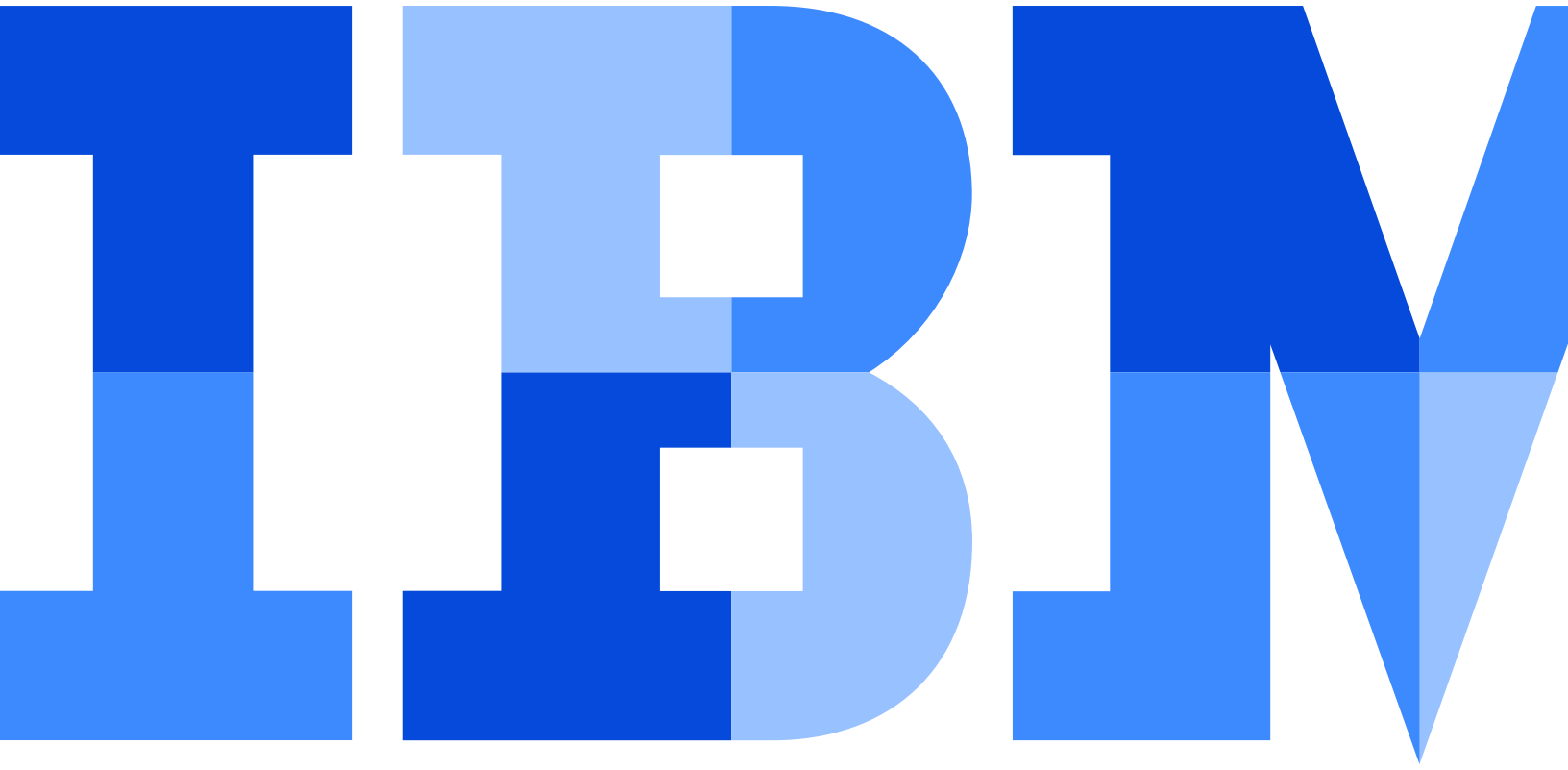
IBM Cloud Event Management
White Paper

February 2018

To put the Ops in DevOps, shift event management to the left



IBM

Operations professionals who are in the early stages of embracing DevOps methodologies face some significant challenges. Some of these challenges have been common for decades, while others are more unique to DevOps situations. It's not new for business stakeholders to apply increasing pressure to achieve key business and efficiency goals. However, the continuous integration and continuous delivery that are typical in DevOps methodologies bring added pressure: release cycles that will be measured by value — delivered in days (or even hours) rather than weeks or months.

