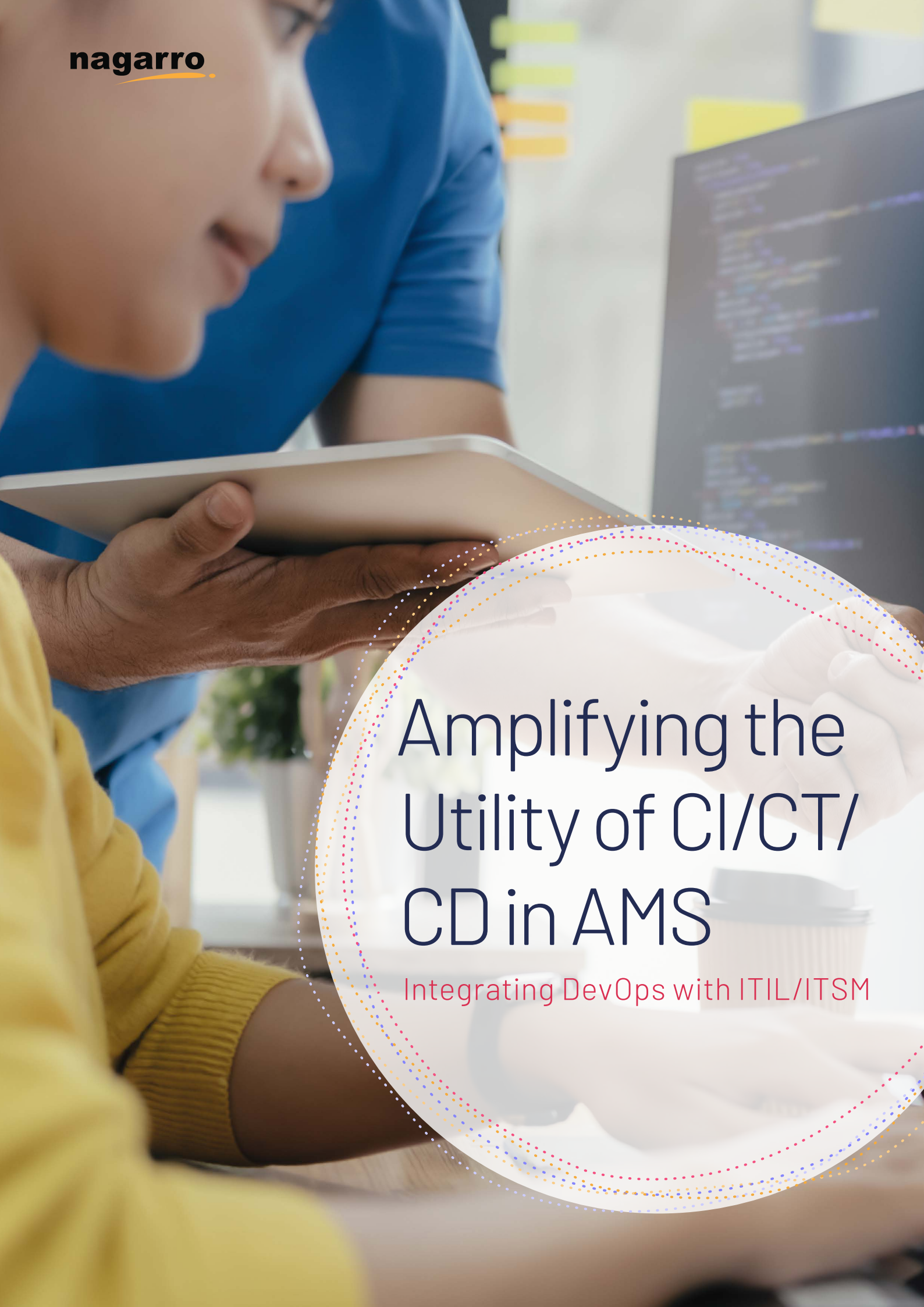


nagarro

Amplifying the Utility of CI/CT/CD in AMS

Integrating DevOps with ITIL/ITSM

nagarro

Table of Contents

ABSTRACT

Conventional software development and delivery methods like the Waterfall method are becoming obsolete rapidly. This has paved the way for a new era of DevOps, i.e., Continuous Integration (CI), Continuous Testing (CT), and Continuous Delivery and Deployment (CD) with as little human interaction as possible, thus, replacing the quarterly, bi-annual, or annual releases with multiple deployments in a day.

