Overview

The word “DevOps” is the coalescence of two terms “Development” and “Operations”. Today, a majority of the people have a perplexity on the DevOps concept whether it is a culture, movement, approach, or a blend of all these things. In this ***DevOps Tutorial*** covered with all your crux over the DevOps concepts in sequential order. It helps you to have a fair understanding of the DevOps concepts, tools, technologies, & other important approaches.

What is DevOps?

The term DevOps is the blend of two words **“Development” and “Operations”** This is the kind of practice that permits a single team to handle the complete application of the development cycle namely development, testing, monitoring, and deployment. Eventually, DevOps aims to reduce the development life cycle whilst rendering the features, updates, and fixes in harmony with the business goals and objectives.

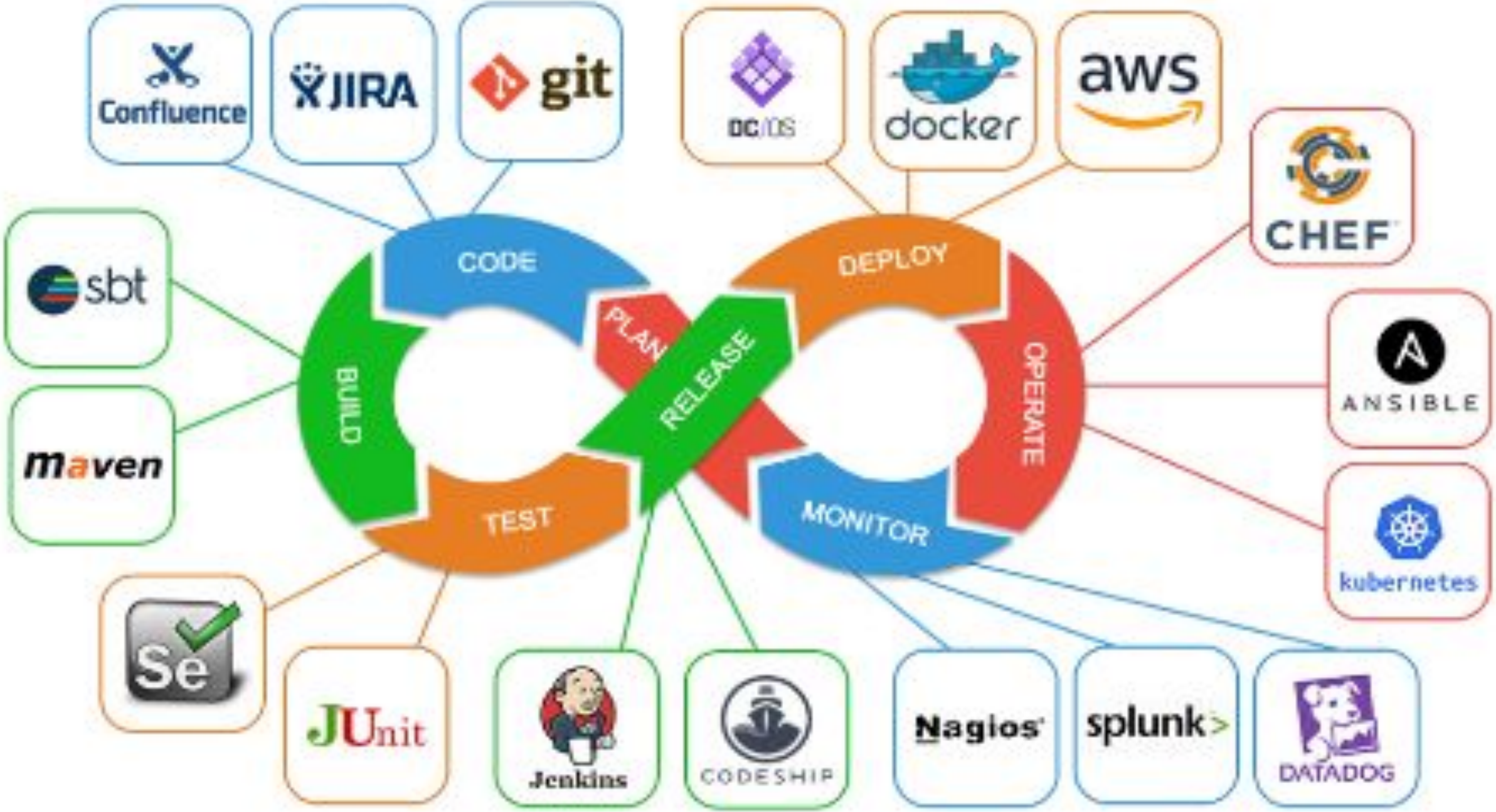
Adopting the DevOps culture alongside its tools and practices, the organizations shall be able to respond to their customer’s requirements at ease and shall boost the performance of the application significantly, and thus support reaching the business goals at a rapid pace. A DevOps consists of different stages and they are –

- Continuous Deployment
- Continuous Integration
- Continuous Testing
- Continuous Deployment
- Continuous Monitoring

Needs to Learn DevOps

DevOps is the method of Software development where the development and the operation team collaborates at every stage of the software development cycle. Below in this **DevOps Tutorial for Beginners**, we have enlisted the important needs to learn DevOps,

- It has made innovative and remarkable changes in the practice of Software Development. The complete team takes part in the development process and it aims for the common goal.
- When there is continuous integration, then there is a consequent reduction in the manual processes that are involved in the development and testing stages.
- It accelerates the chances to work with efficient team members where the knowledge sharing would be significantly higher and helps to have a cordial relationship among the team members.
- Last but not the least, with never-ending changes in the IT industry, the demand for skilled DevOps professionals is expected to increase tremendously.



