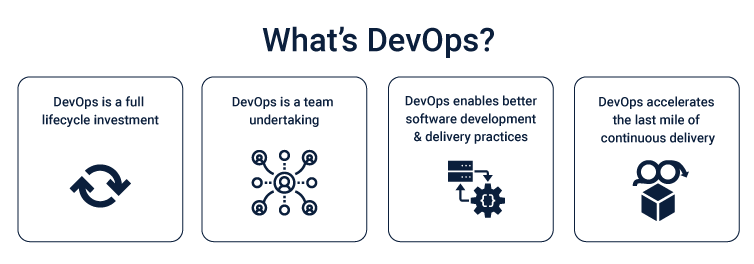
**What is DevOps?**

*DevOps is nothing but a set of philosophies, practices, and tools that help an organization to deliver better products faster by facilitating an integration of the development and operations functions*. This enables companies to serve their customers and markets in a better way and have a competitive edge.

This starts from the design to the entire development process right up to production support.



**Why do we need DevOps?**

Before DevOps, different teams within a company were working isolated from one another. The development team would take their time and develop a product within their test environment and after completing it, would hand it over to the operations team to put it out in the real world. So, there was no intervention between the two teams. There was an imaginary wall between them which plays a major part on performance and efficiency of a business. What it did was, when the development team built a product, they were proud of their work but, when the same product was deployed by the operations team, the product didn’t work as expected at all. Now, what normally happened was that the development team would go on the defensive and assume that if the product works well in their test environment than it must be the operations team’s poor execution. Both teams get frustrated and angry and now they hate their jobs because after so much time and work, they still ended up with a sub-optimal product in the market which would take about the same amount of time to be made somewhat perfect. Which is not really practical as by that time, your competitors might come out with a more appealing and complete product. After the Advent of DevOps, all the teams worked with each other from the get-go. What it does is that, the problems the ops team was going to face after the hard work of dev team, can now be dealt with in parallel to the development of the product. Another useful thing DevOps does for ops people is that, the daily routine tasks which were to be used repeatedly and being done manually, were now being automated helping ops team become more productive and efficient at their work. It boosts the overall performance of the business.

**How Does Using DevOps Benefit an Organization?**

One of the reasons why DevOps saw such rapid adoption is that it truly makes a massive difference to how a tech company operates at a very fundamental level. Let’s take a look at some of the benefits that accrue when a company adopts the DevOps approach.

* **Accelerated Innovation**

This is the major reason that DevOps came into existence. Using DevOps allows companies to develop and deploy products much faster. As we saw in our earlier example, cycle times become significantly longer when there is a wall between development and operations. When the two are integrated, on the other hand, changesets are smaller and problems to be solved each time are less complex. Moreover, team members can make software changes easily since they only need to look at the latest code added and not at all of it. Things like microservices and continuous delivery allow teams to take complete ownership of projects and deliver them faster.

* **Collaboration**

As our example has shown, a wall between development and operations often results in an environment where the two teams don’t trust each other and each is walking around a little blindly. This has long-term repercussions on the morale of the team and how motivated they are to work towards their goals. A DevOps approach results in a collaboration between the two teams where they work with a shared passion to achieve common goals. This creates a much more positive work environment where outcomes can be reached much faster and more efficiently. This also has other positive outcomes like enhanced job satisfaction and lower attrition.

* **Reliability**

Before DevOps updating an application to meet changing user needs was a nightmare. There was always a chance that updating the application would compromise the quality required by the user.

With DevOps tools like continuous integration and delivery, it is now easy to test the functionality of the software and keep security and quality in mind. Other processes like monitoring and logging help keep track of real-time performance metrics which help maintain the reliability of the software.

* **Security**

Without DevOps, you have to often make a tradeoff between speed and security which results in delivery time becoming a lot more. With DevOps, you can use automated compliance policies, fine-grained controls, and configuration management techniques to maintain speed without compromising on security.

* **Scalability**

DevOps started growing in prominence as companies like Google, Amazon and YouTube started finding it harder to manage their technology at scale with a wall separating development and operations. The automation and consistency that comes with DevOps allows you to manage and change complex systems more efficiently.

**What are Some of the Best Practices for Effective DevOps?**

While DevOps still means different things to different people, there has emerged a core of best practices that should be incorporated by companies looking at adopting DevOps.