Application Managed Services (AMS) focuses on providing services, such as incident management, problem management, and change management. This white paper focuses on the AMS projects that are evaluating and implementing CI/CT/CD with the Information Technology Service Management (ITSM) and Information Technology Infrastructure Library (ITIL) processes, thus using DevOps to make the ITSM processes leaner.

The white paper helps you understand precisely what CI/CT/CD is, how it is beneficial for AMS, and how its usage can be amplified in AMS. We will also discuss various processes and technologies that help increase the use of CI/CT/CD in AMS.

1. THE STORY SO FAR

Frameworks and standards are an excellent way for organizations to showcase their capabilities. A few examples of these frameworks are ITIL, Six Sigma, and CMMI.

ITSM refers to the implementation of quality IT services that meet business needs through effective collaboration of the right people, processes, and information technology. Every running IT system can be called as ITSM. It includes planning and managing changes without causing any disruption in business.

ITIL defines a set of best practices for ITSM. It focuses on:

- Providing IT services that fulfill customer needs.
- Improving the quality of IT services by managing the availability and capacity of resources.
- Maintaining the continuity of processes.
- Reducing the cost of IT deliveries by minimizing the wasted effort.
- Getting the right things done at the right time.

ITIL has been successful as a result of its vendor-neutral and non-prescriptive nature. It enables organizations to deliver value and integrate service strategy, business strategy, and customer needs. Furthermore, it also enables the adoption of a standard approach to service management and optimizes the cost.

The Best of Both Worlds

Nowadays, organizations have shifted their focus to delivering applications and services at a higher velocity (by evolving and improving products at a faster pace) than traditional software development and infrastructure management processes. The concept of CI/CT/CD concentrates on saving money and resources, and delivering scalable systems. It also aligns IT performance, and responsiveness to business through speedy software releases.

Having understood the ITSM/ITIL and CI/CT/CD processes, it's time to combine them to enhance their capabilities.

A few benefits that can be achieved as a result of this are:

- Faster time to market.
- Rapid delivery of features and fixes for incidents.
- Higher confidence while moving emergency fixes to production.
- Better service availability.
- Early evaluation of potential threats.
- Reduced risks (risks associated with updates to complex systems).
- Early detection of root cause and respective workarounds for existing issues.
- Automation of repetitive procedures during a deployment.
- Robust workflow to ensure that the service lifecycle undergoes proper implementation.
- Regular feedback from customers that helps in improving the quality of the next releases.

As we can see, the whole is greater than the sum of its parts.

2. THE WAY FORWARD IS TOGETHER

Companies today need to integrate ITSM and DevOps to manage customer requirements effectively. The stability provided by ITSM enhances the process of CI/CT/CD.

What is AMS?

AMS can be defined as the following:

- A set of specialized organizational capabilities for providing value for customers in the form of services.
- A professional practice supported by knowledge, experience, and skills that focuses on a service-oriented approach.

The act of transforming capabilities and resources into valuable services forms the core of service management. Let us see how CI/CT/CD can be achieved in AMS.

What is CI/CT/CD?

Continuous Integration refers to combining the work of individual developers in a repository like GitHub and Visual Studio Team Services. This is done several times a day to encourage the early detection of bugs, thus reducing the integration costs.

Continuous Testing refers to early and continuous testing across the application life cycle. It leads to reduced costs, shortened testing cycles, and continuous feedback on quality. This can be achieved by automated testing and service virtualization (simulation of production environment).

Continuous Delivery is an extension of Continuous Integration that focuses on releasing new changes quickly, i.e., automating the release process and deploying the application with the click of a button. With Continuous Deployment, every change that passes all the stages of the production pipeline is released to the customer. It increases customer involvement to a great extent, thus accelerating the feedback loop that leads to continuous improvement in the application.

