Best Practices for Agile Change and Release Management

by Ben Cody, Julian Fish and Amita Abraham
November 2012
# Table of Contents

A Deluge of Incidents at the Service Desk as a Result of Poor Change Management .......................................................... 1
The Dev-Ops Disconnect Dilemma................................................................. 1
Best Practices for Agile Change and Release Management ....................... 3
The Bottom Line ......................................................................................... 7
A Deluge of Incidents at the Service Desk as a Result of Poor Change Management

Fig. 1

Source: Forrester/itSMF Q2 2011 US ITSM Online Survey

The Dev-Ops Disconnect Dilemma

Increasingly, business is conducted on-line. To stay competitive, your IT organization has to continuously deliver innovative applications and services that are, very often, the face of your business. The ability to rapidly make changes to these applications without sacrificing infrastructure stability is no longer a “nice-to-have,” it’s a necessity.
Development organizations have responded to this need by adopting agile methodologies. They can now quickly push new or updated applications and services down the IT operations chute. However, this leaves IT operations teams scrambling to deploy these changes without introducing additional risks as a result of these changes. With less than perfect handoffs between these two teams in most organizations, it is not surprising that recent research* indicates that more than 40% of the incidents reported at service desks stem from failed changes to applications and their supporting infrastructure. Process disconnects between development and operations teams can seriously impact an organization’s ability to generate revenue.

Further exacerbating the process issues is the fact that most IT operations teams maintain their own systems for managing incidents, problems and changes related to IT infrastructure. These systems are frequently different from those used by the application development teams to track requirements, incidents, enhancements and change requests. The IT operations teams have typically had no access to or visibility into the fixes and changes made by the application development teams. Likewise, the development teams rarely have access to the tools that the IT operations teams use for tracking incidents, problems and changes. These function-specific siloed systems further compound the issue.

The challenges from process and tooling disconnects become apparent in this example of the release of a new online transaction portal at a telecommunications provider. The development team only informed the IT operations team a few days before the release that a different version of the Oracle database was required in the production environment. As the IT operations team had limited visibility into the details of the release, they were unaware of the deployment procedures and the need for a database upgrade. To further complicate the situation, the other applications that shared the Oracle database instance were incompatible with the newer version. As a result, the IT operations team was forced to scramble to procure additional hardware and stand up a new instance of the database. This resulted in an expensive and delayed application release that impacted revenue and drove a deeper wedge between the development and operations organizations.

The business impact of not being able to bring people, processes and systems together across development and operations teams is evident when applications that are the mainstay of a business falter due to failed changes and releases.

---

* Forrester/itSMF Q2 2011 US ITSM Online Survey
So how do you streamline processes that span your development and operations teams? How do you improve and speed change and release management without compromising environmental stability and control?

**Best Practices for Agile Change and Release Management**

**Create a Single Funnel for All Incidents**

When your customers experience issues with an application or have requests for new features, they typically send an email or capture their needs in a spreadsheet or word document. The problem with this approach is that these requests and issues can fall through the cracks. Your customers cannot easily track the status of their requests. A centralized portal that your customers can interact with to submit and track the status of their tickets can go a long way in improving satisfaction levels. For instance, in preparation for a release, an application development manager might need to request that the infrastructure team add a new web tier to an existing server cluster to handle the redesign of an application.
A central portal that displays the associated service level agreements (SLAs) and that collects the necessary cost center information for charge-backs and approvals can further streamline the process of rolling out application changes. In addition, a unified request portal that then automatically routes incidents to the right teams—be it within your application development or operations groups—helps them rapidly respond to and resolve issues.

**Integrate Change and Release Management Processes**

Integrating and automating change and release management processes eliminates the need to write complex deployment scripts, and removes the potential for human error when releasing changes into production.

The key is to provide both operations as well as development teams with complete visibility into planned changes and the scheduled rollout of these changes. Requests for changes and fixing defects get passed to the development teams. To help with rolling these changes into production, these teams can benefit from clear visibility into available pre-defined windows of change to production environments, called release trains. Release trains help with combining requests for both application and operational changes into a scheduled window and then rolling them out at a time that is suitable for both teams.

When both development and operations can clearly see the available release trains, easily combine features with planned operational changes and then track the progress of the release through development, test and production environments, the chances of failed changes are reduced significantly. By being able to trace the changes made to the initial request, IT organizations are better equipped to provide their business counterparts with accurate and detailed status updates.
With linked processes, development teams can track changes associated with a request at the source code level as it moves from development through to test environments and into production. Once an application is delivered into production, updates should automatically be made to the application’s Definitive Medial Library (DML) entry. When these application processes are linked to a configuration management database (CMDB) for infrastructure management, and updates are automatically made to items as soon as changes are released, it makes for a complete and consistent record of what is in production.

Getting your development and operations teams to work in concert through integrated processes better equips them to rapidly roll out application changes to support the business without jeopardizing the stability of the operational environment.

Fig. 3

Combine application and operational changes into a single release train
Provide a Unified Calendar for Complete Visibility
An integrated calendar accessible by both development and operations that displays all planned changes by week or month helps alert teams to scheduled updates to applications.

The ability to see the various applications that are impacted by a release train and drill down into the details of a request for change can be of great value to both development as well as operations teams. This should include the details of the application changes, right down to the artifacts to be deployed, as well as infrastructure change information. A unified change calendar provides development teams, release managers and operations teams with a consolidated view of all planned software as well as infrastructure changes.
Connecting processes across development and operations improves business satisfaction with IT as incidents and problems are tracked through to resolution, application changes are made sooner and business users are proactively notified when issues are resolved. By leveraging a unified calendar, the development and release teams are fully aware of the available windows for change, as well as the scheduled production down-times. This knowledge enables these teams to choose the appropriate time to request an infrastructure change, for example, to resolve recent application performance issues. Once an operational change is associated with a release train, control of the change should pass to that train. When the train is approved and flagged as ready for implementation, the operations teams should be automatically notified to make the necessary infrastructure changes while adhering to SLAs. The process should continue until all changes on the release train are implemented. Following post implementation review, updates should be made to the configuration management system that comprises the CMDB as well as the DML.

By linking approved development releases to existing operational maintenance windows, teams can avoid release delays and deployment confusion.

The Bottom Line

Organizations are increasingly conducting their business on-line and the need for speed in change management is crucial. Enterprises will benefit from having release management or DevOps teams serve as the glue that binds development and operations teams together. Tools and systems that link people and processes across development and operations can go a long way in providing these teams with the necessary visibility to gather and speed requests for changes to applications. An integrated change and release management strategy also reduces incident volumes. Research shows that 40% of all incidents submitted are as a result of failed changes. Connecting processes across development and operations improves business satisfaction with IT as incidents and problems are tracked through to resolution, application changes are made sooner and business users are proactively notified when issues are resolved.