• **Active Stakeholder Participation**

This is the fundamental guiding principle of DevOps. DevOps can succeed only if both the developers and the operations and support staff are truly committed to collaborating and using an integrated approach to achieve goals.

• **Automated Testing**

Automated regression testing is something agile teams adopt very often as it helps to fix problems right away and ship higher quality code. This works well in DevOps too as a dire need of operations staff is that the code shipped should meet a certain quality standard.

• **Integrated Configuration Management**

In a DevOps environment, configuration management applies not only to the current solution being worked on but also on the configuration issues between the solution and the rest of the organization infrastructure. Integrated Configuration management helps operations teams see the potential impact of a new release more clearly which helps in making better decisions regarding when the release should be made.

• **Integrated Change Management**

With integrated change management, operations and development teams work together to understand how using different technologies will impact the organization as a whole and then work toward managing that.

• **Continuous Integration**

With continuous integration, the code is tested and analyzed whenever updated code is checked into the version control system. This provides immediate feedback on code defects which allows developers to build a high-quality solution with little risk.

• **Integrated Deployment Planning**

A DevOps approach means operations engineers will be closely involved with the developers when it comes to planning the deployment of products as per an organizational deployment schedule.

• **Continuous Deployment**

With continuous deployment, when integration is successful in one sandbox it is automatically promoted to the next sandbox and integration begins there. This continues until it reaches the point where it requires human verification. This usually occurs at the point of transition from dev to operations.

• **Production Support**

With DevOps, not only do developers work on new releases, but they also work on addressing critical problems with a solution that is already in production. Although they are the third and last team to get involved in solving production issues, it is a fairly common occurrence and gives them insights on production problems that help them design better solutions in the first place.

• **Application Monitoring**

This refers to the practice of monitoring and logging solutions real-time once they are in production. This gives us performance metrics that improve the reliability of the solution and prevent failures.

• **Automated Dashboards**

DevOps allows us to create automated dashboards for several key metrics. All metrics cannot be automated of course, but several key metrics can be seen real-time using automated dashboards and they provide critical business intelligence.

**What are the DevOps Tools?**

In order to implement DevOps best practices described above, certain tools have been developed to automate and facilitate different DevOps processes. While the right tools play a key role in effective DevOps implementation, simply using the tools does not mean DevOps adoption. Tools are only relevant when they are used as the last stage- after the organization has already adopted the philosophy of DevOps and there is a commitment to execute its best practices.

Although DevOps was not supposed to be about tools, with its evolution in the last few years, a number of technologies that were not part of the original concept have now become an integral part of DevOps. According to research firm Gartner, a linked toolchain of technologies has now become critical if DevOps is to bring about the change it’s meant to. In recent years there has been an explosion of DevOps tools for different DevOps practices. Here are just a few examples.

**Release Tools**

• Jenkins

• Travis

• TeamCity

• Bamboo

**Configuration Management Tools**

• Puppet

• Chef

• Ansible

• Cfengine

• Saltstack

**Orchestration Tools**

• Zookeeper

• Noah

• Mesos

**Monitoring, Virtualization and Containerization Tools**

• AWS

• OpenStack

• Vagrant

• Docker

• New Relic

• Sensu

• Splunk

• Nagios

**Coding Tools**

• Jira

• Git

• Eclipse

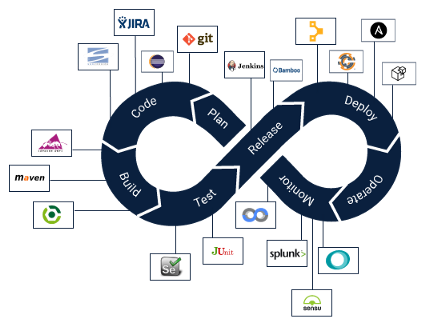
**Testing Tools**

• JUnit

• Zephyr

• Selenium

• Vagrant

• SoapUI  


**The Key to Effective Adoption of DevOps**

Adoption of organization wide DevOps is a slippery slope because it requires a philosophical and cultural change combined with a more practical implementation of tools and best practices. If an organization simply aspires to the philosophy of collaboration and efficiency behind DevOps without doing the hard work of actually executing it on the ground, DevOps will remain a philosophy and nothing more.

At the same time, simply adopting DevOps practices and tools without the philosophy and DevOps culture permeating across the organization is also futile. The starting point of a successful adoption of DevOps within your organization should be getting your development and operations teams fully committed to the cause. It is only after they are fully onboard that the best practices and DevOps tools should come into the picture.