Figure 1 shows how ITIL processes (incident management, change management, and demand management) and DevOps can work together:

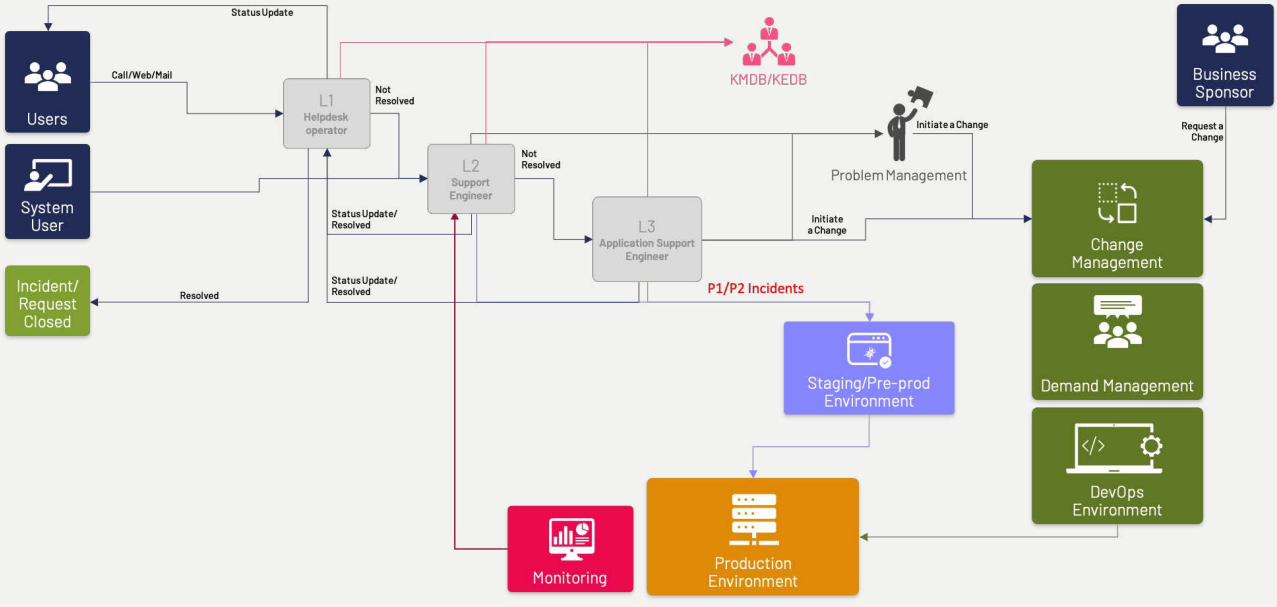


Figure 1: ITIL-based processes for greater resilience

CI and testing naturally lead to the practice of CD, i.e., the process of automating the deployment of software to testing, system testing, staging, and production environments.

3. CI/CT/CD in AMS

The interaction of the Knowledge Management Toolset with the Incident Management Toolset and Project Management Toolset can be understood from Figure 2. The Incident and Project Management Toolsets and development machines store data in a Data Repository Server, from where the data goes for Auto Build, Auto Deploy, and Auto Testing.

Figure 2 shows how AMS can be integrated with the DevOps environment:

- CI allows a large team of developers working on cross-technology components at multiple locations to deliver software in an agile manner. To achieve Continuous Integration, you must keep in mind the following points:
  - A repository equipped with version control management, such as Git, VSTS, or SVN must be used.
  - All the commits must be pushed to a single development branch.
  - All the unit/system tests should be executed on the committed code.
  - On-premise tools (such as GitLab CI, TeamCity, Bamboo, GoCD, Jenkins, and Circle CI) or Cloudbased tools (such as BitBucket Pipelines, Heroku CI, Travis, Codeship, Buddy CI, AWS CodeBuild) must be used.
- CT can be achieved by test environment provisioning and configuration, test data management and integration, functional performance, and security testing. All the unit/system tests are executed again at the application-level production environment. Thereafter, the application is deployed on an environment that mimics the production environment.