Development teams have embraced agile practices and produce production-ready code in short iteration. They make the most of cloud technologies to provide auto-scaling microservice architectures at unprecedented levels of scale and modularity. Dev and Ops teammates must work together to implement an integrated delivery toolchain — one that can build, test and deploy a new release version at the touch of a button. All the while, expectations and criticality have never been higher; organizational leaders expect immediate production readiness, ongoing "five nine availability" (99.999% availability) and high speed performance. Whether your organization is in the beginning DevOps stages or whether you're further along the transformation path, the business objectives and measurements have not really changed. The KPIs are still mean time to restore service, mean time to repair, mean time to fix failures and pressure on costs.

Dev teams commonly produce highly instrumented code that provides a dense trail of data that relates to the state of the applications. These applications are deployed in a way that inherently produces high volumes of measurements and status information. This information includes runtimes and middleware that themselves constantly report their own state. The runtimes, in turn, will be underpinned by physical and virtual compute, along with network and storage infrastructure that also is monitored with tools that provide a continuous stream of state information. Operations are accustomed to high event volumes, but variety and volume are increasing commensurately with the complexity of these systems.

These challenges have led to an evolving role that Ops play. Luckily for Ops professionals, this evolutionary progression is far from revolutionary. The evolving role of Ops will lead to a positive direction by building on best practices and time-tested processes — rather than scrapping them and starting from scratch. Within a mature enterprise that has invested highly in IT, disciplines such as Event and Incident Management depend upon a key set of capabilities that are equally valuable throughout the transition to DevOps.

The proximity of developers to operations (and the cohesion of operations around those teams) are enhanced by the proven and time-tested methodologies. Integrated tools need to consume events from highly heterogeneous environments, minimizing the amount of management "noise" that is presented to the people handling the events. In most situations, actual humans are still required to respond to events. Limiting this human activity to issues that truly deserve a human touch can be achieved with runbook automation, analyzing historical trends and machine learning.

Event and Incident Management tools are even more critical than they ever were as teams shift to practices such as DevOps and Site Reliability Engineering. A common practice is to shift from a large number of lower-skilled people to a smaller number of substantially highly skilled engineers. These highly skilled Ops professionals need to overcome on-going time constraints by reducing the "noise" generated by the managed systems. They must increase adoption of automation and work closely with Dev to prepare for continuous roll outs. These highly skilled Ops professionals don't have time to sift through large files and performance metrics just to restore a service.

Shift right Dev, shift left Ops

The DevOps transformation that's presented to Operations and Development teams is analogous to, and has the potential of, the widespread adoption of Agile Software Development practices just a few years ago. Prior to this, there was often a "brick wall" between coders and software Quality Assurance (QA) professionals. These days, developers and QA professionals consistently work closely together. The analogous "brick wall" between modern Development and Operations groups is slowly tumbling down.

It is becoming more common to see "shift right Dev" as well as "shift left Ops", where the quality of a running service is a shared responsibility. This type of overlap can pay dividends in efficiency, especially when the teams use a common set of tools. For example, making it easy for a developer to author an automatic response, or a runbook, or a suppression rule means that an operator is less likely to escalate an incident. Separating the signal of service status from the "noise" of service state will make all the difference when you need to know quickly that there is a problem with an application or service before users notice — or when you need to alert them about it. This approach may also prevent Ops teams from unnecessarily responding to a page indicating a benign application state.

All of which reduces the probability of anyone being awakened at 3 AM!

