servicenow.

Now on Now:
How DevOps transformed our CI/CD pipeline to daily releases

Using ServiceNow DevOps is estimated to save us 2,800 engineering hours per year

Table of Contents

How DevOps transformed our CI/CD pipeline to daily releases

Regulations result in most companies having some kind of formal change process in place. If you are an application developer or test engineer, you understand how difficult it is to release code changes to a traditional production environment. Once coding is complete, you have to go through a manual process to create changes, wait for approvals, go through change review, and then finally deploy those changes. This can take weeks; then the use of heterogeneous tools makes visibility and reporting a challenge.

In this case study, we will discuss how we adopted ServiceNow® DevOps to transform our development process for our three cloud service teams, and the impact and benefits we're seeing following implementation.

DevOps goals

DevOps has emerged as the leading approach to delivering working software reliably, repeatedly, and consistently. It focuses on rapid product service delivery through the adoption of agile practices and provides teams fast feedback on code quality and developer performance.

Once we made the decision to implement DevOps, we laid out our primary goals:

- **Productivity**: Optimize code deployment and source code version control to drive more rapid deployment frequency
- **Quality**: Streamline code testing and quality to decrease failure rates and improve mean time to recovery
- **Costs**: Reduce QA, dev, and admin hours and number of operational instances to lower costs and focus developer time on value-add work
- **Operations**: Integrate tools to deliver improved end-to-end visibility and reporting

Our cloud environment

The Global Cloud Services team designs, builds and operates the ServiceNow cloud for our customers. We have built several Java applications to perform mission-critical work in our cloud, including those used for monitoring, advanced high availability, and auditing purposes. The applications are the heartbeat of our cloud and are critical to the success of ServiceNow and our customers. Three separate teams with over 70 developers work on these applications daily.

As the number of customer instances grew and our service began to scale, we began to see challenges with our agility and efficiency. Too many tools, difficult to use dashboards, siloed manual processes, and QA testing delays were just a few. Another big issue was the ability to orchestrate tests from one to multiple instances.

With the latest release of ServiceNow DevOps, we realized we could address our scaling needs while simplifying and standardizing our operations. It would integrate the different tools used for coding, testing, and deployment into one dashboard. It would transform our CI/CD process and give us visibility into our pipeline and our operations.

ServiceNow DevOps
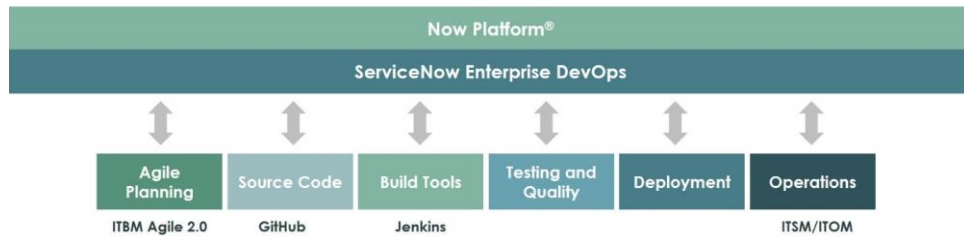
One of the coolest features of DevOps was that—out of the box—it integrates multiple tools that are not traditionally integrated or that take a lot of manual effort to build and maintain integrations. By linking the entire DevOps toolchain together, we would benefit from a higher level of analytics, collaboration, machine learning, and intelligent insights across all tools.

DevOps tool integrations
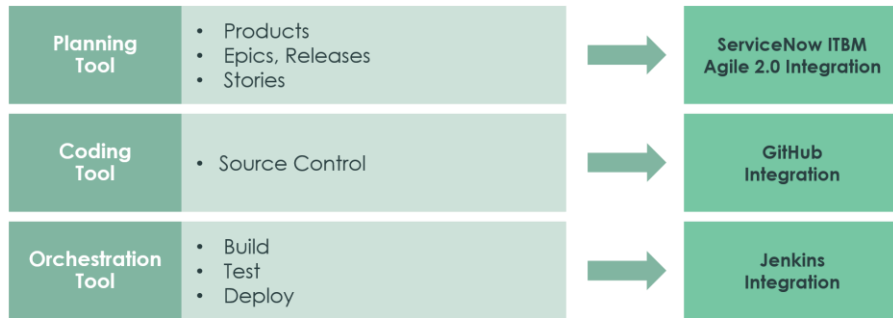
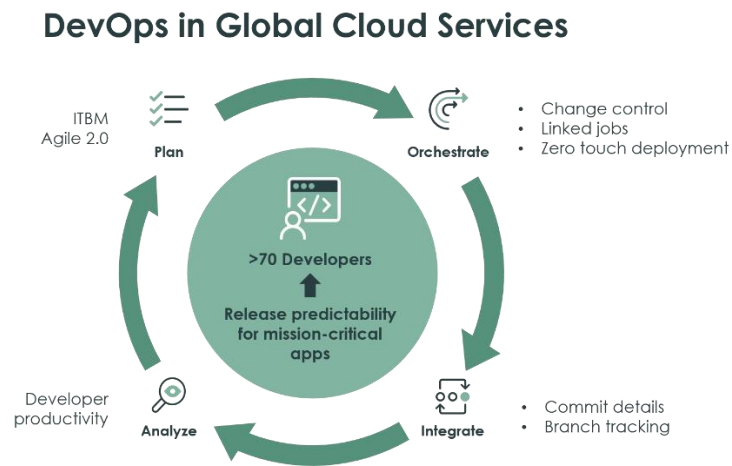In development, we rely on three different types of tools—planning, coding, and orchestration—in development.

