

# Adoption Barriers for DevOps

[www.4waytechnologies.com](http://www.4waytechnologies.com)

BRINGING DEVELOPMENT AND OPERATIONS TOGETHER



*By Mansi Gaur*

# CONTENT



- A. Introduction
- B. What is DevOps?
- C. 4 Pillars of DevOps
- D. Principles of DevOps
- E. Barriers

## Introduction



Every business large or small wants to use DevOps strategies to enhance their business, evolve and stand above all their competitors. The terminology DevOps is getting a lot of popularity worldwide.

There are many misinterpretations about DevOps practice. People usually fail to understand what it actually is, and even if they understand it correctly but still, the implementation remains the biggest challenge for them, that too on their currently working IT processes and organizational management.

There still exists the miscommunication between the development and operation teams. The developers work hard to develop a non-ambiguous, efficient, top-notch quality application and then wait for the operation team to make the testing environment, which the development team doesn't find good enough or takes lots of time to get ready. And on the other hand, the operational team thought the developed application isn't work-efficient or good enough. Due to this ambiguous situation and misunderstanding between the two teams, even the companies who profess to have adopted DevOps as a full-fledged development and operational strategy still fail to understand the difference between the two interrelated processes. Both these fields and their concepts and processing in **silos** were divided and integrated with the concept of DevOps to guarantee constant, stable, reliable and high-quality application/software deliveries. And to understand everything better let's have a look at DevOps first.

## What is DevOps?

DevOps is a set of practices that combine **software development (Dev)** and **information technology Operations (Ops)** together and make it easy to make efficient applications/software. DevOps aims to shorten the systems development life-cycle and provide continuous delivery of applications with high software quality.

DevOps uses sets of tools (not a single tool) for better working that is called **Toolchains**.

Toolchains categories -

- **Coding** - Development of code, reviewing it. Source code management, code merging.
- **Building** - Building of status, continuous integration tools.
- **Testing** - Working with testing tools, testing the application/software before releasing it.
- **Packaging** - Pre-deployment stage. Packaging of the application/software and making it ready for the release.
- **Releasing** - Releasing the application, automation or the updation of the previously released application/software.
- **Monitoring** - Keeping the check on the application performance, end-user experience.
- **Configuration** - Configuring the application/software infrastructure and management.

## 4 Pillars of DevOps

Adopting DevOps means being completely aware of the 4 pillars on which it rests.



### 1. Agility

Agility means the ability to move or grow quickly and easily, the ability to understand and think quickly. Agile methodology/ agile software development in the IT industry has various approaches to software development. Developing an application is about finding the requirements and the solutions to the problems. And this only achieved through the collaboration of **self-organizing** and **cross-functional** teams. Agile methodology is responsible for -

- Adaptive Planning
- Evolution Development
- Early Delivery
- Continual Improvement
- Rapid, reliable and fixable Updation



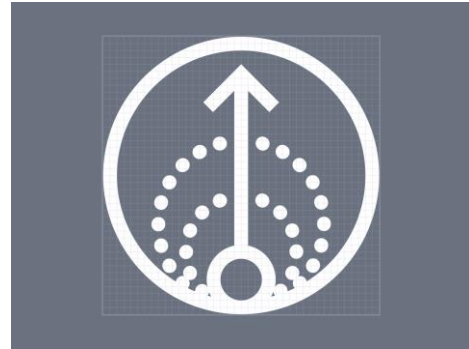
## 2. Consistency

In simple terms, consistency means stability. This one seems to be the easiest one and least important but like the rest of the three, this is also a major part for DevOps. Consistency for the software/application development is a must if the development process isn't consistent then how the end-application can be.



## 3. Quality

The standard of the application/software developed in comparison with other similar applications in the market should be always high. And to maintain the quality is one of the major goals of DevOps. Delivering the application with the degree of excellence is one of the basic foundations of DevOps.



