

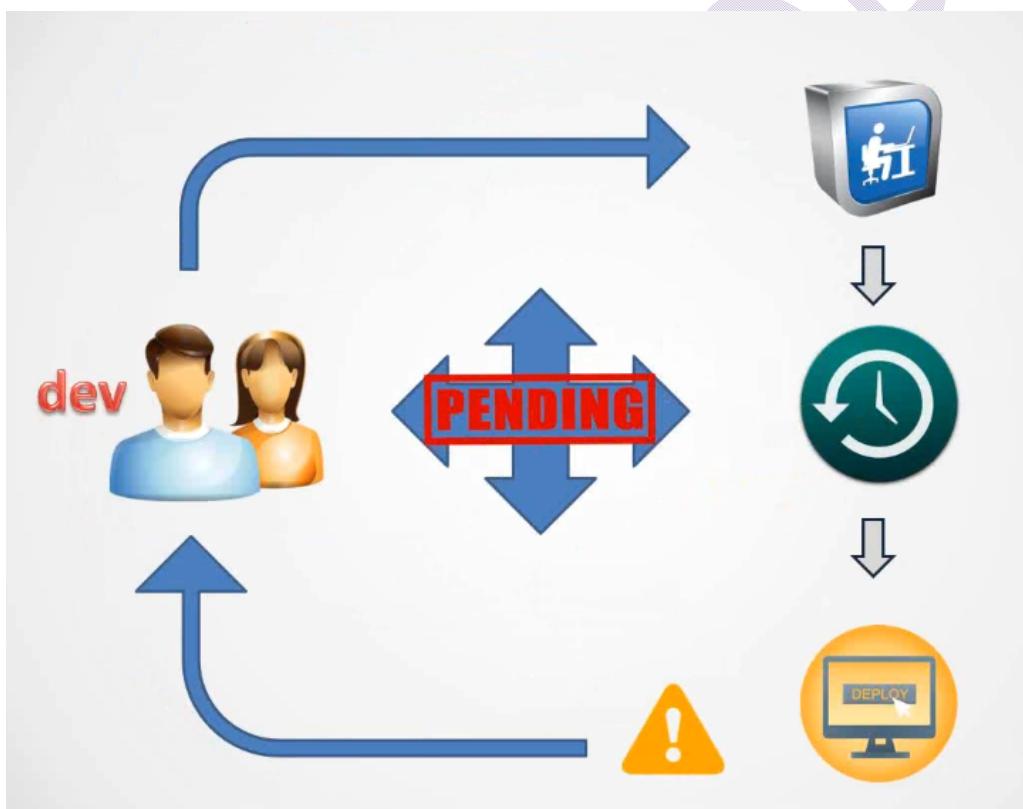
DevOps Introduction

Organization Needs?

- Faster delivery
- Higher quality
- Lesser Spending
- Reduced outages

Traditional development/deployment cycle

- Dev's job is to add new features.



Common issues in Traditional deployment cycle

- Too many feature branches
- Integration issues
- Duplicate verifications
- Bug finding take longer time
- Too many errors

Sysadmin/Operation group activities

- Ops job is to keep the site stable and fast.
- Ops job is enable to the business also. It's same to the developers.



Common Problems Sysadmin/Operation group

- High availability
- Effective maintenance
- Reduce downtime

Why DevOps?

