White Paper

Best Practices in Release and Deployment Management

by Mark Levy, Sr. Product Marketing Manager,
Serena Software (now Micro Focus®)

MICRO FOCUS®

Table of Contents

Through 2016, a lack of effective release management will contribute up to 80% of production incidents in large organizations with complex IT services. —Gartner

The DevOps Disconnect Dilemma

In order to be competitive in today's environment, businesses have to adapt rapidly and cost efficiently in response to changes in the marketplace. This business agility has dramatically increased the volume of change requests into the IT organization. Adopting agile methodologies and continuous integration is the first logical step teams implement to satisfy this "need for speed," but that typically just pushes the bottleneck further along the path to production.

Software is not getting deployed into test environments or released into production environments any more quickly. Lack of control over the release process, poor collaboration between teams, and manual deployments are all leading to poor quality releases at a high cost to the business. With less than perfect handoffs between development and operations teams, it is not surprising that recent research indicates that through 2016, a lack of effective release management will contribute up to 80% of production incidents in large organizations with complex IT services*. 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 and development teams maintain their own set of tools for managing releases and deployments across preproduction and production environments. 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 operational systems and the knowledge base that IT operations teams use to manage and maintain the production environment. These functional silos 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.

* *George Spafford and Ronni J. Colville, "How IT Operations Can Set Up an Effective, Centralized Release Management Process." Gartner, June 3, 2013.*

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 deployments and releases.

So how do you improve the quality and increase the speed of your releases and deployments without compromising environmental stability and control? And how do you streamline processes that span your development and operations teams?

Improving and changing the way you release and deploy application software is a big challenge.

Release and Deploy Transformation

Improving and changing the way you release and deploy application software is a big challenge. Big organizational silos have been created, tools have been embedded, and many processes have been developed to support the current system. These systems tend to become brittle, unwieldy, and break when anything changes. Release management becomes a dreaded set of tasks that everyone has to be ready for, typically on the weekends.

How do you begin your journey to improve the quality and speed of your releases into production? As a first step, you need to take an agile and lean approach to understanding how you deliver value to the business, eliminating waste, making small incremental changes, focusing on continuous improvement, and delivering quick wins. It's amazing how small actions and changes can lead to awesome results. No big bangs, huge rollouts, and long projects—just a clear understanding of how you can deliver value to the business, your current environment, and measurements for success.

Gain Visibility and Insight into the Business

To begin, you need to understand what really matters to the business. What are its goals and objectives, and how does your job fit within the context of the business. Start by interviewing all of your stakeholders to get their feedback and input. You will be surprised by the different

You need to understand the culture and discuss goals, objectives, and prioritized improvements to help you see interrelationships and identify quick wins.

perspectives. Start with the business and finance people and work your way back to IT. Include application development and triage teams, operations and production support, and review incident and problem management reports. You need to understand the culture and discuss goals, objectives, and prioritized improvements to help you see interrelationships and identify quick wins.

Determine Success Up Front

What are your commitments to the business? Define success for your organization. Can you measure success and how you are doing against those measurements? What type of release cadence have you committed to and what can you actually support? Can you measure and report on plan versus actual? What about measuring deployment cycle times? You need to develop the metrics that support what matters the most to the business. It's also important to establish a common vocabulary of terms and concepts.