## 4. Scalability

In layman language, scalability is the capacity to grow/expand according to the changing world. But in computing terms, scalability is the ability of a process to produce effective solutions or modifications in a range of capabilities.

It is a property of a system, to handle a growing amount of work by adding necessary resources. And in terms of software/application scalability is about handling the increasing number of users and be ready for modifications.

## Principles of DevOps

- **Automation:** In DevOps, many processes are automated, especially the iterative processes as they recur often or after a fixed interval of time. For making a process automative researching, analysis and testing is a must.
- **Continuous Integration, Delivery & Deployment:** To enhance your code quality, routine testing of code is a must. Continuous Integration is an amazing way that gives faster, efficient and error-free codes. Continuous Delivery and Deployment assure time-efficiency when it comes to reviewing the code time and testing, increase code coverage and efficiency in moving codes from staging to production or another environment.
- **Version Control:** This is one of the most essential parts of coding. Version control is a part of the system that records changes/upgradation to an application/software over time so that one can see specific versions later. It is also helpful to track errors, exceptional failures, etc.
- **Configuration Management:** CM is a system's engineering process for establishing and maintaining consistency of a product's performance, functional, and physical attributes with its requirements, design, and operational information throughout its life. Configuring your application to stand on the user-expectation is a must. Automation,

version control and configuration management together can work wonders for your application.

- **Cloudification:** It is all about data storage but with a slight twist. It's not like you using servers to store the information or data of your application/software on a server that you physically purchased or rented. It's more like a virtual server, the cloud where your data is being stored and you paying to the companies who are providing you with the cloud services. And you don't have to worry about the physical damage to the server because the data is being stored on the cloud, the virtual server.
- **Search, Learn, Experiment:** Search, learn and experiment is the key for any organization to succeed. Search about everything related to your idea, learn about it and from your failures and mistakes, recognize the best possible tools, practices or processes, experiment with it bring innovation and improvement in the system in all. This very strategy shields DevOps from unsought risks and helps it rebound to its original state much faster.
- **Buy-In Strategy:** Buy-In Strategy is a must-follow for DevOps. Management/team buy-in is also essential as Employee buy-in. Employee buy-in implies the commitment of employees towards the perception, aims, and ideas of the company. Management buy-in is when the stakeholders know the need and importance of above DevOps practices and approaches along with those who execute it. A management buy-in team often competes with other necessary purchasers in the search for a suitable business for DevOps. Although management buy-in is important at different levels of the organization, this DevOps transformation must have buy-in at the leadership level.

The approach we should follow for DevOps is the **Think-Work-Act** approach. Think for the ideas, work on it, find more about it and finally act upon it.

## Barriers

## 1. Legacy System



Legacy systems (Infra and uses both) are unquestionably the best systems that many businesses have relied upon as they run their basic vital functions. Some outdated platforms still survive because of the risk and cost associated with replacing them, even if they get the required skills and support.

These systems were not created, keeping DevOps practices in mind, so they cannot easily fit into the DevOps approach. Upgrading these systems also seems daunting since budgets are confined to ‘the show’ and ‘something new’.

Right now it is necessary to replace these systems with or adopt another system that is made to support DevOps practices or move to some external service provider (mainly SaaS) to decrease software development, maintenance efforts and to support agility. It’s all about including models for legacy infrastructure and applications to make room for DevOps.

## 2. Lack of Executive Buy-In

In layman language, Buy-In is purchasing shares by a broker after a seller has failed to deliver similar shares, the original being charged any difference in the cost. Internal Buy-In for the employees as well as management. Buy-In is much better if a business bought

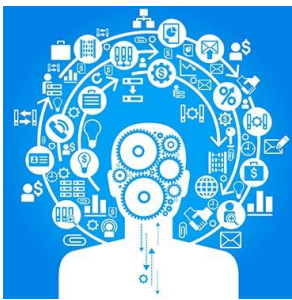




two teams and brought them together to work as one. This is what we call **Buy-In Management**. Although buy-in management is important at different levels of a company, the DevOps transformation must have buy-in at the leadership level. If a business lack this buy-in, there is always a risk of being wrecked before it even begins or eventually a drastic decline in later stages. There should be a willingness at the management level to implement changes.