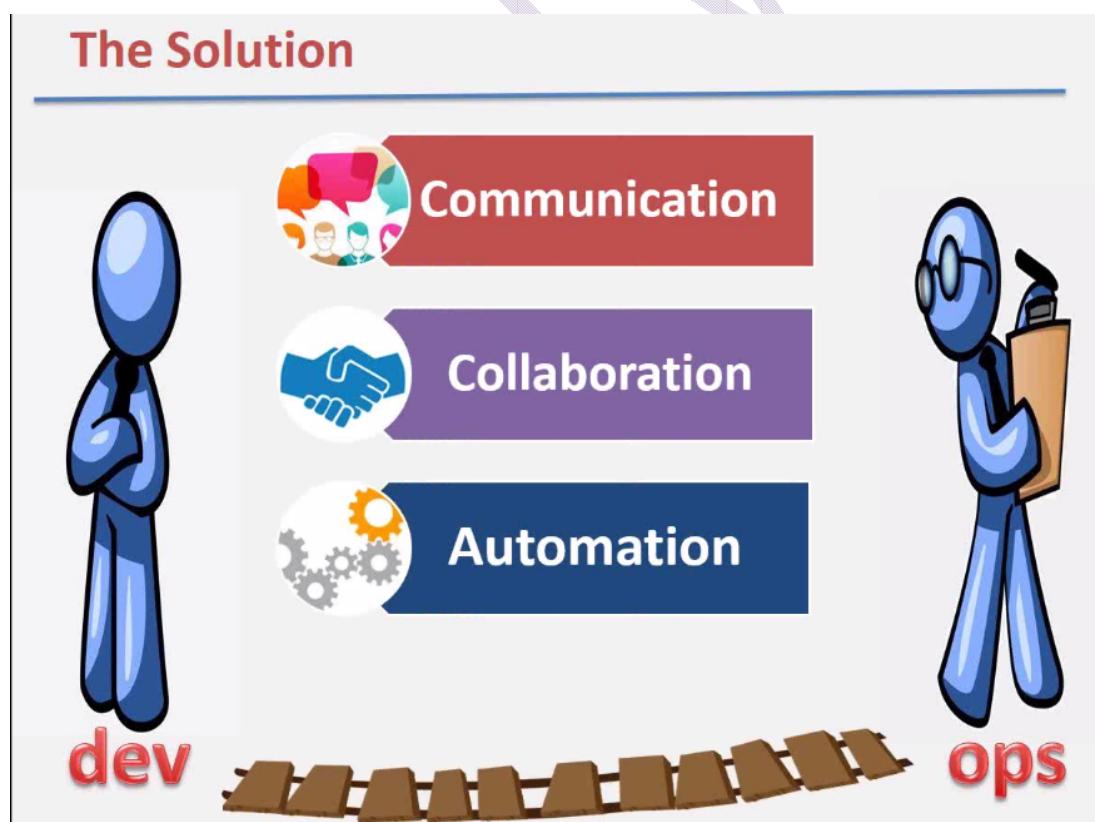
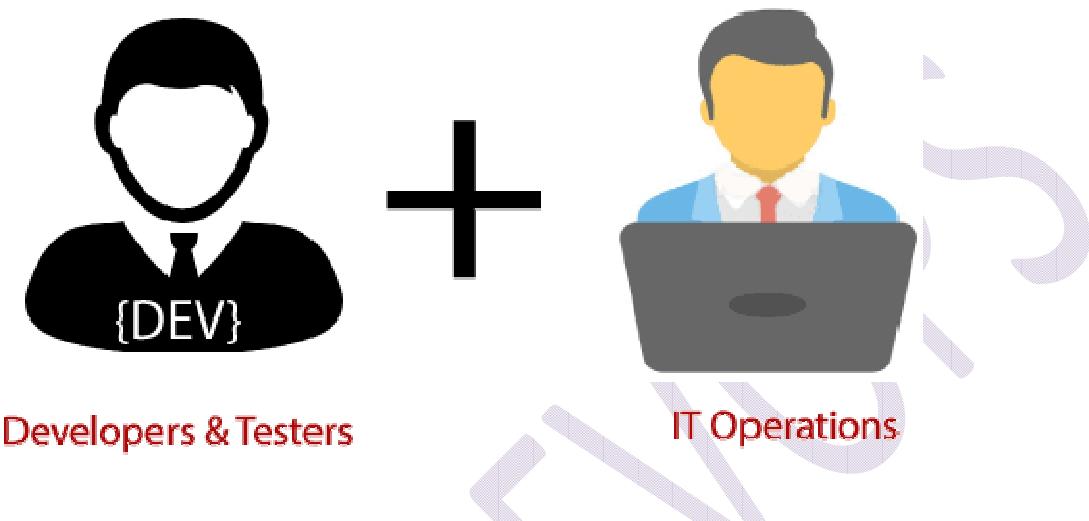
Before going further, we need to understand why we need the DevOps over the other methods.

- The operation and development team worked in complete isolation.
- After the design-build, the testing and deployment are performed respectively. That's why they consumed more time than actual build cycles.
- Without the use of DevOps, the team members are spending a large amount of time on designing, testing, and deploying instead of building the project.
- Manual code deployment leads to human errors in production.
- Coding and operation teams have their separate timelines and are not in sync, causing further delays.

