Perfecting the Application Release Process and How to Adapt the Modern Way

DataDrone

Giving Data, The Wings!

White Paper

An Overview of Application Release Process

Application release is part of the overall software release management process within the organization, and it encompasses the delivery of the final version of the software product after DevOps' enhancement and bug fixes. In this case, the final version of the released software doesn't necessarily mean the endpoint of the software version control. So, application release is part of what is known as the Continuous Delivery (CD) process, which is an ongoing practice by the DevOps team to build, test, and deliver improvements to the software and its user environment.

The Relationship Between Application Release and Continuous Delivery

Modern software and application development processes tend to leverage agile and DevOps practices to fasten software release and also effectively ensure uninterrupted continuous delivery of updates, improvements, patches, and bug fixes to the application. When the application is released to the end-user, its developers and engineers keep researching on how to improve it in one way or the other so that user experience is enhanced to the maximum. This whole process turns out to be cyclical, forming the application release life cycle. In other words, Continuous delivery fastens up the application release process by catalysing a cycle of continuous changes and improvements to the final software release.

Understanding the Application Release Life Cycle.

The application Release Life Cycle forms a series of stages implemented by the DevOps team to optimize application quality and better user experience. There are a couple of reasons why modern DevOps teams adhere to the application release lifecycle;

- Optimize quality and enhance User-experience: These days, if you're going to build better software, you need to prioritize quality and usability. Many users don't want buggy applications that keep breaking. So, the intervention of an application release lifecycle helps DevOps teams to release an application through a series of well-calculated stages to measure and analyse users 'reactions and feedback regarding the overall performance of the released version.
- It helps to fine-tune the application business model. Most software developed on the market is backed up with a business model that must achieve a desirable return on investment (ROI). Some software can be a total failure because of poor design and lack of desired features needed by the users. Whenever users refuse to embrace the application's design, it could drastically affect its entire business model. So, its release life cycle could be the best way to tease users first and evaluate its performance in the market.
- It helps to avoid costly future breakdowns in the production environment. Application breaks down because of unfixed bugs or other technical complications. They can lead to downtime issues which could spark frustration from the users and affect the overall brand reputation. In Critical and enterprise-level applications, downtime or breakdown cannot be an option, and users do not like to use a system that constantly faces interruptions. This is where the release life cycle plays a warning radar role to help the DevOps team foresee the future performance issues of the application in the live production environment.

The application release lifecycle encompasses a 5-stage framework which includes;

1. **Pre-Alpha Stage:** This involves activities that are implemented before the application testing stage is achieved. Some of these activities include; analysing requirements, application designing, and development. The following stages in the application release life cycle sequence depend on the activities implemented in this stage in order to complete every release loop cycle.
2. **Alpha Stage:** In this stage, the application testing is done internally by the DevOps team and its external availability is limited. Alpha software versions usually contain serious bugs and errors, and this is why developers reserve the testing to themselves before they release the beta version to the public.
3. **Beta Stage:** In this stage, an application could be released but may still contain some errors and bugs. The beta application version is usually unreliable as it could cause performance issues, crashes, and data losses for the user. Application developers always warn users about the possible breakdown of the beta application release, so their expectations are balanced.
4. **Release Candidate:** This is basically the beta version of the application with the potential to be the stable release. Unless there are significant bugs or errors identified in this release, there is potential that it could serve as a final product that could be launched.
5. **Stable Release:** After resolving all the major bugs and performance issues in the first stages, a stable release is launched for the final users to use. If bugs or errors are present in this stage, they're considered minor.

Perfecting the Application Release Process and Possible Bottlenecks.

As we've seen, the application release process encompasses a series of sequential stages with the primary goal of perfecting the overall performance and usability of the product. Some of the major activities involved in perfecting an application release include;

- Hunting for bugs & errors and fixing them as soon as possible.
- Optimizing performance issues like speed and user interface enhancement.
- Testing and investigating security vulnerabilities and making all the necessary patches as a prerequisite before official release.
- Evaluating the business model of the application before it goes to the market to avoid affecting return on investment.
- Mitigate all other pitfalls and errors before the application is officially launched.

DevOps teams find themselves caught up in making sure the final product offers the best user experience as expected. It is a daunting task, but developers can't sideline their role in perfecting the final release so it could suit the expectations of the users.

Assessing the challenges of perfecting application releases

No doubt, there are various challenges in the implementation of application releases and some of which include:

- **Lack of better collaboration:** In an enterprise-level setting, collaboration is part of the agile and DevOps software development practices. Without better collaboration, it would be challenging to fulfill specific goals as far as application release implementation and management is concerned.

- **Increased workload:** The various stages involved in the application release process make it so daunting to implement. Planning a release of applications would need a huge DevOps team to do all the necessary troubleshooting of identified bugs and errors. The situation is even worse if the troubleshooting process depends on manual methods that are tiresome.

- **Increased risks of failing to beat the project deadline:** Software and application testing is full of unpredictability. Developers cannot be 100 percent certain when to achieve the most acceptable stable release version of the application. Depending on the size of the application, if the DevOps team really goes the traditional or old-fashioned way, there are more chances of failing to beat the actual deadline of the project.

Application Release Automation and Dealing with the Challenges.