  Before starting with CT, the following prerequisites must be met:
  - A planning tool like JIRA for tracking changes.
  - A version control system like GitHub.
  - A build tool like Jenkins or Bamboo.
  - A testing tool like qTest Pulse that provides test validation against JIRA issues and connects to the source code repositories. A few other open-source tools for continuous testing such as Selenium Automation Framework, Appium, and Cucumber.
- CI and testing naturally lead to the practice of CD, i.e., the process of automating the deployment of software to testing, system testing, staging, and production environments. A few other tools available for CD are ElectricFlow, Octopus Deploy, and DeployBot.

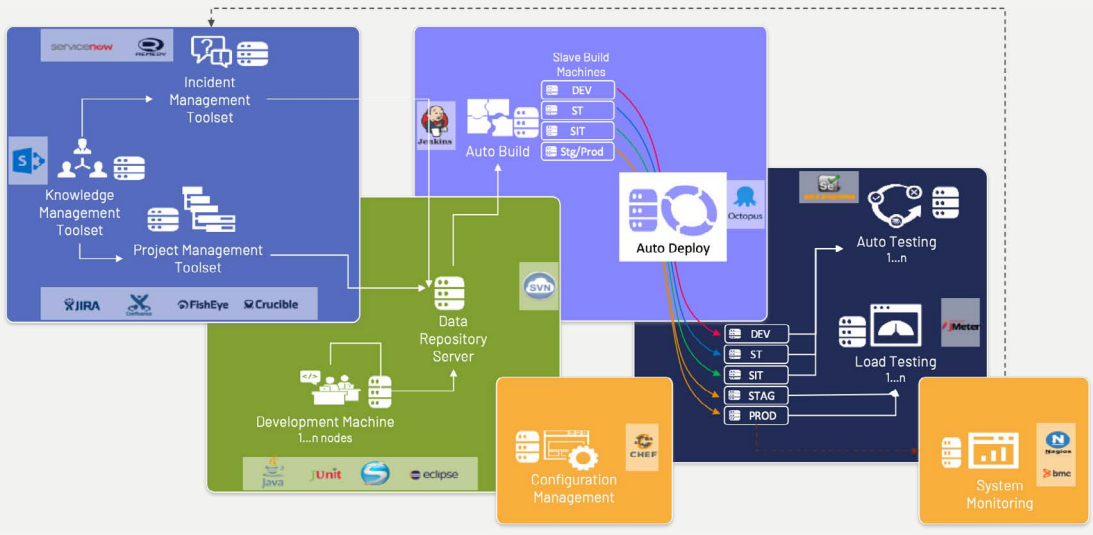


Figure 2: Integration to utilize DevOps environment

4. BENEFITS OF CI/CT/CD IN AMS

1. Periodic Discovery

It helps in detecting potential threats at an early stage that leads to finding the root cause and identifying workarounds at the right time. This considerably reduces the risks associated with a complex system. The automated discovery of the configuration changes makes it possible to keep the Configuration Management Database (CMDB) and the Content Management System (CMS) updated.

2. Auto Upgrade

The process of DevOps aims at continuous communication between the customer and the stakeholders. The mechanism of continuous delivery helps the client to stay updated with the latest code changes. This ensures that changes in the releases remain in agreement with the client's expectations, thus, ensuring the code updates are in sync, and adhere to the standard regulations.

3. Incident Management

Through auto-discoveries, such as Probe, and AWS, physical and virtual assets are automatically imported into the CMDB. It allows tracking of all hardware and software inventory required for operations that is beneficial for both the development and the operations team. There is a requirement of regular communication between the development and operations teams to make proper use of this benefit. The communication can be improved at a hierarchical level by keeping everyone in the loop. This helps in managing the incidents effectively.

4. Problem Management

DevOps helps in automating repetitive tasks that lead to faster and quality deliveries. Automated discovery of configuration items makes it easier to update the CMDB. This automation is achieved by creating an automated script that enhances the deployment and builds a robust workflow to ensure that the service lifecycle undergoes proper implementation.