DevOps

The DevOps is the combination of two words, one is Development and other is Operations. It is a culture to promote the development and operation process collectively.

What is DevOps?



DevOps is a set of practices that automates the process between software development and IT teams, in order that then build, test and release software faster and more reliable.

Advantages:

- Continuous software delivery.
- Less complex problems to fix.
- Faster resolution of the problems.
- Faster delivery of the new features.
- More stable operating environments.
- More time available to add value.(Rather than fix/Maintain).
- DevOps is an excellent approach for quick development and deployment of applications.
- It responds faster to the market changes to improve business growth.
- It improves customer experience and satisfaction.

Disadvantages

- DevOps professional or expert's developers are less available.
- Developing with DevOps is so expensive.
- Adopting new DevOps technology into the industries is hard to manage in short time.
- Lack of DevOps knowledge can be a problem in the continuous integration of automation projects.

DevOps History

- In 2009, the first conference named DevOpsdays was held in Ghent Belgium. Belgian consultant and Patrick Debois founded the conference.
- In 2012, the state of DevOps report was launched and conceived by Alanna Brown at Puppet.
- In 2014, the annual State of DevOps report was published by Nicole Forsgren, Jez Humble, Gene Kim, and others. They found DevOps adoption was accelerating in 2014 also.
- In 2015, Nicole Forsgren, Gene Kim, and Jez Humble founded DORA (DevOps Research and Assignment).
- In 2017, Nicole Forsgren, Gene Kim, and Jez Humble published "Accelerate: Building and Scaling High Performing Technology Organizations".

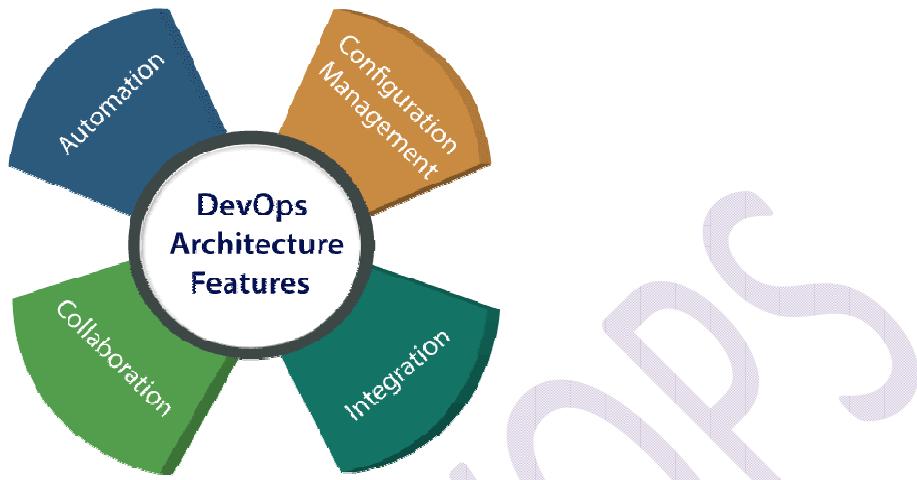
What is the need for DevOps?

Instead of releasing big sets of features, Companies are trying to release small features this has many advantages from customers and better quality of software's.

- Increase deployment frequency.
- Lower failure rate of new releases.
- Shortened lead time between the bug fixes.
- Faster mean time to recovery in the event of new release crashing.

DevOps Architecture Features

Here are some key features of DevOps architecture, such as.



Automation

Automation can reduce time consumption, especially during the testing and deployment phase. The productivity increases, and releases are made quicker by automation. This will lead in catching bugs quickly so that it can be fixed easily. For contiguous delivery, each code is defined through automated tests, cloud-based services, and builds. This promotes production using automated deploys.

Collaboration

The Development and Operations team collaborates as a DevOps team, which improves the cultural model as the teams become more productive with their productivity, which strengthens accountability and ownership. The teams share their responsibilities and work closely in sync, which in turn makes the deployment to production faster.

