

Mastering servicenow DevOps



Power Up Your
ServiceNow Development
with These 3 DevOps Building Blocks

Table of Contents:

- 3** Introduction
- 4** DevOps: The Main Ingredients
- 5** DevOps Challenges in ServiceNow
Why All the Headaches?
- 6** **3 Building Blocks for ServiceNow Success**
Continuous Integration
- 7** Continuous Release
Continuous Monitoring and Compliance
- 8** Getting It Done Right

Introduction

ServiceNow is incredibly flexible and powerful. Already a core technology, it's rapidly growing in importance for enterprise digital transformation of all kinds. But some businesses are experiencing growing pains and can't always keep up with demand when it comes to their ServiceNow development.

Born in the ITSM world, ServiceNow has evolved into a flexible, powerful platform to scale and automate business processes of all kinds, from HR and customer service to IT assets and facilities management. For all the benefits of ServiceNow, however, many organizations find the transition more challenging than they had anticipated.

While some workflows are available out of the box, many are not - and that means you'll be customizing and developing apps yourself in-house for the Now Platform®. And for many companies, this is where the trouble begins.

Unfortunately, ServiceNow delivery practices are behind the times when it comes to deployment and release.

Most of the development world has adopted DevOps and Agile practices and methodologies. And yet, in a world where code should take days to get to production, ServiceNow code still takes weeks and months to do the same.

This ebook will take a look at the building blocks needed to build an agile ServiceNow organization and triple its output. Then it will explore what it takes to make ServiceNow truly agile.

According to a survey of over 10,000 ServiceNow professionals on LinkedIn, it takes weeks and sometimes months for a ServiceNow feature to make it to production after it is finished in development.

Are you scrambling to deliver at the fast pace that today's applications demand? Nowadays, you need to get code into production in days. But ServiceNow code can still take weeks or months to get there. And you can't afford to wait.

DevOps: The Main Ingredients

ServiceNow has been a game-changer for a huge number of organizations. It is a powerful tool with versatility and impressive capabilities at its core, and it promises to help integrate and optimize business processes, eliminating human error and making sure nothing falls through the cracks. Those are probably some of the reasons that your business chose to implement ServiceNow in the first place.

But let's face it, it's not the easiest system to scale and administer development.

As ServiceNow teams continue to drive digital transformation using the platform, many find it challenging - to say the least - to harness the platform's full potential. Some are even seeing it become a bottleneck. The Now Platform® itself often fails to meet expectations when it comes to developing, customizing, and integrating applications.

As you know, most development today takes place in a modern development environment, meaning one that relies on some combination of DevOps, Agile, and CI/CD.

While each of these terms means something slightly different, they all evolved as alternatives to older "waterfall" development techniques. Under the old model, each phase depended heavily on the deliverables of the previous stages - meaning that a holdup in one stage created massive, expensive bottlenecks when it came to deployment and production.

Modern development methods all share a few common principles:

- **Incremental development**, rather than massive, full-scale application overhauls
- **Frequent releases**, once a day or more in some cases
- **Sustainable processes**, integrating automation and orchestration wherever possible

Modern development models are generally on a cycle of continuous development, integration, verification, deployment, monitoring, and feedback.

But when it comes to developing for ServiceNow, teams are often stuck with a process that's little better than the traditional waterfall model.

DevOps Challenges in ServiceNow

Why is ServiceNow development so prone to bottlenecks and delays? Why do teams find themselves stuck when it comes to customization and app development on the Now Platform®?

There's no single reason; instead, the answer to that question stems from the very nature of ServiceNow development.

For example, development teams share instances, and the organization needs to align these multiple streams into shared pre-production and production environments. Additionally, ServiceNow applications are usually complex, with multiple artifacts, new code, customizations, and data, which all need to be propagated from one instance to the next, and eventually into production. Finally, multiple scopes and update sets can contain colliding code and see team members overriding one another's work. Plus, the complexities of development are multiplied when quality and security are added into the mix - going beyond just achieving orderly deployment. All of this leaves ServiceNow development teams reliant on error-prone and laborious manual processes.

Why All the Headaches?

ServiceNow development bottlenecks tend to cluster around a few key issues with the platform:

- Core platform process complexities, especially at scale
- Multi-instance architecture, lack of "environment view"
- Feature and mechanism complexity; update sets, store applications, version control, plugins, data...
- No out-of-the-box support for CI, CD, nor Release Management/Release Automation
- Administrative complexity

In addition, there is no orderly release process built into ServiceNow, meaning organizations create their own ad hoc manual processes. Team leads and product managers are essentially using spreadsheets and even Microsoft Word to document the release process, including what needs to be deployed and in what order prior to release. With these types of failure-prone manual processes that vary from organization to organization, it's no wonder that delivery times for ServiceNow apps have been notoriously slow, until now.

