RELEASE MANAGEMENT IN DEVOPS

The rise in popularity of DevOps practices and tools comes as no surprise to those who have already utilize the new techniques centered around maximizing the efficiency of software enterprises. Similar to the way Agile quickly proved its capabilities, DevOps has taken its cue from that and built off of Agile to create tools and techniques that help organizations adapt to the rapid pace of development today's customers have come to expect.

As DevOps is an extension of Agile methodology, DevOps itself calls for extension beyond its basic form as well.

Collaboration between development and operations team members in an Agile work environment is a core DevOps concept, but there is an assortment of tools that fall under the purview of DevOps that empower your teams to:

- Maximize their efficiency
- Increase the speed of development
- Improve the quality of your products

DevOps is both a set of tools and practices as well as a mentality of collaboration and communication. Tools built for DevOps teams are tools meant to enhance communication capabilities and create improved information visibility throughout the organization.

DevOps specifically looks to increase the frequency of updates by reducing the scope of changes being made. Focusing on smaller tasks at a time allows for teams to dedicate their attention to truly fixing an issue or adding robust functionality without stretching themselves thin across multiple tasks.

This means DevOps practices provide faster updates that also tend to be much more successful. Not only does the increased rate of change please customers as they can consistently see the product getting better over time, but it also trains DevOps teams to get better at making, testing, and deploying those changes. Over time, as teams adapt to the new formula, the rate of change becomes:

- Faster
- More efficient
- More reliable

In addition to new tools and techniques being created, older roles and systems are also finding themselves in need of revamping to fit into these new structures. Release management is one of those roles that has found the need to change in response to the new world DevOps has heralded.

(This article is part of our DevOps Guide. Use the right-hand menu to navigate.)

What is Release Management?

Release management is the process of overseeing the planning, scheduling, and controlling of software builds throughout each stage of development and across various environments. Release management typically included the testing and deployment of software releases as well.

Release management has had an important role in the software development lifecycle since before it was known as release management. Deciding when and how to release updates was its own unique problem even when software saw physical disc releases with updates occurring as seldom as every few years.

Now that most software has moved from hard and fast release dates to the software as a service (SaaS) business model, release management has become a constant process that works alongside development. This is especially true for businesses that have converted to utilizing continuous delivery pipelines that see new releases occurring at blistering rates. DevOps now plays a large role in many of the duties that were originally considered to be under the purview of release management roles; however, DevOps has not resulted in the obsolescence of release management.

Advantages of Release Management for DevOps

With the transition to DevOps practices, deployment duties have shifted onto the shoulders of the DevOps teams. This doesn't remove the need for release management; instead, it modifies the data points that matter most to the new role release management performs.

Release management acts as a method for filling the data gap in DevOps. The planning of implementation and rollback safety nets is part of the DevOps world, but release management still needs to keep tabs on applications, its components, and the promotion schedule as part of change orders. The key to managing software releases in a way that keeps pace with DevOps deployment schedules is through automated management tools.

Aligning business & IT goals

The modern business is under more pressure than ever to continuously deliver new features and boost their value to customers. Buyers have come to expect that their software evolves and

continues to develop innovative ways to meet their needs. Businesses create an outside perspective to glean insights into their customer needs. However, IT has to have an inside perspective to develop these features.

Release management provides a critical bridge between these two gaps in perspective. It coordinates between IT work and business goals to maximize the success of each release. Release management balances customer desires with development work to deliver the greatest value to users.

(Learn more about IT/business alignment.)

Minimizes organizational risk

Software products contain millions of interconnected parts that create an enormous risk of failure. Users are often affected differently by bugs depending on their other software, applications, and tools. Plus, faster deployments to production increase the overall risk that faulty code and bugs slip through the cracks.

Release management minimizes the risk of failure by employing various strategies. Testing and governance can catch critical faulty sections of code before they reach the customer. Deployment plans ensure there are enough team members and resources to address any potential issues before affecting users. All dependencies between the millions of interconnected parts are recognized and understood.

Direct accelerating change

Release management is foundational to the discipline and skill of continuously producing enterprise-quality software. The rate of software delivery continues to accelerate and is unlikely to slow down anytime soon. The speed of changes makes release management more necessary than ever.

