The ABCs of continuous release and deploy in a DevOps approach

Contents

Adapting and responding quickly to business needs is dramatically changing how IT teams operate. Teams need to be responsive to business requirements, enabling companies to gain ground in their market and keep up with changes in technology. Both established and upstart companies are competing for the same space. The more established company may have a higher budget, larger staff, and a more recognizable name; however, the upstart company may be flexible enough to deploy applications daily, which may include bug fixes, new features, and timely responses to business needs. If the more established company is not able to deliver software quickly and minimize risks, then it may lose that market.

Examples:

Amazon disrupted the market with web-based technology, then did it again with the Kindle. Craigslist out maneuvered traditional print classified advertising, putting newspaper companies at risk or, in some cases, out of business.

In this paper, we focus on the bottlenecks in enterprise software delivery caused by the handoff from development to operations. Typically, the operations team struggles to deliver applications at the pace that the development team creates them. Agile practices have enabled development teams to quickly create change. But manual processes, poor communication between developers and operations, and inconsistencies between processes and tools prove too error-prone for operations to

deploy safely every release to production (sometimes even to test, UAT, etc.). As a result, applications with interdependencies pile up, and organizations are not able to keep up with business needs, market trends, and customer demands. Companies need a way to deploy changes at an accelerated pace while maintaining stability.

Adopting DevOps

DevOps is an enterprise capability for continuous software delivery that enables organizations to seize market opportunities and reduce time to customer feedback. The most forward-thinking companies are establishing a DevOps approach in order to reduce time to customer feedback, improve governance, and balance quality and cost.

They can do this with greater lean and agile processes that expand across a fully automated lifecycle, which includes customers, LOB/business owners, development, test, and operations teams, all focused on achieving:

- Accelerated software delivery
- Reduced time to gather and analyze customer feedback
- Improved governance across the lifecycle
- Retaining the balance of quality, cost, and speed

A full adoption of DevOps allows teams to create releases quickly and deploy these updates more frequently. To achieve the benefits of Agile and keep up with market trends, businesses need to adopt a DevOps approach. But that requires organizations to make changes that extend across the people, processes, and tools involved in the application's lifecycle.

There are six "adoption paths" IBM advocates as a way to get started with DevOps. These practices allow you to focus on your immediate needs, then grow from there. These adoption paths are:

- Continuous business planning
- Collaborative development
- Continuous testing
- Continuous release and deployment
- Continuous monitoring
- Continuous optimization

The remainder of this paper will focus mainly on "continuous release and deployment," although these adoption paths often cross each other as teams work together. For more information on the complete view of DevOps, see the IBM paper titled "*DevOps: The IBM approach*".

Align teams

Teamwork is vital to making continuous release and deployment work. Traditionally, a "wall" separated the development and operations teams. Due to conflicting professional responsibilities, teams often did not truly understand each other's work. For instance, a development team's responsibilities include implementing changes to applications and adding features. Operations teams are responsible for keeping the system running and deploying applications to production.

In order for both teams to do their jobs effectively, they must communicate early in the application lifecycle to achieve an overall goal, which is to meet business demands. Developers need to inform operations of new applications in development and involve them early in architecture decision-making.

Developers should also include health and monitoring statuses of the applications. When developers inform operations teams early, operations teams can be prepared for the deployment when the time comes.

An alignment example

A large financial services company based in the United States knew they wanted to implement a DevOps approach in their software delivery and had automated tasks. However, applications were still failing when they were deployed to production. The company soon realized that tensions between the development and operations teams were a major factor in the failure. When the tensions were resolved and teams were aligned with the same overall goal to release and deploy continuously, they were able to make further improvements to achieve the benefits of agile.

Operations teams also need to communicate with development teams and share network diagrams. Operations must also support teams with disposable test environments and collaborate with QA to implement smoke tests for all environments. To break down the wall between development and operations, teams must trust each other and understand each other's professional responsibilities. Aligning the development and operations teams with the same overall goal is a good framework for adopting a continuous release and deployment practice. Once this is done, the process can be configured to meet both teams' requirements.

Build a consistent process

Inconsistencies and environmental differences result in release failures and costly downtime, but a carefully planned deployment process can ensure that releases run smoothly. A successful adoption will consist of a holistic process with end-to-end visibility. To ensure visibility across environments, stakeholders should share a single promotion process across the software delivery lifecycle. If the same deployment process is used in all environments, it will have been used dozens of times before it actually runs in production, and thus will be a thoroughly tested process. By offering a common process, operations teams can control it and maintain integrity of the build.