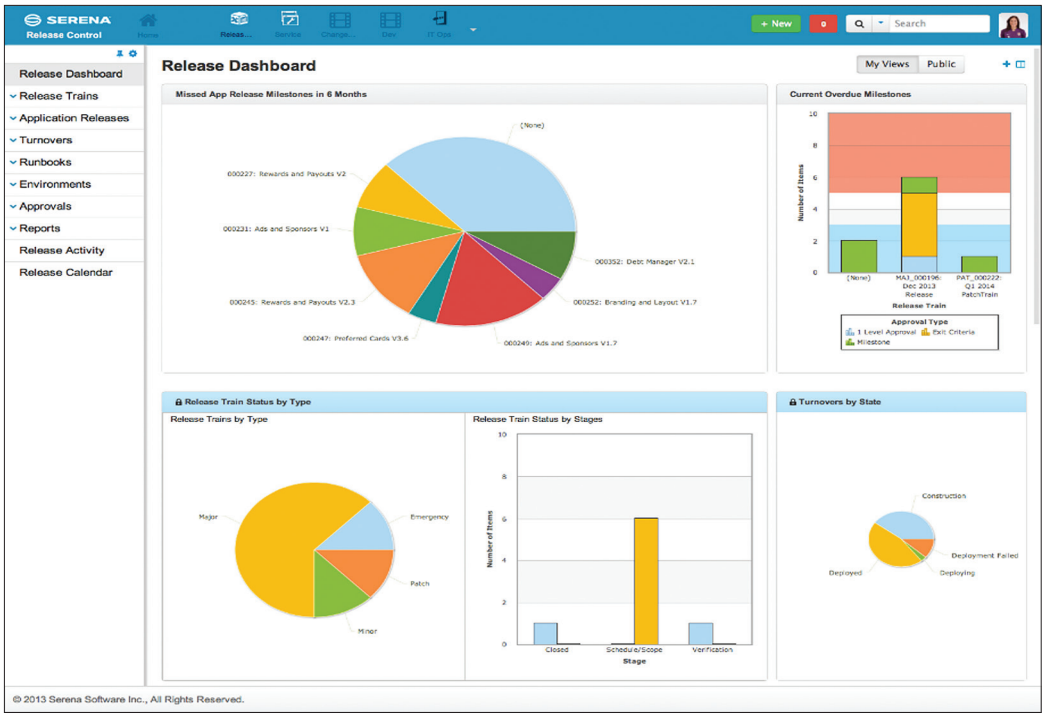

Fig. 1

Release management dashboard

See the System

You need to "See the System" and get a detailed picture of your current release process and how it aligns with your business objectives and release policies. Focus on the process flow from the business request all the way to production and identify what feedback loops are needed.

You need to understand the process information flow and where all of your deployable assets reside, along with the environments they target.