The next section will explore three key DevOps methods that will power up ServiceNow to overcome these problems and get up to speed, meeting the challenges of modern development.

3 Building Blocks for ServiceNow Success

Implementing a continuous delivery model with ServiceNow requires that you both automate deployment and handle all the inherent complexities, including chaining of test automation, change management processes, and security and compliance tools. Creating a continuous deployment chain will turn your landscape into a truly agile continuous delivery factory and help your team keep up and get ahead. Let's explore the three most critical elements in building a truly CI/CD-enabled ServiceNow development pipeline.



1. Continuous Integration

In the waterfall development model, when multiple developers are working on a single project, code is merged infrequently, meaning that developers are often working with an out-of-date code base. When code finally is integrated (generally a manual process), many complex problems can arise, including missing libraries, broken dependencies, and more. Developers then work independently to fix these problems, creating new problems when the repaired code is integrated.

With a continuous integration (CI) model, on the other hand, developers can avoid major errors and problems further down the line. CI uses automation, in collaboration with a source code repository, to ensure that every single developer has the most current copy of the entire project code, which is generally updated several times throughout a typical working day.

Continuous integration automates development and QA flows (integration testing) to minimize manual steps, avoid costly merge conflicts later in the process, and ensure instances are in sync across the entire organization.

Incorporating CI comes with many additional benefits, such as setting up automated code scans for security and code quality checks, embedding automated testing (ATF or other) into the CI chain, and real-time notifications of problems—so you can resolve them early enough to avoid a bottleneck.

Obviously, while automation is the driving force behind efficient CI, it also includes the ability to pause for manual intervention (for approval, manual code review, and other processes that must be completed before code can be released).



2. Continuous Release

In the waterfall development model, releases are extremely infrequent and can only take place once all previous phases have been completed: design, coding, testing. While this model is easy to understand, it does not adapt well to complex projects with changing specifications. And if problems are found prior to release, it can lead to numerous delays.

Most ServiceNow developments currently use a waterfall model for production releases, which are even more complex than downstream deployments. Releases involve multiple applications and dependencies (for example, into global scope) and often include a combination of scoped applications, update sets from different scopes, data and elements that are not captured in update sets or scoped applications, and more.

Due to the complexity of production releases with ServiceNow, many organizations are using completely manual processes, relying on paper-based instructions that are difficult to follow and are not repeatable. Plus, releases are often done during off-hours and on weekends to avoid the risk of downtime during work hours. The continuous release model begins with automation. Moving away from manual-release “run books” and adopting automated release packages can immediately reduce errors; eliminate overtime and weekend work; and ensure stability, compliance, and consistency.

But the benefits of moving to continuous release don't end there. By adopting a true DevOps model, you have more control over release packaging. That means you can track release versions across your environment, ensure that releases are deployed in order, and even support automated multiple-scope releases.



3. Continuous Monitoring and Compliance

In the waterfall development model, it can take a lot of work to manage that final push to production. But then, developers can generally sit back, take a deep breath... and hope that nothing breaks that will send them back to the drawing board and have to scramble to fix code that misbehaves in production—or that contains performance issues, vulnerabilities, or security problems.

In most ServiceNow development environments, these issues don't come to light until right before release - or later, with problems becoming even more costly to resolve once the app is in production. This is one of the areas where a DevOps model can be most valuable, letting ServiceNow adapt to a modern, faster-paced approach.

One of the biggest advances that comes built into the DevOps process is continuous monitoring and compliance throughout all stages of development and production. Generally, this involves testing teams as well as security, incorporating automation wherever possible. This essentially bakes automated verification testing and security scanning into the ongoing process.

With the DevOps model, deployment to production isn't the end of the line; it's an ongoing process of improving the app from a wide range of perspectives: usability, security, functionality, user experience, and more. For that reason, your DevOps solution needs to consider more than just orderly deployment - you need to consider quality and security, too.

When problems arise, your organization needs tools to help you track and pinpoint them accurately. This makes for easier debugging and faster solutions to a wide range of issues. An effective DevOps platform will provide a single command-and-control center for all releases, letting you take control of your environment.

Having a comprehensive DevOps dashboard in place will also help you track ROI with relevant KPIs in a clear, measurable format.

Getting It Done Right

Why are so many businesses struggling with the ServiceNow development and release model? Well, many of them jump on board the platform mistakenly assuming that:

- ServiceNow offers what they need out of the box
- ServiceNow's native tools are sufficient for modern development
- ROI will automatically follow once ServiceNow is in place

Few, if any, of these assumptions actually turn out to be true - at least not right away.

