Five vital steps for successful software delivery in a chaotic world

Applying DevOps throughout the development lifecycle

Contents

Executive summary

Delivering applications that meet the needs of the business can be a challenge in a complex business climate that is constantly changing. Diverse, multiple heterogeneous environments are the norm, all of which must be maintained and deployed. Multiple tools and technologies connect, support and create work. Processes are inconsistent, which contributes to difficulties with end-to-end lifecycle governance. Waste, rework and technical debt abound. At the same time, organizations are striving to be leaner by eliminating tasks that do not add value and by preserving existing IT investments. Woven throughout almost every one of these challenges is the need for speed and innovation while balancing quality and cost.

If you have found yourself in this complex, chaotic world, what should you do about it? What improvements should you consider? What practices should you have in place? This paper covers five steps that can help you address today's challenges and deliver software that yields better business results.

A changed landscape

The current IT landscape has moved from one characterized by monolithic applications with infrequent update cycles. Today's environments are characterized by frequently updated system-of-engagement applications. Delivery has shifted from updates that occurred once or

twice a year for system of record applications (for example, billing systems) to building, testing and deploying system-of-engagement applications as often as daily. These web and mobile applications that engage users have a dramatic effect on the business and are critically important to capturing market share. In addition, they have created the expectation for system-of-record applications to be rapidly enhanced with innovative capabilities. As a result, the business expects IT to act with speed and flexibility and release all types of applications faster. Balancing speed of delivery with improving quality, reducing cost and minimizing risk, and engaging the customer early and often for feedback become imperative. Several factors that define the IT environments of today make this a significant challenge.

Development, infrastructure, and operational environments are more diverse and complex. Your organization must develop for many platforms and then support them, including distributed, mainframe, mobile and cloud. The norm is multiple investments in technology and tools from many commercial software vendors and open source solutions to deliver applications that run on these platforms. On the positive side, this approach puts the right tool in the hand of the right role. On the negative side of this diversity are competing internal tools that overlap in capability and functionality. Typically, this diverse environment is also connected by brittle point-to-point integration that is costly to maintain and does not scale easily. Companies are reluctant to "rip and replace" their investments for sole sourcing to address integration challenges and barriers to effective communication, collaboration and connection. A platform that enables the various delivery team roles to collaborate and integrate their activities and key data elements has become essential.

Managing the software delivery lifecycle as a true end-to-end supply chain becomes more difficult as time goes on. Diverse tools and technology stacks are certainly contributing factors here. Combined with organizational silos, diverse tools tend to produce information in multiple applications and repositories. As a result, complete traceability and governance of the flow of information from business planning, release planning, requirements, design, development, test and deployment seem almost insurmountable. Everyone from producers to consumers of information encounter poor handoffs and long waits. Key information is often missing or is error prone because communication is hindered by the organizational silos. These issues are magnified as the delivery ecosystem spans global regions and third party subcontractors or partners. A platform that enables collaborative management of workflow and information and makes it easier to apply the necessary governance is needed.

A disconnect between development and operations teams is another contributing factor to today's challenging software development environment. The DevOps movement addresses this need and area. DevOps arose from the desire to have these two groups work together, even though their measures and focus differ. Development teams focus on speed and change, and operations teams focus on stability. These teams typically have a mix of processes that defines how applications are planned, coded, tested, built, deployed and delivered to production systems. Generally, these processes have different levels of maturity and are applied in techniques that range from manual and on-demand to advanced automation. If an organization is operating with more manual processes, it can experience inconsistencies that result in higher incidence of errors, wasted work and prolonged wait times. This impacts delivery speed. And lack of reliable, repeatable process standards can have an effect on operating efficiently to scale. Areas that are targets for process automation are code construction, test, build and deployment. The organization standards should be documented and institutionalized with configurations defined and managed by automation. An organization can achieve benefits by applying proven DevOps practices that can lead to improved enterprise scalability.

Managing the changes to code and configurations through development, staging, user acceptance and production environments is another software delivery issue. Many organizations

struggle with the contention for these environments and the lengthy ordering and provisioning processes to set them up. More "production like" environments that can be controlled and serviced by the development team can add value, eliminate waste and speed delivery. Creating such environments can eliminate risk because more errors can be found well ahead of production deployment when they are less costly to fix.

Despite these factors, the challenge of software delivery in today's changed IT landscape can be overcome. Five vital steps can help you adopt DevOps, improve your execution and deliver high quality software for better business results.