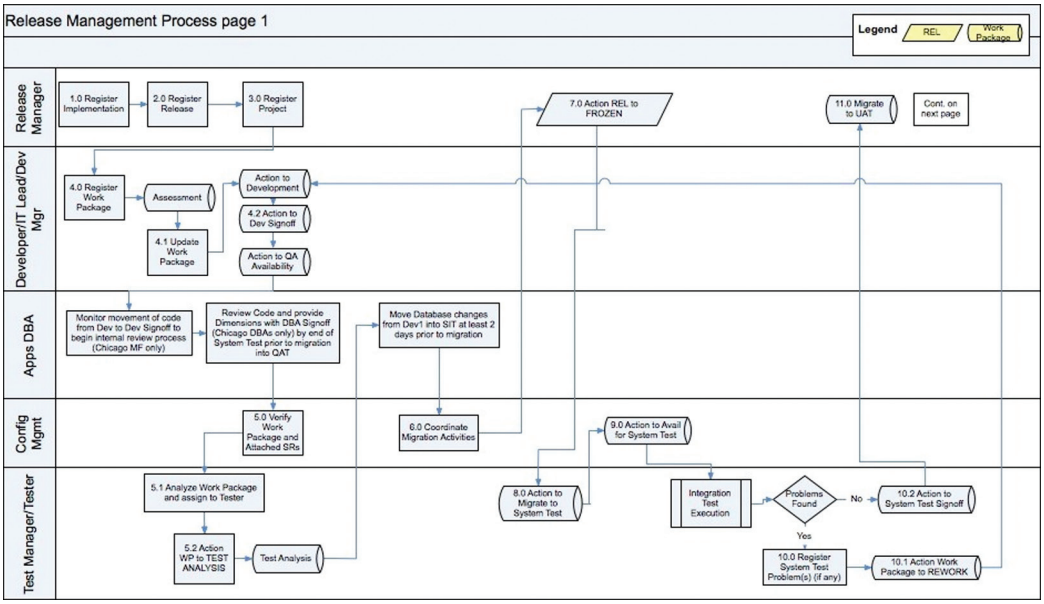

Fig. 2

Release management process flow

Where in the process are you spending the most time and what are the quality issues? Lean principles such as value stream mapping should be leveraged to further analyze the flow of information and artifacts. You need to understand the process information flow and where all of your deployable assets reside, along with the environments they target. Set an objective to reduce lead times across your release and deployment processes. Stay within your boundaries and look for ways to optimize throughput, simplify the process, and be more responsive. Eliminate waste and long wait states in the system.

Your objective is to “lean in,” understand the business, see the system, and “lean out” by identifying and eliminating waste across the release lifecycle. Think globally and act locally.

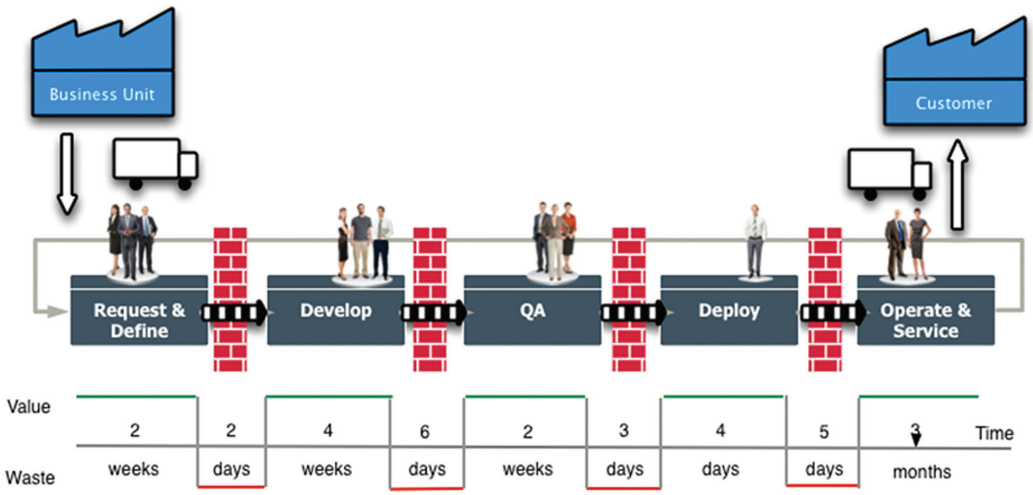

Fig. 3

Basic value stream map

Lean In and Then Lean Out

This assessment will give you good visibility into where you are spending your time and where you need to optimize and simplify. There is no single prescriptive order or starting place. Typically, organizations start by automating manual code deployments, but you might find that automating the routing of approvals could provide big immediate gains. Your objective is to “lean in,” understand the business, see the system, and “lean out” by identifying and eliminating waste across the release lifecycle. Think globally and act locally. There is typically a lot of slack in the system. Here are several best practices that organizations implement to improve both speed and control, while maintaining and improving the quality of their application deployments and releases.

Automation Is a Great Place to Start

Automation enables you to do repetitive tasks without tying up valuable human resources. People should not move or deploy the "bits" as machines are far better and much more consistent at deploying applications than humans. Automating manual tasks and handoffs is one of the first things you should look at. You can get a lot of quick wins with automation, and this bottom up approach can be very incremental and delivered rapidly without major organizational changes.

Automating manual tasks and handoffs is one of the first things you should look at. You can get a lot of quick wins with automation, and this bottom up approach can be very incremental and delivered rapidly without major organizational changes.

