentation and self-testing are baked in from the beginning, rather than bolted on later.

+ Best-Practices Automation
REPLACES
Traditional standalone UAT
WITH
A/B Testing

Cloud-aware applications expect failure and are able to self-heal with fail around. As a result, the monitoring "commodity" underlay becomes much less important.

Dev/QA is Being Transformed

Dev-initiated Unit Automated Tests (UAT) enable continuous integration. Note: many self-tests can also be used to watch the app in production, rather than re-writing tests in another toolset)

A/B tests call for introducing an updated version of the app into a small x% of the user base – rolling back quickly if not successful.

The combination of self-testing, self-healing / fail-around and A/B testing have all ushered in notable changes in Dev/QA and Ops monitoring.

The Cloud-Aware Era Requires New-Era Monitoring Tools

Moogsoft AIOps Moogsoft enables dramatic improvements in productivity during A/B testing when it is crucial to quickly assess and correlate potential failures.

The DevOps continuous integration engine allows for automation of the unit testing for all the core capabilities of an application.

AVOID THE COST OF SEPARATE TEAMS AND TOOLSETS

The tools you select should perform well across your full lifecycle, from Dev to Ops.

Moogsoft AIOps is designed to cover the entire lifecycle.

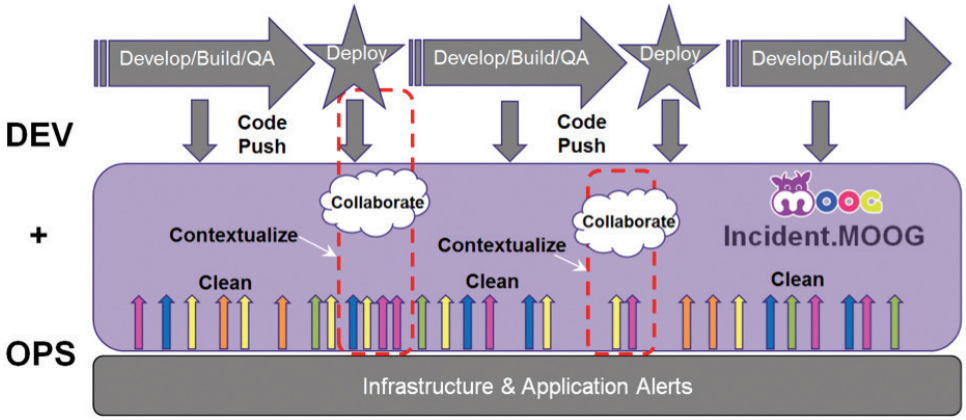
It is doubly cost-effective to upgrade to a new-era tool like Moogsoft AIOps.

1. You can replace the cost of two full toolsets with one
2. You can catch and prevent issues early in the lifecycle where it is orders of magnitude less costly to do so

Moogsoft AIOps consumes Log4* and blends it with infrastructure monitoring data so you gain insight into new application behavior during Dev/QA.

Moogsoft AIOps helps you assess the relevance and accuracy of application log messages in Dev/QA so you can fine-tune their effectiveness prior to production roll out.

Figure 8. Moogsoft AIOps exposes more defects in Dev/QA and reveals more about the consequences of interactions between the application and the underlying environment. This provides strong support for experimentation and learning.



Following this approach, using Moogsoft AIOps you can run straight on Production Stack more smoothly following Dev/QA.

HANDLE CODE COMMITS AND CHANGES BETTER

Moogsoft AIOps monitors in Dev/QA as if the application were in production so you can see the impact of changes in Dev/QA. For example:

- A change is deployed
- Moogsoft AIOps monitors the application logs to reveal changes in behavior between releases and highlight impacts that may not have been understood in development
- A release deployment tool (e.g. Puppet, Chef, Ansible) sends an release

update event and Moogsoft AIOps automatically relates the application change to a cluster of alarms relating to a service-affecting situation, so you can visualize the causal relationship of that change

AND THERE'S MORE...

For customer-facing applications that require continuous and rapid release updates, Moogsoft AIOps can ingest customer sentiment feedback from social media channels like Twitter, analyzing and correlating it with application-service performance data. In this way, Moogsoft AIOps is able to get an even earlier start to detecting failures and relating it to those that getting affected. Moogsoft AIOps can identify problems as they start to affect customers, and before incidents cascade into P1 outages.

Summary

Moogsoft AIOps can help you detect failures earlier, distribute alarm notification better, and collaborate easier to diagnose and remediate problems faster.

Moogsoft AIOps can help you see the impact of changes in pre-production (Dev/QA), so you anticipate some problems earlier and more confidently move into production.

Moogsoft AIOps can help you achieve convergence of monitoring across Dev/QA and production stages – and provide a way to merge visibility across your separate Dev and Ops tools.

All of this works together to help you more confidently and efficiently as you pursue continuous experimentation and learning.

Moogsoft AIOps helps DevOps teams move faster and achieve greater productivity in three significant areas:

A. **DEV/QA AND AUTOMATED PRE-PRODUCTION TESTING**. Moogsoft AIOps reveals changes in behavior between releases to highlight impacts that might not have been understood in development.

B. **ALARM AND INCIDENT NOTIFICATION.** Moogsoft AIOps inserts nicely and easily into current alert notification process and makes it all work better. You can eliminate "alert fatigue", without missing detection of complex service-affecting situations as they unfold.

C. **CORRELATION, DIAGNOSIS AND COLLABORATIVE REMEDIATION.** Moogsoft AIOps correlates event data pushed by all the components in your "stacks" (e.g. applications, cloud / data center services, release deployment tools), finds the relationships to understand failures and other service-affecting situations. This helps DevOps teams find probable cause faster and trigger remediation sooner.

Moogsoft AIOps makes the entire Dev – QA – Production cycle more efficient by consolidating tools sets and detecting problems faster. This melding of monitoring across Dev, QA and production is a necessary breakthrough for DevOps to evolve and scale.

For the latest in-product images, please visit www.moogsoft.com.

About Moogsoft

Moogsoft AIOps helps modern IT Operations and DevOps teams become smarter, faster, and more effective by providing technological supplementation that automates mundane tasks, enables scalability, and frees up human beings to do what they do best — ideate, create, and innovate. For more information, visit www.moogsoft.com.

USA (Headquarters)
1265 Battery St., SF, CA 94111
+1 415 738 2299

UK
River Reach 31-35 High Street,
Kingston Upon Thames, KT1 1LF
+44 203 743 8748