Step 1: Establish a collaborative platform

As mentioned previously, today's software delivery environments are characterized by silos of teams and stakeholders plagued by disconnected communication. This communication breakdown can dramatically affect the ability of software delivery to function as an end-to-end business process. Communication challenges can result in deliverables that must be reworked, which can add costs and delays. Communication difficulties between stakeholders can also cause other issues, such as:

- **Poor artifact handoffs.** Key deliverables are produced but are not readily available to others. These artifacts can be in different repositories or not moved through the lifecycle efficiently. They become difficult to locate, which creates delays that affect delivery speed.
- **Stale information.** Information is out of date, not updated in the context of work being done or maintained outside the software delivery process. An example of this is when project plans are not updated in real time based on software delivery activity.
- **Out of context discussions.** Important information is exchanged outside the context of work being done. This exchange can lead to inadequate feedback and negatively affect the timeliness of deliverable review, which then affects quality. An example is the use of email. Email requires scanning through inboxes and remembering to copy everyone on important correspondence.

A collaboration platform breaks down the silos and provides a way for groups of individuals to work together toward the goal of improved software delivery. Without a common platform to share ideas, plans, activities, tasks, work items and status, the software delivery process can encounter significant delays.

The IBM® Jazz™ Platform (Figure 1) unites, in real time, the contributions of stakeholders and team members, even when they are spread over different time zones and countries and include subcontractors. This cross-discipline collaborative platform records information in a single repository so all teams have a collective understanding of what needs to be done. In addition, a workflow is established to update the status of deliverables and complete reviews. Navigation is made simple so that all information for all disciplines can be easily accessed.

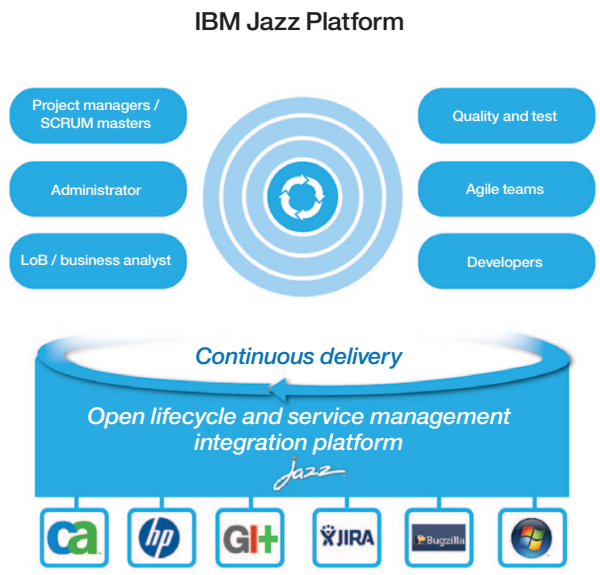

Figure 1. The Jazz platform helps facilitate collaboration throughout the development lifecycle

For example, suppose you are a tester who is creating a test script from a requirement. With Jazz, you can find the information without having to ask or schedule a meeting or rely on email communication that might not have included you. You can also answer questions that are posed by other team members quickly.

In summary, a common collaboration platform like IBM Jazz can eliminate communication challenges that lead to rework, poor quality and delays.

Step 2: Connect your investments

IT environments today are heterogeneous. They are a result of investments in different kinds of tools. Some of these tools were internally developed, others are from one or more commercial vendors, and still others are open source solutions. A common goal is to assemble these investments into an integrated lifecycle solution. However, the diversity of applications can hinder the DevOps and Application Lifecycle Management (ALM) efforts aimed at integration and traceability. Each tool might have different APIs, and some might be proprietary. Data is often stored inside the tools or in multiple repositories. Several approaches can solve this integration challenge.

One integration choice is point to point. In this scenario, each functional integration path is a custom bridge to the data needed or is moved by an understanding of each API involved. These architectures can be brittle and difficult to maintain or scale.

A better method is to keep the data where it is created and stored and link to it. Developers can reference information and use it to their advantage while staying in their tool of choice. The linked data approach is implemented as documented in the Open Services for Lifecycle Collaboration (OSLC) initiative. OSLC provides the community and the specifications for providers and consumers of lifecycle data.

Jazz is not only a collaboration platform; it is also an integration platform. Jazz embraces the OSLC linked data approach. The Jazz platform provides interfaces that organizations can use to assemble their preferred tool environment investments in flexible, seamless integrations. The Jazz architecture is designed to connect information and tasks from a diverse set of tools. Jazz is an open platform, and any tool can connect to the platform by following the OSLC specification.