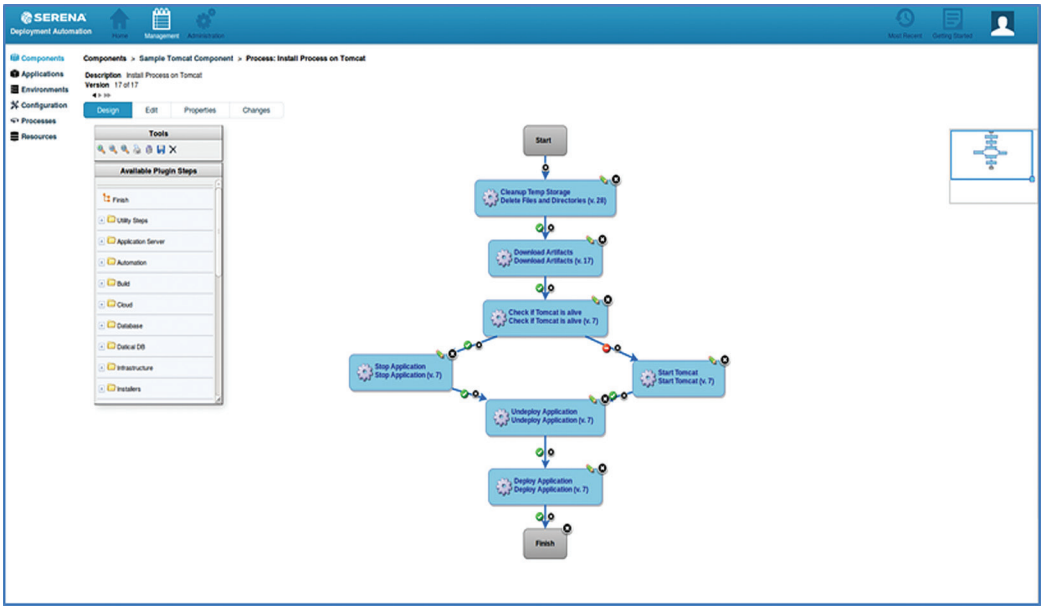

Fig. 4

Automating application deployments

Using version control as your single source of truth gives you the ability to pinpoint failures, easily roll back to a known state, and quickly deploy a new application.

Control and Secure Your Release Artifacts

Automating code deployments and using version control have been identified as two of the best practices high performing organizations implement. Using version control as your single source of truth gives you the ability to pinpoint failures, easily roll back to a known state, and quickly deploy a new application. Everything required to support and run the application, including infrastructure and configuration code, should be under version control. And, all of these release components must reside in a secure release repository that supports version control.

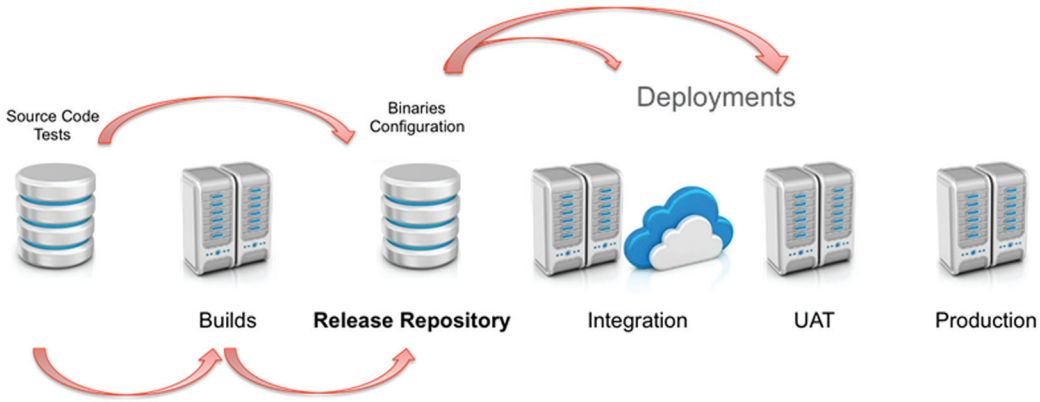

Fig. 5

Release artifact repository

The release repository ensures that deployed components are identical to those tested in preproduction environments. Without the repository, artifacts would have to be pulled from network shares or some other system, increasing both security risks and the potential for error.