The move towards CI/CD and increases in automation ensure that the acceleration will only increase. However, it also means increased risk, unmet governance requirements, and potential disorder. Release management helps promote a culture of excellence to scale DevOps to an organizational level.

Release management best practices

As DevOps increases and changes accelerate, it is critical to have best practices in place to ensure that it moves as quickly as possible. Well-refined processes enable DevOps teams to more effectively and efficiently. Some best practices to improve your processes include:

Define clear criteria for success

Well-defined requirements in releases and testing will create more dependable releases. Everyone should clearly understand when things are actually ready to ship.

Well-defined means that the criteria cannot be subjective. Any subjective criteria will keep you from learning from mistakes and refining your release management process to identify what works best. It also needs to be defined for every team member. Release managers, quality supervisors, product

vendors, and product owners must all have an agreed-upon set of criteria before starting a project.

Minimize downtime

DevOps is about creating an ideal customer experience. Likewise, the goal of release management is to minimize the amount of disruption that customers feel with updates.

Strive to consistently reduce customer impact and downtime with active monitoring, proactive testing, and real-time collaborative alerts that enable you to quickly notify you of issues during a release. A good release manager will be able to identify any problems before the customer.

The team can resolve incidents quickly and experience a successful release when proactive efforts are combined with a collaborative response plan.

Optimize your staging environment

The staging environment requires constant upkeep. Maintaining an environment that is as close as possible to your production one ensures smoother and more successful releases. From QA to product owners, the whole team must maintain the staging environment by running tests and combing through staging to find potential issues with deployment. Identifying problems in staging before deploying to production is only possible with the right staging environment.

Maintaining a staging environment that is as close as possible to production will enable DevOps teams to confirm that all releases will meet acceptance criteria more quickly.

Strive for immutable

Whenever possible, aim to create new updates as opposed to modifying new ones. Immutable programming drives teams to build entirely new configurations instead of changing existing structures. These new updates reduce the risk of bugs and errors that typically happen when modifying current configurations.

The inherently reliable releases will result in more satisfied customers and employees.

Keep detailed records

Good records management on any release/deployment artifacts is critical. From release notes to binaries to compilation of known errors, records are vital for reproducing entire sets of assets. In most cases, tacit knowledge is required.

Focus on the team

Well-defined and implemented DevOps procedures will usually create a more effective release management structure. They enable best practices for testing and cooperation during the complete delivery lifecycle.

Although automation is a critical aspect of DevOps and release management, it aims to enhance team productivity. The more that release management and DevOps focus on decreasing human error and improving operational efficiency, the more they'll start to quickly release dependable services.

Automation & release management tools

Release managers working with continuous delivery pipeline systems can quickly become overwhelmed by the volume of work necessary to keep up with deployment schedules. This means enterprises are left with the options of either hiring more release management staff or employing automated release management tools. Not only is staff the more expensive option in most cases but adding more chefs in the kitchen is not always the greatest way to get dinner ready faster. More hands working in the process creates more opportunities for miscommunication and over-complication.

Automated release management tools provide end-to-end visibility for tracking application development, quality assurance, and production from a central hub. Release managers can monitor how everything within the system fits together which provides a deeper insight into the changes made and the reasons behind them. This empowers collaboration by providing everyone with detailed updates on the software's position in the current lifecycle which allows for the constant improvement of processes. The strength of automated release management tools is in their visibility and usability—many of which can be accessed through web-based portals.

Powerful release management tools make use of smart automation that ensures continuous integration which enhances the efficiency of continuous delivery pipelines. This allows for the steady deployment of stable and complex applications. Utilizing intuitive web-based interfaces provides enterprises with tools for centralized management and troubleshooting that helps them plan and coordinate deployments across multiple teams and environments. The ability to create a single application package and deploy it across multiple environments from one location expedites the processes involved in continuous delivery pipelines and makes the management of them much more simplified.

Related reading

- BMC DevOps Blog
- DevOps Metrics for Optimizing CI/CD Pipelines
- Containers & DevOps: Containers Fit in DevOps Delivery Pipelines
- ITIL® Release and Deployment Management
- Today's Top Trends in Software Development