The easiest way to create a holistic process with end-to-end visibility is to examine the process in the production environment. The reason teams should start with the production environment is that it is where a successful deployment matters most. Once teams design the process for the production environment, teams should use the same process in earlier environments in the Software Development Lifecycle (SDLC). Doing so will allow teams to practice the process well before it is performed in production.

A consistent process example

A large dot-com company was experiencing a high rate of deployment failure. While automation helped, a change in the overall process was critical. To drive errors out, they had to use each failed deployment as a prompt to stop, fix the automation, and execute the automation. The test machine was out a bit longer in the short term, but availability improved greatly in the medium term. Over a few months, failure patterns were eliminated and the failure rate dropped by over 95 percent.

After creating a consistent process, teams should meet regularly to refine the process. Discuss what is going well and what is not going well, and then adjust the process. Perform this review and adjustment with each production release, after any unplanned outages, or in step with development sprints. Teams should consider automating steps in the process that create the most pains. Once the process is well defined, it can remove errors, improve consistency, and reduce duplication of effort.

Create an automated culture

Automation is an important practice for companies to utilize, not only for DevOps adoption, but also for business best practices. Automation can provide operations teams with a visible, reportable, and secure means of deploying applications to production environments. In addition, automation removes the possibility of human error in the deployment.

Manual and half-scripted steps invite human error, causing release failures and potential outages. Because of these risks, operations teams are reluctant to introduce change. Automation minimizes the possibility of human error, lowering the risk involved with frequent deployments. In turn, operations teams become more willing to deploy changes, thus enabling IT teams to respond to business demands.

There are limits to the amount deployments a team of release engineers can do at once. Applying the principle of automation relieves this bottleneck of human capacity, allowing for more consistency and speed in the continuous release and deployment practice. Teams stop being a deployment bottleneck when they are simply initiating and monitoring the deployments.

An automated culture example

A US-based entertainment company shifted from twice-weekly release windows, which had to be reserved far in advance, to simple on-demand production deployments by implementing an automated solution. By using automation, the release team stopped being a bottleneck, and deployments were better aligned with the business demands. As a result, the business was able to keep up with market trends and customer demands.

An added benefit of automation is the amount of visibility it offers. Without automation, companies rely on release notes and verbal records to find information about deployments. Automation helps organizations track information during deployments for greater traceability. This provides feedback for what has been deployed, when, and by whom.

Automating application deployments enables operations teams to version their deployments, which provides a safety net for operations. If a deployment fails, operations can automatically roll back the application to a previous version.

With automation, companies can reduce or eliminate error-prone manual steps and add visibility into the process. In addition, automation provides a solid foundation for further improvements.

Conclusion

The continuous release and deployment practice within DevOps addresses existing problems in traditional software development, such as:

- Conflicts between development and operations
- Processes that do not scale to the complexity of applications
- Teams using different tools across the SDLC

And it offers organizations the following clear benefits:

- Speed time to market
- Stable and predictable releases
- Increased visibility
- Fewer outages

IT teams are rapidly adopting a DevOps approach as business needs and market trends push for quicker response time. While fully adopting DevOps may take time, implementing the above practices can help eliminate risks, speed time to market, and keep up with business demands and market trends.

IBM Release and Deploy Products

IBM's suite of continuous release and deployment solutions provide visibility and control of the entire lifecycle. From build to production, we deliver solutions and practices adjustable to your unique processes. Read more about the tools here: **ibm.com**/ibm/devops/us/en/deploy

- **IBM UrbanCode Build** is an automated build solution providing enterprise scalability, capable of managing builds from simple to the most complex. With UrbanCode Build, project templates provide local control, while enforcing enterprise policies (e.g., security scans, test coverage, and approved dependencies).
- **IBM UrbanCode Deploy** orchestrates and automates the deployment of applications, middleware configuration, and database changes into development, test and production environments, accelerating time to market, driving down cost while reducing risk.
- **IBM UrbanCode Release** transforms error-prone and chaotic release planning into streamlined release events, replacing spreadsheets with a collaborative solution that eliminates breakdowns in communication enabling more frequent releases at lower risk.

For more information

To learn more about DevOps solutions from IBM, please contact your IBM representative or IBM Business Partner, or visit the following website: **ibm.com**/software/rational

Additionally, IBM Global Financing can help you acquire the software capabilities that your business needs in the most cost-effective and strategic way possible. We'll partner with credit-qualified clients to customize a financing solution to suit your business and development goals, enable effective cash management, and improve your total cost of ownership. Fund your critical IT investment and propel your business forward with IBM Global Financing. For more information, visit: **ibm.com**/financing

IBM Corporation
Software Group
Route 100
Somers, NY 10589

Produced in the United States of America
May 2013

Please Recycle

RAW14324-USEN-01