Integrate Your Deployment/Delivery Toolchain

A toolchain philosophy espouses that a set of complementary task-specific tools are used in combination to automate an end-to-end process. There are many tools from different vendors that use different APIs within the release and deployment process. There will be overlap and you need to manage this. While a common set of tools across organizations is desirable, many times you will not have the ability to control tool selection, so it's important to have the flexibility to stitch these tools together.

Within the release process, you need to ensure that your release management tool can integrate with your change management system. It's very important to provide linkage to change and business requests. You should have complete traceability from the change request to the deployable artifact. The change and release processes need to be able to drive state transitions between the two processes. A release management tool that can orchestrate other tools and processes based on the state of the release process is a real advantage.

The change and release processes need to be able to drive state transitions between the two processes. A release management tool that can orchestrate other tools and processes based on the state of the release process is a real advantage.

Fig. 6

Integrates third-party tools to automate the deployment toolchain

The effective and efficient use of test and staging (preproduction) environments is critical to a successful release deployment.

A single deployment process provides a number of benefits previously discussed. The problem is that typically there could be a number of deployment tools as well as internally developed scripts that participate in the application deployment process. You need the capability to drive integration and deployments with a single process that can integrate with existing third-party products. This toolchain integration concept enables you to orchestrate and automate complex application and component automation processes across different tools and environments in a consistent, repeatable way.

Manage Preproduction Environments

The effective and efficient use of test and staging (preproduction) environments is critical to a successful release deployment. The combination of increased release velocity, number of applications, and the complexity of the application infrastructure stack have exponentially increased the complexity of managing these environments. Lack of test environment availability and/or environment contention can delay releases and increase the cost of release deployments.

Using office productivity tools (Excel or Word) does not work in large test environments. You need to be able to implement a process to schedule, manage, track, and control all of the test and preproduction environments in your release lifecycle. Automated and self-service environmental provisioning can further streamline the process to enable reduced cycle times.

A calendar-type view can give you a unified test management schedule, providing insight and visibility into who and what is scheduled for which set of environments. The schedule view should show the scheduled deployments, when maintenance is scheduled, and the release windows for each environment.