Traditional Application Release versus Automation; the Modern way

Traditionally, application releases were implemented manually, and developers had to spend 100s of hours, days, weeks, or months trying to perfect the final release of the product. It worked in such a way that the three critical components of the application or system life cycle, i.e., development, testing, and deployment teams, worked independently of the other without collaboration. This old-fashioned way proved to be counter-productive, daunting, ineffective, and unreliable, especially when working in an enterprise setting.

Traditional application release management did not embody collaboration, a crucial part of the modern DevOps model that sums up the entire application life cycle (development, testing, and deployment) into one umbrella.

With these overwhelming challenges, automation became the best solution and turned out to be an integral part of modern DevOps operations. Today, there are various application release automation (ARA) tools that are used to combine automation, facilitate workflow management and ensure environment modelling.

Major companies like HP, Microsoft, Amazon (AWS), Google, Oracle, and many others either use in-house proprietary or third-party application release automation platforms integrated with their software development and management processes.

The truth is that you don’t have to be like Google or Amazon to adopt application release automation and integrate it into the DevOps operations of your organization’s software development processes.

Why Application Release Automation?

The enormous benefits of application release automation over the traditional old-fashioned way are unchallengeable. As we've just seen, even major industry players endorse these benefits as they fasten up their DevOps processes and operations.
If you have some doubts, here is why application release automation (ARA) could tremendously alter the rules of the game in the DevOps management of your organization;

- **Speeds up the Continuous Delivery process.** The automated framework of the ARA tools makes it easier for the DevOps team to continuously deliver software as part of the development and release lifecycle. Forget the manual way of updating everything because automation can take care of everything during the continuous delivery process.
- **Collaboration is made easy.** Gone are the days when the critical components (development, testing & deployment) of the application lifecycle were independent of one another. Application release automation is a widely acceptable practice in DevOps and collaboration is an integral part of it.
- **Effective releases have never been more accessible**. ARA tools eliminate logistical challenges posed by the traditional method of application release. Pipeline optimization, task assignments, and visual workflow mapping make almost everything easier and faster.
- **Costs are cut short**. In the traditional application release process, there is a need for more workforce to deal with daunting tasks, which often involve manually fixing bugs and errors. But automation eliminates this need by simplifying how to handle troubleshooting of the problems even with a limited workforce.

DataDrone Database Release Automation Platform

Both application releases and database releases virtually share similar needs and challenges that could only be solved by automation. In addition, there is no way you can separate databases from application development, testing, and deployment. They're just an integral part, and so they share similar challenges during architecting and deployment.

We use databases during application development and deployment to hold critical data. Without databases, application delivery would be impossible, more so if they're enterprise applications. The biggest threats to application releases and deployment are related to database breakdowns.

Databases are delicate, and so, any slight error in their scripts could break down the entire application.

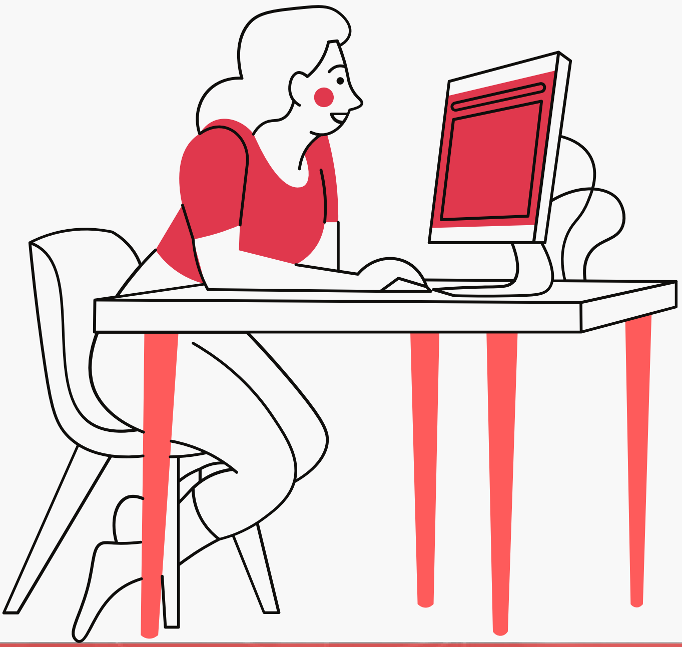
Usually, huge applications have more enormous databases, and they pose a significant challenge to manually or traditionally working on them when fixing script bugs or errors.

What is DataDrone Database Release Automation?

Having understood the challenges faced by database deployment, DataDrone automation solves the puzzle and takes your DevOps operation to a whole new level. You can improve your database release cycle by 34% through a cost-friendly pricing model. We offer a database-as-a-Service model to allow our customers to accomplish and implement the following functionalities with ease;

- Implement version control.
- Automation of releases without the hustle
- Continuous Delivery Process
- Faster Migrations

Contact us **today**, and we will talk about the best way to manage your databases and avoid breaking your application.



DataDrone
Giving Data, The Wings!

About us

DataDrone is committed to delivering a fully automated, rapid database deployment solution. We aim at simplifying the complexes, taking pride in saving every minute contributing to the release, resolving for all the database related bottlenecks.

Contact us

https://datadrone.it/contact/

https://www.datadrone.it/

info@datadrone.it

1800-506-6854