- **Planning**: Our developers use ITBM Agile 2.0 to track their epics and stories for various products, giving us visibility into the pipeline.
- **Coding**: We store our code in a GitHub repository with DevOps integrations, which allows us to track commits. Once a commit is ready for deployment, we take it through a change control process. We can fetch commit details and also do branch tracking.

- **Orchestration**: For orchestration, build, test, and deploy we use Jenkins. The Jenkins jobs are linked so that we can verify the commits, enable change control, and track our deployment. We have orchestrated a zero-touch deployment using Jenkins and ServiceNow.

We have standardized the above tools across all three teams but the fact that the ServiceNow DevOps solution normalizes data from a wide range of similar tools means that we will be able to easily include other teams using other toolchains in the future if needed.

How our DevOps process works

This diagram shows how DevOps works in Global Cloud Services. The first step is for our developers to create a story on ServiceNow ITBM Agile 2.0 for every commit.

DevOps in Global Cloud Services

ITBM Agile 2.0 — **Plan**

Orchestrate
- Change control
- Linked jobs
- Zero touch deployment

>70 Developers

Release predictability for mission-critical apps

Developer productivity — **Analyze**

Integrate
- Commit details
- Branch tracking

We have a Jenkins job that polls for new commits from GitHub. The job build status is tracked in DevOps so we can see the relationship between job builds to commits and then to stories. We get an aggregate of all commits for a day. The Jenkins job then creates a change using DevOps change acceleration. For pre-production changes, we plan to automate the change completely so that we retain the change governance, but the code can be pushed directly to deployment.

Once the development managers have signed off on the change, the change is approved, and DevOps starts the Jenkins job for deployment testing. We also track the status of deployment in DevOps so we have end-to-end visibility—from commit to deployment—within the DevOps project itself.

DevOps product release pipeline

Continuous integration → Continuous deployment to lab → Controlled to deployment to production
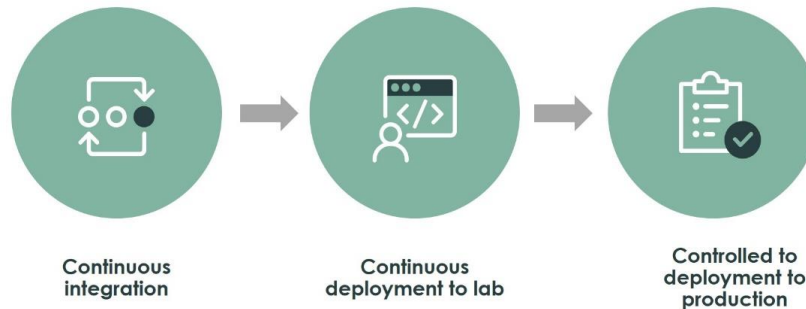
Our pipeline is a simple, three-phase process. Code is continuously integrated, then deployed to the lab for further testing. The last step is deployment to production. Using ServiceNow DevOps, we can track:

- Commits quality, deployment status, and build info
- Builds deployed in lab, production, including status info
- Metrics on commit quality: build failures, unit test failures
- Metrics on deployments

We also enabled the DevOps Insights feature to track the quality of our commits so we could better understand the time from a commit to deployment. With this analysis, we can identify and remove bottlenecks and further improve the process. Because the data is gathered from the pipeline automatically, it is always complete and consistent.

Impact of DevOps on Global Cloud Services

With DevOps, we have seen many improvements in our operations, including:

- **Better transparency:** A dashboard centralizes information on a number of key metrics like commits deployed, commits pending deployment, and status of the commits that are merged and deployed. It also gives us visibility into the development and quality of all commits.
- **Improved efficiency:** We have centralized tracking of development activities including branch and commit details, which gives managers complete visibility into productivity and efficiency.
- **Increased quality:** We have improved the quality tracking of commit status and see fewer errors. We can also track the quality of builds deployed against the build verification tooling (BVT) results.

Scaling is important to our success. Using ServiceNow DevOps is estimated to save us 2,800 engineering hours per year, or the time of 1.5 engineers. The savings come from reducing the time developers devote to change management and automating the deployment process. This increases our overall agility and our ability to scale up to multiple clouds, such as public clouds.

Adopting the ServiceNow DevOps solution has given us centralized visibility across heterogeneous tools so we can manage the entire CI/CD process in one place. It enables us to align with our existing business and change management while still reducing the overhead for the developers. We can gather insights to improve pipeline velocity and developer productivity.

By combining ServiceNow workflows and Jenkins, we now have zero-touch deployment and our development operations are on the same platform as our cloud instances. With all ServiceNow and third-party tools connected to one platform, we avoid the need to integrate and manage multiple tools.

About ServiceNow

ServiceNow is making the world of work, work better for people. Our cloud-based platform and solutions deliver digital workflows that create great experiences and unlock productivity for employees and the enterprise. For more information about ServiceNow DevOps, click here.

Now on Now is about how we use our own ServiceNow solutions to work faster, smarter, and better. With Now on Now, we're achieving true end-to-end digital transformation. To learn more, go to the Now on Now website.