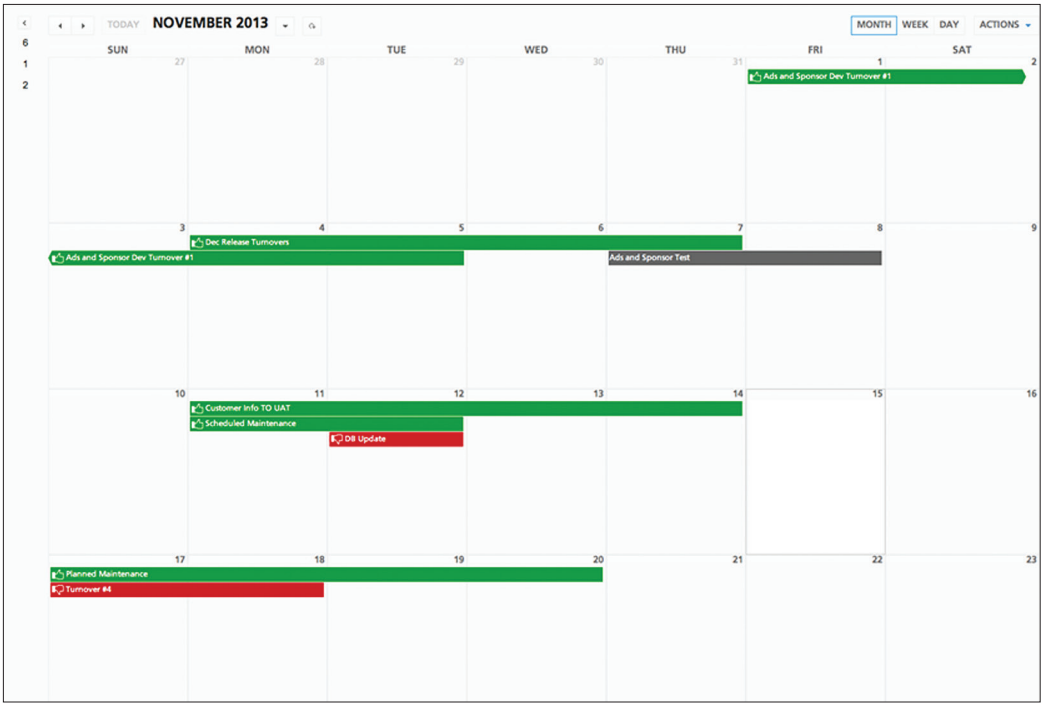

Fig. 7

Environment schedule calendar

Dashboards and reports based on environmental metrics enable management to easily and efficiently plan, schedule, and coordinate preproduction environments to support the release lifecycle. Stakeholders should get automatic notifications whenever there is a change in the state of an environment and when their changes have been implemented. Environment management is a critical part of the release and deployment lifecycle. It ensures that the right team has the right environment at the right time.

Environment management is a critical part of the release and deployment lifecycle. It ensures that the right team has the right environment at the right time.

Eliminating inefficient handoffs is critical for the success of your release.

Remove Inefficient Handoffs

The amount of required effort to move information and release artifacts from one process to another, or from one team to another, in support of a release is considerable and is a major bottleneck in the flow of the release. Any ambiguities require additional communication between teams to resolve and can result in significant delays, high costs, and failed releases.

Handoff problems occur when supported by conversations, e-mail, office documents, and ticketing systems. Unfortunately, only so much can be successfully conveyed with words and diagrams. There is a great deal of tacit knowledge, which is difficult to convey by just writing it down or verbalizing it. As the handoffs transition downstream, the actual knowledge bleeds off much like the telephone game you played as a child. The original intent loses meaning as each handoff transfers to the next team.

Eliminating inefficient handoffs is critical for the success of your release. The value stream mapping exercise mentioned above should provide you with good insight into the handoffs that can be eliminated. If you can't get rid of the handoff, make sure you have formalized the actual packaging and process of the handoff. Consistency and a set of semantics that everyone can understand will reduce the possibility of misinterpretation.

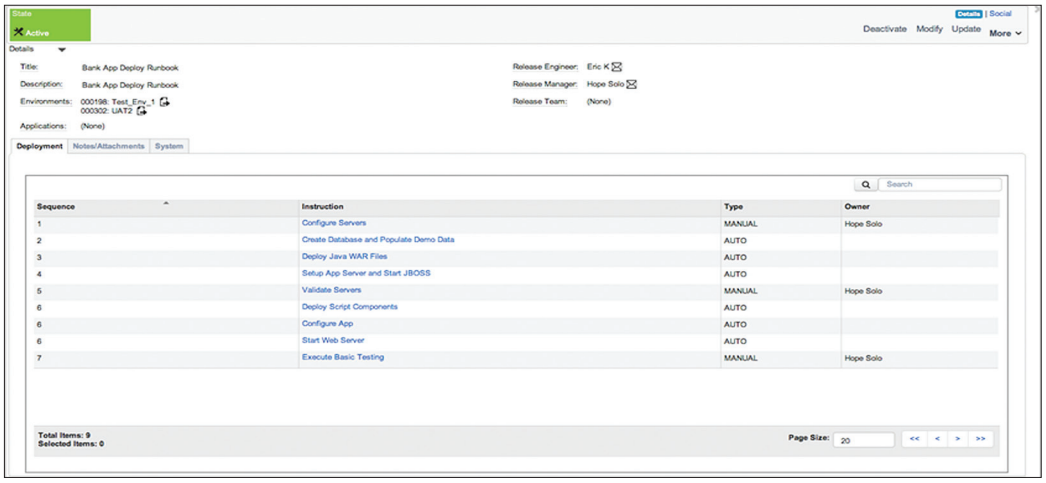

Fig. 8

Automated turnover deployments

Appropriately document knowledge where possible. Get rid of tribal knowledge, aggregate the multiple sources of knowledge, and use the concept of a "runbook" to document and standardize deployment processes and tasks. Runbooks provide a single source of reference for deployment procedures and tasks capturing best practices of deployment tasks for each associated application.