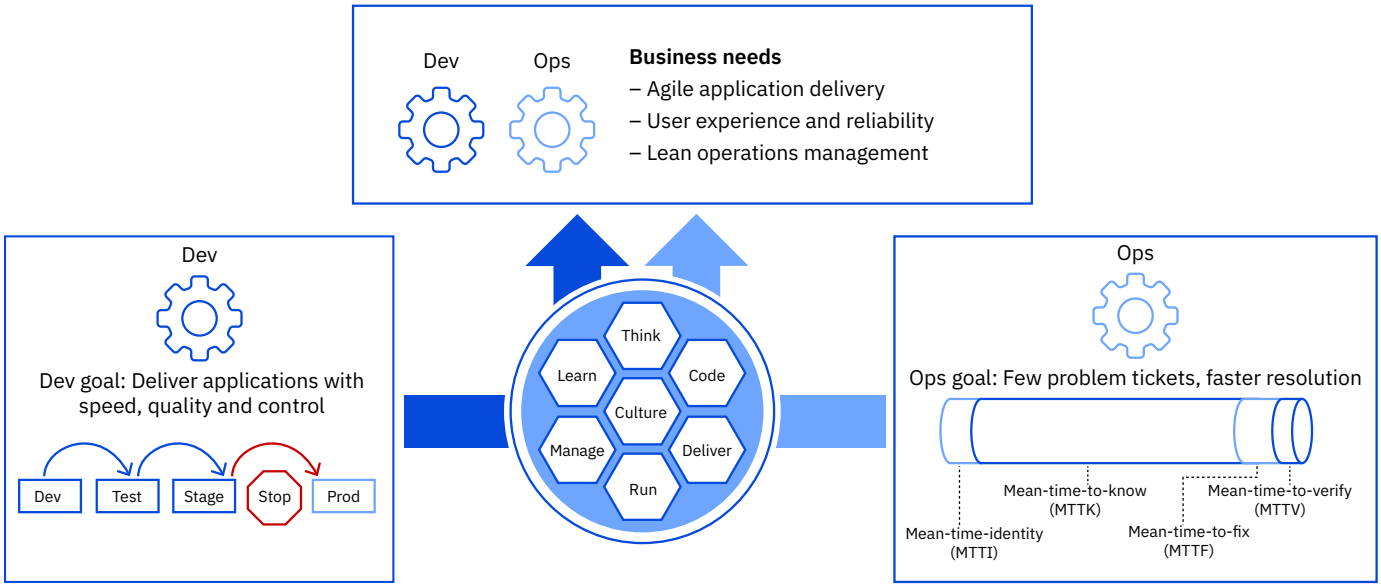


Figure 1: Shift Dev right and shift Ops left to deliver applications with speed, create a reliable end-user experience, and support lean operations management.

Event and Incident Management processes, supported by the right tools, are critical in empowering DevOps teams to achieve these aims. They need tools that can:

- Filter out events that are not likely to be service-affecting. A trace message from an app log is not worth anyone's attention, unless they need to view it as a part of a diagnostic procedure. A synthetic user transaction timeout may require immediate investigation.
- Correlate events that are symptoms of the same cause, so that there is one notification per true incident, rather than one for each symptom event.
- Enrich events with context, so that if a single service instance fails in a redundant array of, say, five instances, the event gets routed to the operations console — but people are not awakened unless the service is affected.
- Implement X-in-Y policies. In a large and complex system, a single microservice-to-microservice HTTP timeout may not be a huge problem. But you may want to investigate if you start getting 20 timeouts per second.
- Collaborate with development on the implementation and definition of runbooks for common failures. Tie those runbooks to incidents as they occur, so that the first responder has a process that they can follow to reinstate a service or prevent an outage in the first place. And, if possible, find a way to automate those runbooks. If the problem is a failed process, then restart it. A disk is full? Free up some space.
- Gain insights from the reams of data your tool collects by the application of artificial intelligence and machine learning techniques. Apply that insight to streamline the incident resolution and identification process so that you can maximize operational efficiency.

With Dev and Ops working together moving forward, organizations can continue to build on what Ops has built over the last few decades. The common DevOps challenges can be embraced and turned into catalysts for improvement and innovation. Institute the right tools and platforms to foster more-productive use of people's time, and make the most of automated knowledge sharing and collaborative problem-solving. These actions can turn the potential benefits of DevOps methodologies into an achievable reality.

To find out how to restore service disruptions promptly, prepare for the operational support of continuous application rollouts into production, and find ways to drive improved efficiency over time, explore IBM Cloud Event Management.

For more information

To learn more about IBM Cloud Event Management, please contact your IBM representative, or visit the IBM Cloud Event Management website: **ibm.com**/cloud/event-management.

IBM Corporation
New Orchard Road
Armonk, NY 10504

Produced in the United States of America
February 2018

06013506-USEN-00