5. Service Level Agreement

Services take into consideration the service transition from design to development to operations. DevOps makes it easier to gauge the actions and give immediate responses if the service transition from design to operations has been satisfactory. It uses feedback as a medium for the development team to improve the quality of the next release and reduce the cost of disruption.

6. Application Management

CI/CT/CD increases the delivery velocity of business applications and optimizes their performance. This plays a key role in application management by helping in Application Performance Management (APM) and on-premise system monitoring.

7. Automation

The process of CI/CT/CD focuses on automating your software development lifecycle without much human intervention. Automation helps the client to cut costs, speed up the service delivery and business process automation across physical assets, IT, and cloud. The processes described in the ITIL v3 framework rely on workflow automation to ensure that stakeholders understand and complete the assigned tasks.

8. Continuous Delivery

Continuous Delivery is achieved by removing barriers between the development and operations teams and by creating automated scripts for the entire application build package and deployment tasks. It shortens the lead time for new releases of the applications, improves feedback loops for quality releases, and ensures that information is automatically updated in the CMS and CMDB.

5. HOW TO AMPLIFY THE UTILITY OF CI/CT/CD IN AMS

Let us see how CI/CT/CD helps in crucial AMS processes, such as change management, release management, configuration management, and so on.

1. Change Management

Every type of change request is evaluated in terms of risk and value. These change requests can be divided into the following categories:

- Standard change (low-risk change)
- Normal change
- Emergency change (needs immediate attention)

CD is responsible for reducing risk and decreasing the cost of regular releases in the early cycles of the delivery process. DevOps tracks changes from the application to the deployment level by operating the business efficiently, conducting change planning, and responding to incidents as and when they occur.

2. Release Management

The focus of release management is to build a detailed plan and test the new services before any release or deployment. You must ensure that it is conducted in a controlled premise to deal with all the possible environments and outcomes. Also, automated build and Continuous Integration practices create releases in verifiable packages.

3. Configuration Management

Configuration Management helps in understanding the interfaces between configuration items to track changes and verify that the right version of the code is implemented (that has not undergone any unauthorized changes or technical errors).

In addition to the benefits listed above, a few other ways in which CI/CT/CD can be helpful in AMS are:

- Analyzing the current processes and tools to create a prioritized roadmap for optimizing the release pipeline.
- Setting up Continuous Integration, testing, and deployment processes for the apps, including automated tests in automated release cycles.
- Automating the provisioning of new environments. This will help in saving the setup and maintenance costs and making better use of infrastructure and slash new environment lead times.
- Setting up a continuous program of improvement instead of adopting different isolated best practices. This helps in removing bottlenecks in the delivery cycle.
- Using DevOps to invent things in an agile way and using ITSM to save money with the ITIL processes.

6. SUCCESS STORY

Nagarro worked with a customer — a leader in online education — to implement an automated integration and deployment environment that helped in achieving shorter resolution time by reducing the manual efforts. The environment was in complete sync with the customer's strategic vision, i.e., achieving agility in all the IT processes.

Figure 3 shows how Continuous Integration, testing, and deployment was efficiently used in AMS.

The customer was responsible for sending change requests and the Nagarro support engineers created tasks in JIRA. The following toolset was used:

- JIRA as the planning tool.
- SVN for version control that acted as a single code repository for the code commits including the changes in the application components. While committing changes, the JIRA ID of the task for which the code was being committed was used as a link.
- Jenkins for CI. Scripts to integrate SVN and Jenkins were created so that on every commit in SVN, a job gets triggered in Jenkins.
- JUnit for unit testing. The commands on Maven were run to execute the test cases.
- Selenium for system testing (both dev and staging environment).
- JIRA and Jenkins integration using the JIRA plugin.
  For more information, see https://wiki.jenkins.io/display/JENKINS/JIRA+Plugin.
  In Jenkins, the following jobs were performed:
  - Build
  - Deploy
  - System testing
- In the build phase, all the JUnit test cases were executed. In case of success or failure, the task for the respective JIRA ID was updated with the appropriate result. In case of failure, a mail was sent to the configured email IDs.
- Similarly, deployment and system testing results were updated in JIRA and notified through emails. If the system testing was successful, then the above procedure was repeated for the staging environment too.
- For continuous monitoring, Nagios was used as it follows the client-server architecture. In case of any misbehavior, such as the server not working or high server load, an email was sent to the engineers and to the mail server, thus bringing OTRS into the picture. OTRS and Nagios were integrated in such a way that emails were picked automatically from the mail server by OTRS.