Start with best practice release management processes that can be customized and aligned to your release policies.

Orchestrate Your Release Process

A formal, automated release management process helps organizations maximize the value of their existing IT staff. Release management is an application lifecycle management process that guides IT efforts from application code development through testing and into production, helping to focus resources on timely delivery of the functionality that the business needs. A well-designed and comprehensive release management process enables organizations to:

- **Improve visibility and insight** into project timelines and progress, to help prevent surprises and keep all stakeholders on the same page
- **Increase release flow** to get critical changes into production sooner with existing or fewer resources
- **Improve release quality** to increase the number of successful deployments and reduce downtime, whether planned or unplanned
- **Simplify compliance** by enforcing the consistency and traceability of changes

Start with best practice release management processes that can be customized and aligned to your release policies. The ITIL V3 Service Transition book should definitely be reviewed. Streamlined and lightweight processes are a great start as you can customize specifically to support your business.

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.

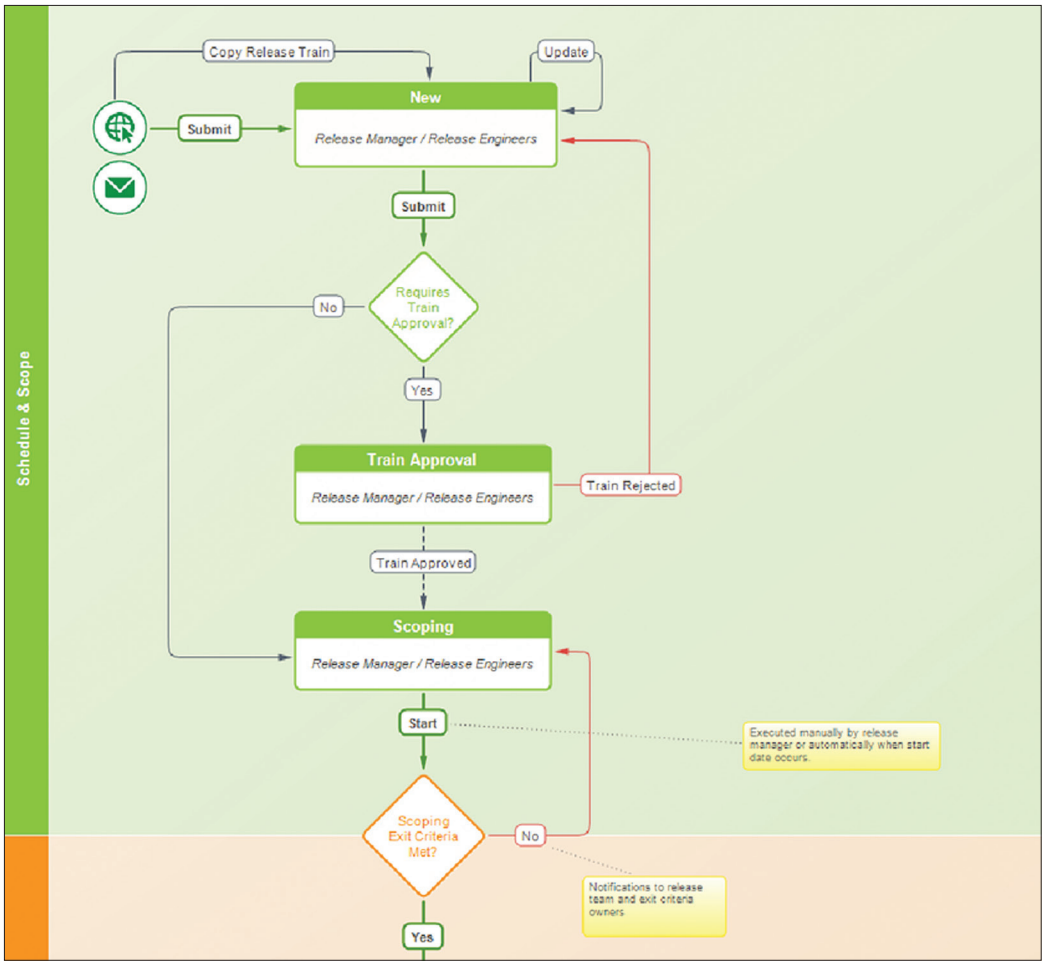

Fig. 9

Best practice release process

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.

