**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](https://www.bmc.com/blogs/saas-vs-paas-vs-iaas-whats-the-difference-and-how-to-choose/) (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](https://www.bmc.com/blogs/what-is-ci-cd/) 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.

**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](https://www.bmc.com/blogs/what-is-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](https://www.bmc.com/blogs/product-owner-product-manager-scrum-master/), 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.