### 3. Limited IT Skills

Limited IT skills is another one of the biggest barricades when it comes to adopting DevOps. Training the teams right, standardizing the processes, establishing common operational procedures, managing both the teams and striking the balance between the



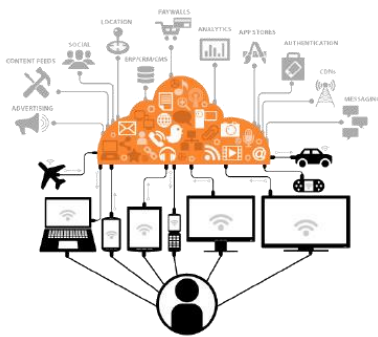
two with the right skillset. Skillfully organizing both teams is a must. There are many different tools, multiple open source projects, but lack of DevOps skills and structured training inhibit adoption is a big threat. Organizing and staffing I&O pros were required for successful DevOps.

## 4. Budget



Usually organizations assume that DevOps tools are open-source that they're free and don't really need the money or financial budgeting but that's not true. There are many factors that need to be considered. There is integration and operation complexity that also needs to be focused after all these things also cost businesses and help them to grow. 2017, was reported as the worst year for under-budgeting for DevOps. Buy-In management also needs a budget plan, which people usually ignore. This needs proper analysis as well.

## 5. Application Complexity



Application complexity is another big challenge. The development team developed the software, the operational team provided the required environment for testing and both teams think their work is done. But both forget the application architecture changes based on **on-perm**, cloud and containers storage and of course on customer's \demands. Application complexity should be kept in mind.

## 6. Fragmented ToolChains



Organizations should avoid fragmented toolset adoption. Toolchain sprawl can impose operational overhead. Fragmented toolsets including multiple open-source tools hinders standardization and slows down adoption.

## 7. Testing Automation



Usually organizations neglect test automation while focusing on CI/CD deployments, which should not be the case. Continuous testing is a very important step in DevOps. Security should not be an afterthought. Organizations should embrace automation for continuous integration and deployment. Test Automation should a “must” not be an “afterthought”.

## 8. Resistance to Change

*“Change in a traditional way is a must.”* Automation must be accepted by organizations



for better projects and for more customers.

Reevaluation and modifications of old processes should be done periodically to make sure the organizations are making forward not backward. With the growing technology, customer needs, economy, opportunities and challenges it becomes inevitable to change and adapt. Their resistance to change does not want them to deviate from a set of routines and comfort since they are familiar with how things work. The need of the hour is to learn to accept change as a supporting arm rather than something to be avoided. If an organization stops experiencing different things, it becomes stagnant and the growth will stop eventually.

## 9. Overcoming the Dev Vs Ops Mentality



The DevOps practice is all about integrating teams together and breaking down silos within IT organizations. There is the old DevOps cliché of developers tossing code over an imaginary wall to operational team and difference in priorities. Devs trying to innovate and make changes as quickly as possible and operations trying to maintain 100% service levels. The truth is the goals of both teams seem to counter each other. They need to align the goals and priorities of the teams. Also handovers between different teams are costly, expensive and a source of delay. Integrating different teams is at the heart of DevOps.

## 10. Culture



Early DevOps initiatives required a collaborative culture, with risk-taking, experimentation and of course overlooking the failures. An organization's focus should be building a collaborative culture with shared goals. This also includes finding employees who are DevOps champions in the organization. Product owners, development team, testing team, and operations team work in different time zones making coordination extremely difficult. Development teams do not wish to understand ops challenges and are limited to working only on dev tasks. The culture of the two traditional teams and mind-set can limit automation.