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.

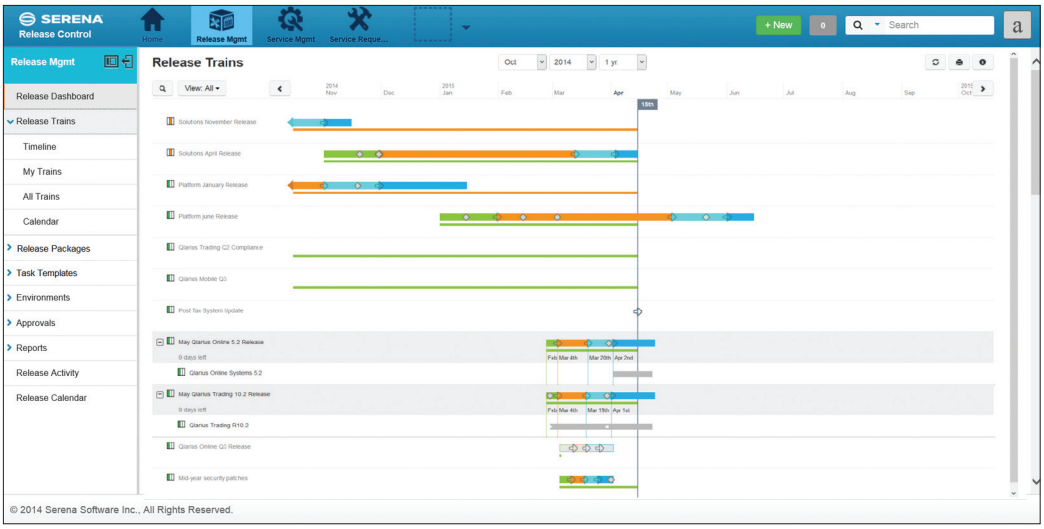

Fig. 10

Release Train timeline calendar view

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 and 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.

Release management teams require formal processes and tools for planning and tracking the steps of releasing applications, greater visibility into those processes, and a way to enforce release policies. Only with these tools and processes can the various teams synchronize their efforts to meet the expectations of the business, ensure the quality of released applications, and protect the integrity of mission-critical production systems.

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 and operations teams.

It is important to build on the success of automation by optimizing and integrating the toolchains, simplifying the handoffs and communications between development and operations teams, and standardizing on a single set of release and deployment processes.

Conclusion and Summary

The velocity and complexity of application releases continue to increase as businesses adapt to new economic conditions. Manual deployments, poor collaboration between teams, and lack of control of the release process all lead to poor quality releases at a high cost to the business. In order to achieve higher levels of performance, organizations should use an agile and lean approach to release management.

Automation is the quickest route to high performance, increasing quality and speed. However, although automation is necessary, it is not sufficient. It is important to build on the success of automation by optimizing and integrating the toolchains, simplifying the handoffs and communications between development and operations teams, and standardizing on a single set of release and deployment processes.

Micro Focus
UK Headquarters
United Kingdom
+44 (0) 1635 565200

U.S. Headquarters
Rockville, Maryland
301 838 5000
877 772 4450

Additional contact information and office locations:
www.microfocus.com

www.microfocus.com

162-000088-001 | S | 08/16 | © 2016 Micro Focus. All rights reserved. Micro Focus and the Micro Focus logo, among others, are trademarks or registered trademarks of Micro Focus or its subsidiaries or affiliated companies in the United Kingdom, United States and other countries. All other marks are the property of their respective owners.