Integration

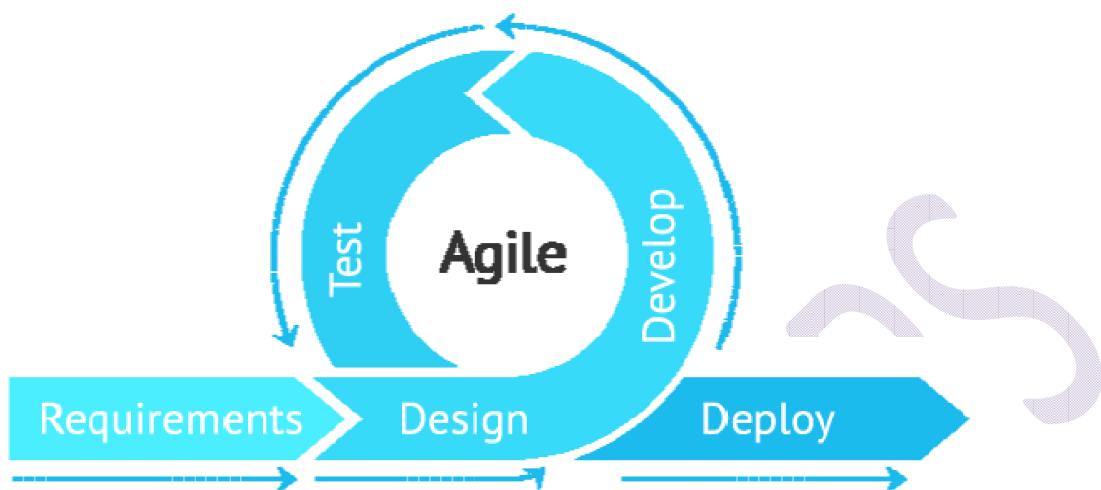
Applications need to be integrated with other components in the environment. The integration phase is where the existing code is combined with new functionality and then tested. Continuous integration and testing enable continuous development. The frequency in the releases and micro-services leads to significant operational challenges. To overcome such problems, continuous integration and delivery are implemented to deliver in a quicker, safer, and reliable manner.

Configuration management

It ensures the application to interact with only those resources that are concerned with the environment in which it runs. The configuration files are not created where the external configuration to the application is separated from the source code. The configuration file can be written during deployment, or they can be loaded at the run time, depending on the environment in which it is running.

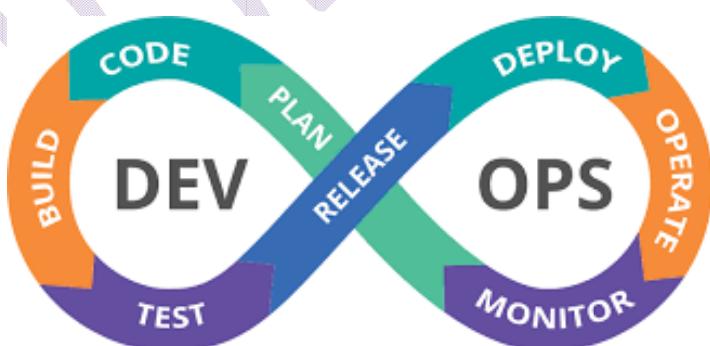
Agile Vs DevOps

Agile



- Agile is about software development.
- Involves practices such as Agile Scrum, Agile Kanban, etc.
- Testing as early as possible.
- Test automates as much as possible.
- Integration is continuous and testing steps forward.
- Potentially code shift at the end of the development.
- It's the link between design thinking and DevOps.
- Timeliness is the main priority.
- Smaller release cycles
- Feedback is from customers.

DevOps



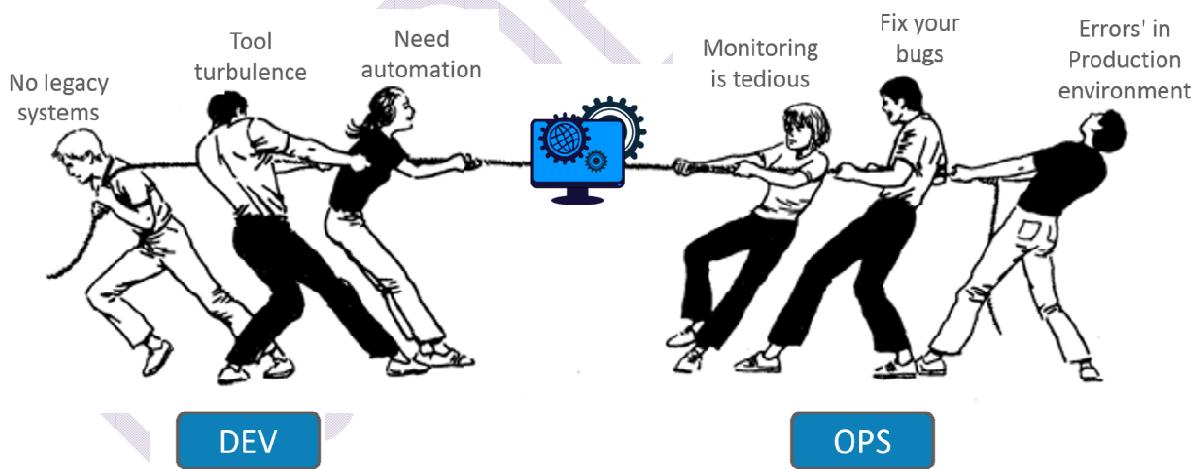
- DevOps is about software deployment.
- Involves processes such as CI, CD, CT, etc.
- DevOps doesn't replace Agile OR Lean.