But when you have projects you need to launch now, you can't afford to wait. By implementing the three DevOps building blocks explored above - continuous integration, release, and monitoring - you can achieve greater ROI from ServiceNow, much faster than ever before.

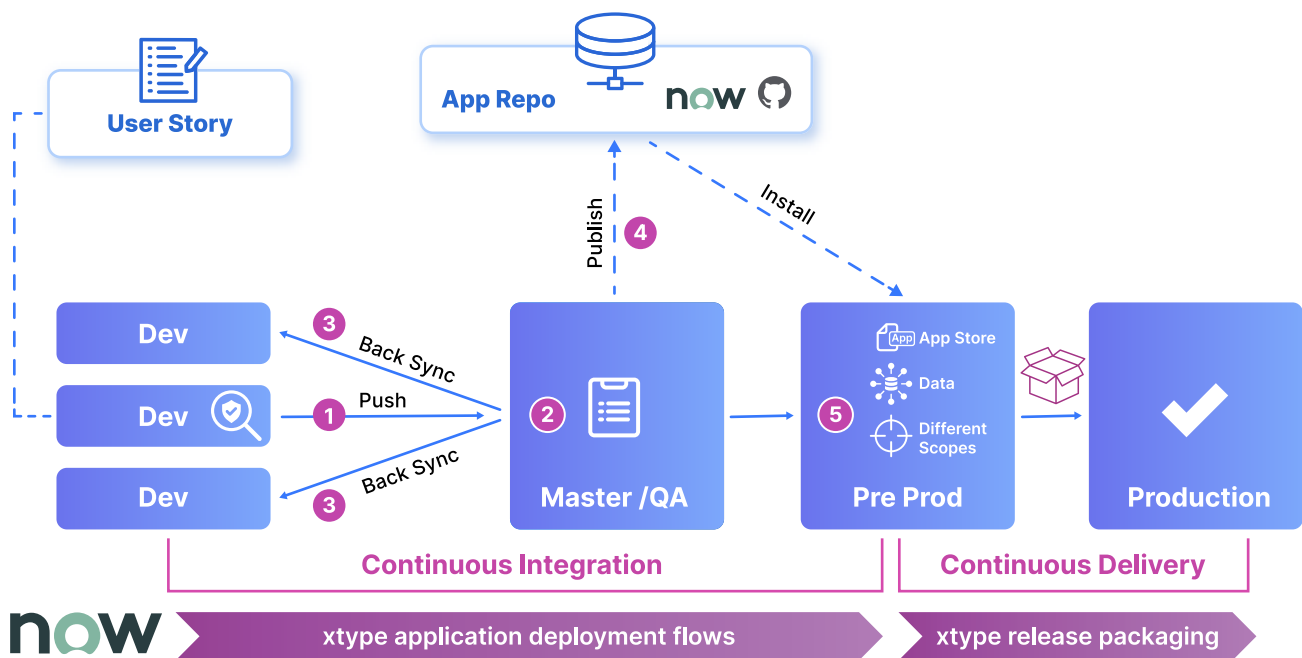
But when you have projects you need to launch now, you can't afford to wait. By implementing the three DevOps building blocks explored above - continuous integration, release, and monitoring - you can achieve greater ROI from ServiceNow, much faster than ever before.

But how can you implement these? Give your development team the edge they need with xtype, a ServiceNow-native DevOps application designed to help you build, test, deploy, and monitor ServiceNow applications on demand.

Xtype lets your team achieve high velocity by bringing true DevOps to the Now Platform with:

- An out-of-the-box DevOps process built into ServiceNow
- Continuous integration and deployment
- Continuous delivery using release packages

How xtype enables CI / CD for ServiceNow



These steps build the three critical DevOps building blocks into your process, helping you easily take your release cycle from weeks to days and letting your organization actually triple its ServiceNow ROI through a combination of speed, quality, and compliance.

With xtype, you can bring your ServiceNow app development into the modern age with a simplified, streamlined approach for continuous integration and deployments, automated delivery pipelines, versioned release packaging and automation, integrated testing and security, and continuous audit and compliance.

Plus, xtype works with all of your familiar tech stack - update sets, version control, store, GitHub, and more - letting you harness all the efficiencies of modern DevOps while keeping your pipeline free of bottlenecks.

You can install xtype in minutes and roll out your first automated deployment in 20 minutes flat. And, as our client success stories demonstrate, xtype consistently delivers impressive results:



Deploy five times faster to production



Dramatically reduce bugs and production issues



Remain more compliant and secure

Harness the power, scale, and flexibility you need to build the apps that drive your business - and do it easily. Xtype lets you get the most out of ServiceNow, at the speed and scale of modern DevOps.

Find out how easy it is to take your ServiceNow delivery to the next level.

Schedule a demo today