

WHITE PAPER

Enabling your enterprise to make a smooth transition to DevOps

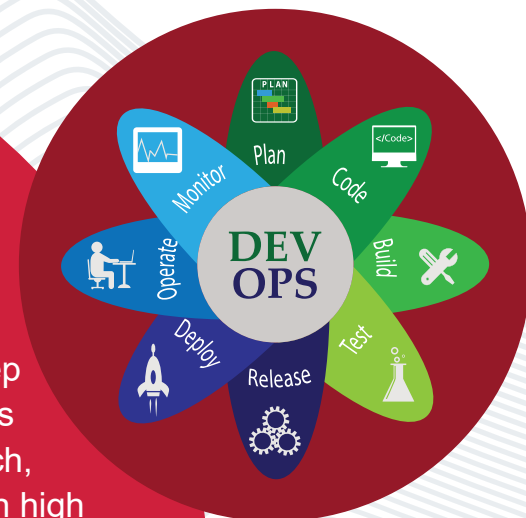
The pandemic has substantially changed the way businesses operate. A majority of the global workforce now functions remotely as a result of the COVID-19 crisis. Enterprises have had to navigate this new normal, and company leaders have been forced to adapt to the unexpectedness of the current scenario. While some organizations already had a digital framework in place to make the transition, others struggled to find their footing.

This situation has only highlighted the need for reliable infrastructure capabilities and self-service IT ecosystems within an organization. This is one of the critical areas where DevOps can make a difference. DevOps practices are designed to provide improved business value to customers, suppliers, partners with lean management principles, and an agile approach to development.

What is DevOps?

Before the introduction of DevOps in 2009, software delivery typically followed a waterfall model that consisted of a set of predefined stages that were executed in sequential order. The output of one step was necessary for the next stage to function, and all stages were mutually dependent on one another. With this approach, fixing bugs and adding new details to the program resulted in high costs, more errors, and management bottlenecks.

DevOps refers to a broad approach to IT delivery that aims to break silos and build communication, collaboration, and integration between processes, team members, and technology. It bridges the gap between development and operations teams and helps multiple teams avoid a siloed approach. Team members that are part of both operations and development will work together right from the design process through the development stage and support. This enables organizations to operate with shorter feedback loops and faster time-to-market.



Steps to take before a DevOps transition

Before organizations embark on a DevOps journey, there are a few essential pointers to keep in mind. Some of them are:

- **Establish the need for DevOps:**
While DevOps practices promise to offer innumerable benefits, it is often helpful to determine your reasons for making the switch from a legacy system. While DevOps provides advantages such as increasing the speed of delivery and reducing the cost of operations, the organization's end business goal and solving specific pain points should be the dominant factor when opting to shift to DevOps.
- **Ensure that company stakeholders agree:** The fundamental goal of adopting DevOps is to develop an integrated perspective among individuals with different viewpoints on solving critical issues. Therefore, before making the shift to DevOps, it is essential to make sure that the business leaders and other relevant stakeholders are on the same page about technological evolution, and the problems DevOps will solve at an organizational level.
- **Define key metrics to gauge progress:** Company leaders must find a way to get current metrics of the various stages of the software development cycle before making the transition. These metrics can be captured again after the shift to DevOps to determine and analyze whether DevOps practices are making a difference.
- **Monitor inter-team collaboration:** A successful DevOps shift would stem from breaking down organizational silos and increasing communication and collaboration. Some operational processes may overlap between teams. It is essential to keep those in mind and encourage cooperation amongst teams to ensure a successful transition.
- **Plan the budget:** Creating a budget with a rough estimate of the organizational expenses during the transition will ensure minimal wastage of capital.
- **Start with the basics:** It is unwise to transition the entire organization to DevOps within a short period. Instead, start by incorporating a culture of collaboration in individual teams, note their shifts in behavior, and what worked and what did not. This will allow company leaders to replicate the same transition model on subsequent teams and make the shift on a larger scale.
- **Don't automate all processes at once:** Though automation is one of the core principles of DevOps, it will be fundamentally unwise to automate everything together. Some methods work better in a manual setup, and it is prudent to analyze and understand if the effort behind automation is worth it for smaller, infrequently used processes.

- **Select the right tools:** The tools that are selected while transitioning to DevOps should be compatible with each other and the current working environment. This will ensure smooth operation and enhance the working environment.
- **Ensure continuous integration and delivery:** Continuous integration (CI) and continuous delivery (CD) are two of the main principles in a DevOps journey. Agile processes that ensure CI and CD – where the product is developed in smaller sprints with regular evaluation of possible flaws – should be incorporated.
- **Make security a key checkpoint:** Most organizations fail to incorporate the right resources to monitor and strengthen the security of the infrastructure. It is recommended that strong security measures be put in place to ensure robust safety in configuration setups and infrastructure.
- **Enhancing customer satisfaction:** Ultimately, customer satisfaction is one of the key metrics with which a successful DevOps transition can be measured. Transparency among different sectors will work to keep consumers happy and increase business gains.



Embarking on a successful DevOps journey


The key factors that ensure an organization's successful implementation and working of DevOps are:

Continuous integrated operation:


This involves collecting code changes and collectively testing them in systematic and automated phases. This helps in detecting flaws, correcting them early, and ensuring quality before releasing the product.

Consistent inter-team communication:

Ensure that there are no internal hiccups while breaking down individual segregated models and connecting them to a unified model.



Policy management: As manual management becomes a thing of the past, tracking flaws and reconfiguring setups have become automated, and it saves time and increases efficiency.



Configuration management: The implementation of DevOps leads to the elimination of manual management of host configuration. Both operational work and configuration will systemically get managed through Infrastructure-as-Code (IaC).

Key DevOps tools

Code repositories	GitHub, Bitbucket
Release management	Jenkins, Travis, CircleCi, TeamCity, Gradle, Bamboo
Configuration management	Puppet, Chef, Ansible, CFEngine
Orchestration	Mesos, Apache ZooKeeper, Kubernetes
SaaS environment	Azure DevOps, Jenkins, AWS CodePipeline, TeamCity
Virtualization and containerization	OpenStack, Vagrant, AWS ECS, Docker, Kubernetes

Most organizations worldwide are on a quest to become more responsive to the changing market landscape, and DevOps is making that possible. Adopting DevOps methodologies is a process of continuous improvement that will gradually increase operational control, decrease deployment time, and integrate users, developers, and system administrators into a unified workflow based on providing cutting-edge customer experiences. It is a surefire way for organizations to gain a competitive advantage and become more agile and achieve significant business benefits.