- Testing Continuously.
- Test automates almost everything.
- Integration is continuously and testing is mandatory.
- Potentially code shift after every integration.
- Timeliness & quality have equal priority.
- Smaller release cycles with immediate feedback.
- Feedback is from self (Monitoring tools).

Why DevOps is better Than Agile?

Agile software development is about following a set of best practices for creating quality software in a timely manner. But the problem is, the best practices followed, involves people working in Silos. By Silos, I mean there are people who will be working as Developers, or as Testers, or as ITOps with very little communication between them. And since, there is very little communication between them; they are not aware what the others are working on despite being a part of the same process. This Silos-ed working of teams is the reason for the infamous “Blame Game” that goes about when software fails or has major flaws.

When a client has complains about a software, the blame is internally thrown at each other. The ‘Dev’ team would point fingers at the ‘QA’ team. ‘QA’ team will then point fingers at the ‘ITOps’ team, who would redirect the blame to the ‘Dev’ team.



Solution

DevOps breaks the Silos right through the middle. In DevOps, the ‘Dev’ team, the ‘ITOps’ team and ‘QA’ team are not independently working pieces of the gamut. But, they are ‘one’.

How DevOps is helpful to Developers?

- DevOps can be helpful to developers to fix the bugs and implement the new features quickly.
- It also helps for clear communication between the team members.

DevOps Life Cycle



Plan:

DevOps use agile methodology to plan the development. With the operations and development team in sync, it helps in organizing the work to plan accordingly to increase productivity.

Code:

Code the application as per the client requirement. With the plan, you have made in the initial step.

Build:

Build the application by performing the integration of various codes you have done in the previous step.

Test:

This is the heart of the application. Test the application that you have built so far. And then rebuilt the application if necessary.

Releases:

If you succeed in the Test phase, then it's time to release the application into Live.

Deploy:

Deploy the code into a cloud environment for further usage. It is performed in such a manner any changes made should not affect the functioning of high traffic website.

Operate:

DevOps changes the way traditional approach of developing and testing separately. The teams operate in a collaborative way where both the teams actively participate throughout the

service lifecycle. The operation team interacts with developers, and they come up with a monitoring plan which serves the IT and business requirements.

Monitor:

Monitor the performance of the application as per the client requirement. Keep a note on the performance of the application. Make modifications if any to satisfy the clients. If does not reach up to the mark make changes in that particular area to satisfy the client.

DevOps Principles

The main principles of DevOps are Continuous delivery, automation, and fast reaction to the feedback.

End to End Responsibility:

DevOps team need to provide performance support until they become the end of life. It enhances the responsibility and the quality of the products engineered.

Continuous Improvement:

DevOps culture focuses on continuous improvement to minimize waste. It continuously speeds up the growth of products or services offered.

Automate Everything:

Automation is an essential principle of the DevOps process. This is for software development and also for the entire infrastructure landscape.

Customer Centric Action:

DevOps team must take customer-centric for that they should continuously invest in products and services.

Monitor and test everything:

The DevOps team needs to have robust monitoring and testing procedures.

Work as one team:

In the DevOps culture role of the designers, developers, and testers are already defined. All they needed to do is work as one team with complete collaboration.

These principles are achieved through several DevOps practices, which include frequent deployments, QA automation, continuous delivery, validating ideas as early as possible, and in-team collaboration.