IBM has developed adapters that help facilitate this OSLC approach very nicely. Integration is enabled via the Jazz platform to tools like HP QC, Jira and GIT under a product heading called IBM Rational® Lifecycle Integration Adapters.

IBM also partners with Task Top, which provides synchronization solutions that enable many diverse tools to integrate so ALM and DevOps become a reality. Task Top follows a more classical synchronization approach for integrating the data between various tool repositories. The benefit of this approach is that you can bring that data into the Jazz repositories where metrics and measures can be applied and viewed in dashboards. IBM's OSLC approach and Task Top's synchronization approach both enable the integration of data throughout the lifecycle.

After you have established your collaboration platform, you can build on it by connecting your preferred tool investments to create an end-to-end lifecycle delivery solution.

Step 3: Improve the speed of application delivery

The agile movement gained traction by providing a workable alternative to the challenges of traditional *waterfall* software development processes. These processes relied on rigid plans with hard transition dates. This approach worked well for years when software was less complex and more importantly based on relatively well-defined and understood requirements.

In today's world, developing complex applications can start with what might be termed high-level *use cases* or even just *ideas* rather than well-defined requirements. These requirements are not well understood, evolve as the software is being developed and in some cases are refined based on what the developers produce. This requires constant collaboration between the business stakeholders and development team, along with rapid iterations. In these iterations, small pieces of capabilities are developed, tested, deployed and validated with the customer to get feedback for refinement toward a complete application. This method, based on lean thinking, is the essence of agile development.

As agile development entered the mainstream and evolved, the need to extend these lean principles to the entire set of stakeholders in an application development project became evident. This led to the evolution of the DevOps movement, whereby development (Dev) and operations (Ops) began to work together to remove barriers and improve collaboration and communication. The principles of DevOps now include the entire application delivery team and encompass ALM processes. As a result, the entire process is more "lean," that is, more efficient with reduced waste.

Jazz works with IBM UrbanCode tools to create a DevOps platform that enables you to provide a single work item management and collaboration tool for all stakeholders. These stakeholders can include business owners, developers, testers, operations engineers and others. With Jazz and UrbanCode, information from across the entire DevOps lifecycle is available from one source to achieve collective understanding of what needs to be done. A common communications platform enables information to flow forward with the release assets as they progress through the delivery pipeline and also feedback to stakeholders in earlier stages (Figure 2).

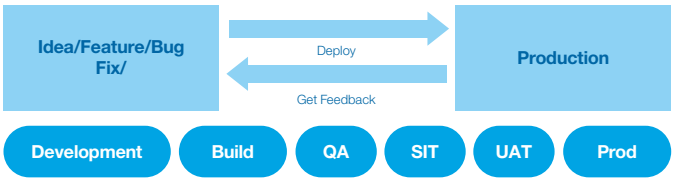

Figure 2. Information flow throughout the software delivery pipeline

Establishing a common collaboration platform integrated with deployment automation tools provides the ability for development, QA and operations to collaborate with one another throughout the delivery lifecycle. One of the key concepts in DevOps is "shift left." This concept promotes the shift of operational concerns to earlier in the development cycle. A collaborative platform that includes development, QA and operations enables these teams to establish common goals and communications right from the beginning of the development cycle. Operational concerns are visible sooner and are made available to developers and vice versa.

The combination of IBM UrbanCode Deploy, UrbanCode Release and Jazz enable development, QA and deployment work items to be traced to work items that represent actions in a release plan. Automated deployment tasks are enabled by UrbanCode Deploy, and tools such as Jazz Build Engine or Jenkins handle build tasks.

UrbanCode Deploy also integrates with IBM, HP and open-source QA tools, IBM service virtualization solutions and IBM and open-source artifact repositories to complete the DevOps tool chain. UrbanCode Deploy can target vendor and

open-source middleware with 65 plug-ins that automate deployment and configuration of components and content to these targets. UrbanCode Deploy can also target mobile app stores and cloud-hosted environments.