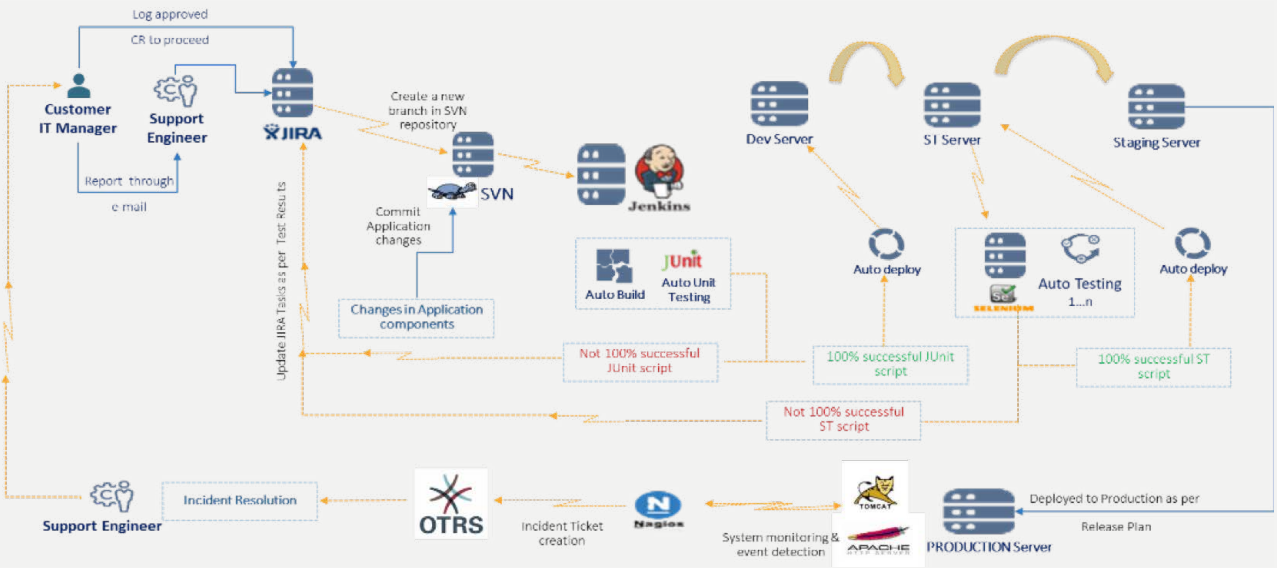


Figure 3: An example of an Automated Continuous Integration & Deployment

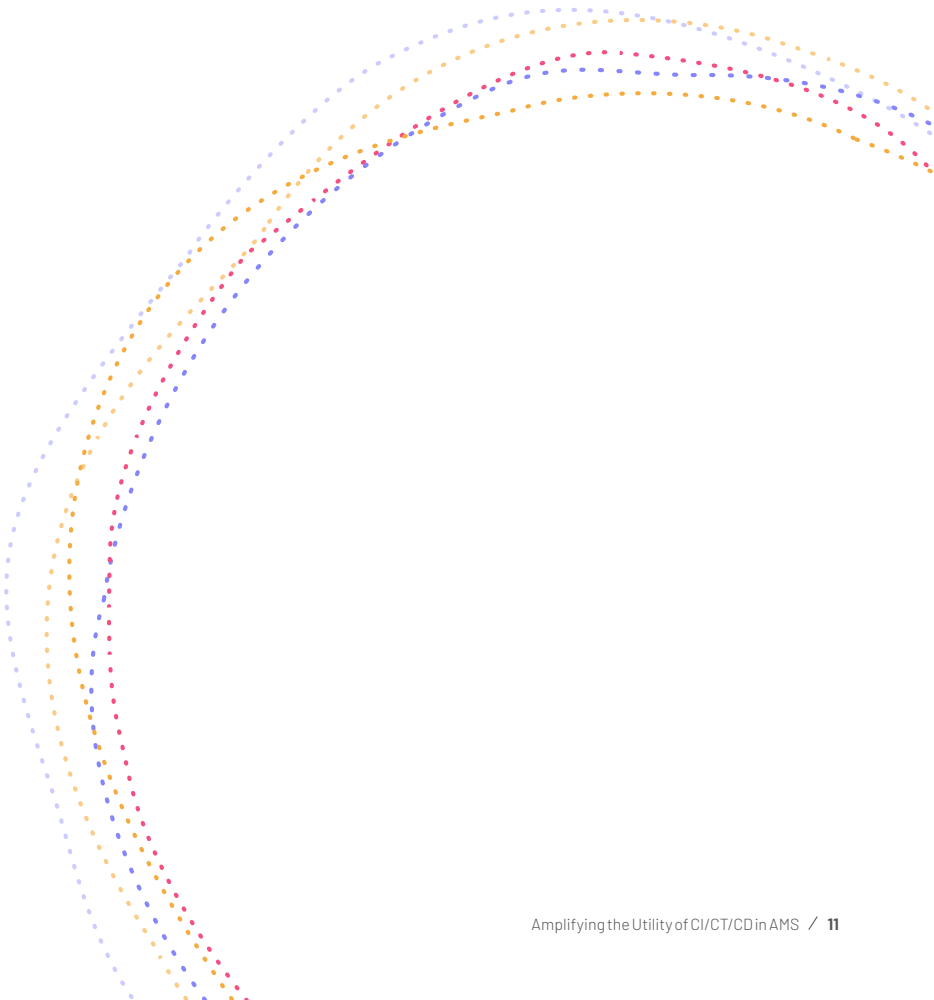
CONCLUSION

With all the above discussion in mind, it can be concluded that:

- ITSM and DevOps (CI/CT/CD) are not required to be mutually exclusive.
- Using ITSM and DevOps together can be highly beneficial.
- ITIL is quite important around change management and applying CI/CT/CD in this process helps in maintaining control over change management.
- Both ITSM and DevOps focus on delivering value to the customer.
- ITSM provides visibility to support the rapid pace of operations created with DevOps.

REFERENCES

1. DevOps for Dummies by Sanjeev Sharma, IBM Limited Edition
2. https://freshservice.com/solutions/itil-devops
3. http://www.enterpriseappstoday.com/management-software/6-tips-for-using-itsm-in-a-devops-world.html
4. http://www.cigniti.com/blog/top-10-open-source-tools-for-testing-devops/
5. https://dzone.com/articles/21-automated-deployment-tools-you-should-know
6. https://dzone.com/articles/how-does-devops-handle-change-management
7. https://stackify.com/top-continuous-integration-tools/
8. https://wiki.jenkins.io/display/JENKINS/JIRA+Plugin



AUTHOR

Nikunj Goel is a Delivery/Program Manager with over 15 years of IT experience. He has delivered projects in various methodologies like Agile, DevOps, CMMI and ITIL. He has been working with Nagarro since 2012 as an offsite Delivery Manager. Nikunj holds a Bachelor's Degree in Computer Science from a premier institute of India.

MARKETING TEAM

Shailesh Dhaundiyal
Editor

Lisa Pradhan
Content Marketing

Mehak Vohra
Designer

nagarro
THINKING BREAKTHROUGHS

ABOUT NAGARRO

Nagarro drives technology-led business breakthroughs for industry leaders and challengers.When our clients want to move fast and make things, they turn to us. Working with some of our SAP clients, we continuously push the boundaries of what is possible related to innovations through process optimization and technology progress. Today, we are more than 6000 experts across 21 countries. Together, we form Nagarro, the global services division of Munich-based Allgeier SE.

CONNECT WITH US

info@nagarro.com www.nagarro.com /nagarroinc /nagarro /nagarro /nagarroinc /lifeatnagarro