Step 4: Strive to gain customer feedback early

A critical component of agile development and DevOps is the feedback loop from all stakeholders who are further "right" in the delivery lifecycle and from customers or customer surrogates. To obtain the full benefit of DevOps principles, the feedback needs to be continuous throughout the delivery lifecycle. QA should be communicating with development. System testers and system integration testers should be communicating with QA and development. All relevant feedback should go to the business stakeholders. The goal of this rapid feedback is continuous improvement of:

- The application being developed
- The operational efficiency of the environments being delivered, where applicable
- The process of delivering the application itself

Often, customer feedback, which is a critical part of this loop, is overlooked or addressed very late in the process, such as after a release. However, making customer feedback part of the continuous information flow drives greater efficiency and effectiveness, which can lead to an overall increase in the business value of the software. When collaboration is not limited to one side but occurs throughout and beyond an organization, the result is a connection between business and delivery. This connection enables teams to respond quickly to changes, mitigate risk early and adjust priorities as appropriate. Management, development communities and customers are properly engaged, which increases the levels of control and agility in development cycles.

A record of customer needs combined with establishing and communicating priorities can ensure that the right requirements are being addressed at the right time and thereby reduce rework. Customer issues and concerns are therefore identified more quickly and easily, and more stakeholders are involved in the resolution process. Other than general feedback from testing and from observations of customer usage models, higher value reviews, comments, ideas and opinions can be gathered with A-B testing, application instrumentation and customer sentiment measurement. The IBM Tealeaf® solution and IBM mobile quality assurance tools have specific capabilities for obtaining and recording these types of customer feedback.

The end result is the successful delivery of software that is driven by business needs rather than reactions to errors in earlier applications and versions. Releases can be planned and are predictable because few surprises occur, and project teams can resolve an issue before many customers even notice one exists. Organizations can provide differentiated and engaging user experiences that are based on customer input, which builds customer loyalty and increases market share.

Step 5: Standardize your processes

One of the key values of adopting DevOps is that it represents a start in standardizing processes throughout the entire enterprise. Most large enterprises adopt standard processes for their key types of projects. These processes are typically at a high level and the project teams adapt them for their individual needs. In some cases, such adaptation is at a very low level, such as for deployment processes or work item or task management. Although they adhere to the broader process guidance, these lower level adaptations can differ enough to require teams to spend time learning and adopting them. Process standardization reduces the delays and learning curves associated with lower-level process differentiation, which can enable improved productivity and more effective software delivery projects.

Documenting these processes and automating them further improves productivity. When teams have standard processes they can follow and tools that automate error-prone tasks, they can be more efficient. In addition, process standardization and automation can result in less waste and fewer delays that are caused by errors or rework.

The IBM Jazz platform provides tools for standardization with process templates. IBM UrbanCode Deploy similarly captures and automates common, repetitive deployment processes.

Conclusion

Over the last 25 years, IBM developers and specialists have developed best practices that are based on common experiences. These experiences and best practices enable IBM to offer solutions that can help you take the five steps for successful software delivery. IBM solutions feature the capabilities you need for effective ALM and DevOps. The single open, extensible, integrated Jazz platform enables you to extend the software infrastructure investments you have already made. Real-time collaboration, visibility in the context of the work at hand, automation and process control enable ALM and DevOps from initial requirements definition to release management and beyond.

With Jazz and other IBM DevOps solutions, such as IBM UrbanCode Deploy and UrbanCode Release, you can unify your infrastructure into a single platform that does not require custom integration work or multiple maintenance fees. Fewer customizations and fees can reduce the overall costs of software delivery. DevOps teams, other stakeholders, management and customers can share knowledge and best practices for continuous improvement and delivery. Lifecycle artifacts are linked and processes are standardized to align teams, drive quality and prevent gaps and missed work. The result is rapid delivery of higher-quality software at a potentially lower cost.

For more information

To learn more about IBM DevOps and ALM solutions and the Jazz platform, please contact your IBM representative or IBM Business Partner, or visit the following websites:

- **ibm.com**/software/rational/jazz/
- **ibm.com**/devops
- **ibm.com**/software/products/en/category/SW860
- **ibm.com**/software/rational/alm/collaborate/
- **ibm.com**/software/products/en/ratlia

In addition, IBM's development team uses their own tools to openly develop the solution for Collaborative Lifecycle Management. Track their progress or get involved.

CLM project on jazz.net:
https://jazz.net/projects/clm/?ref_content=ribbon

Play with CLM in a sandbox:
https://jazz.net/development/sandbox/

CLM real-world sample and scenario:
https://jazz.net/wiki/bin/view/Main/MTM_Lifecycle_Scenario

Try DevOps services in the cloud: http://jazzhub.com

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 2014

Please Recycle

RAW14350-USEN-00