DevOps Lifecycle

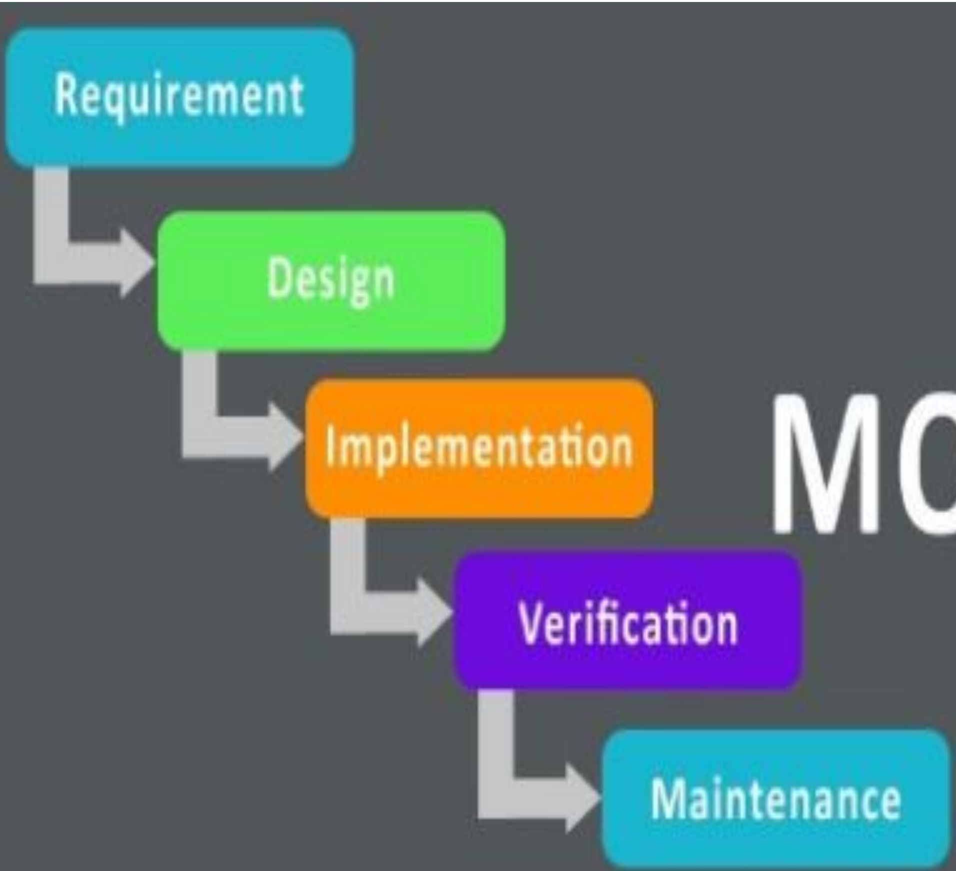
History of DevOps

Having seen what DevOps is and the need to learn DevOps. In this **DevOps Tutorial**, we will guide you deep into the Core of DevOps and how it came into the scenes in the industry. Before introducing the DevOps the Software Development industry had two main approaches and they are – Waterfall and Agile model of development.

Waterfall Model

- This is the kind of Software Development model which is direct and linear. The Waterfall Model follows the approach of the top-down method.
- Also, this model has different starting with the requirement analysis and gatherings. This is the stage where you gather all the requirements from the clients to develop the application.
- The next stage is the Designing stage where you are required to prepare the Blueprint for the software. Here, you can sketch out the designs as how your software is going to look like.

WATERFALL MODEL USE CASE



Requirement

Design

Implementation

Verification

Maintenance

- Once when the designs are done, they can further be mapped to the Implementation phase, where the coding for the application starts. And here, the team of developers coordinates and works together on different components of an application.
- In case if you are done with the application development process, then now you are required to test the developed application in the verification stage. There are different tests conducted on the applications namely integration testing, unit testing, and performance testing.
- Once when you are done with all the test processes of the application, then they are deployed in the production servers.
- Finally, comes the Maintenance stage, here in this stage the application is scrutinized to ensure its performance. Also, any issues that are associated with the performance of an application are dealt with in this phase.

Upsides of the Waterfall Model

- It is simple and easy for understanding and using it
- It allows performing analysis and testing
- It saves maximum time and money
- This is mainly preferable for smaller projects
- The Waterfall model allows the access of Managerial and Departmentalization of control

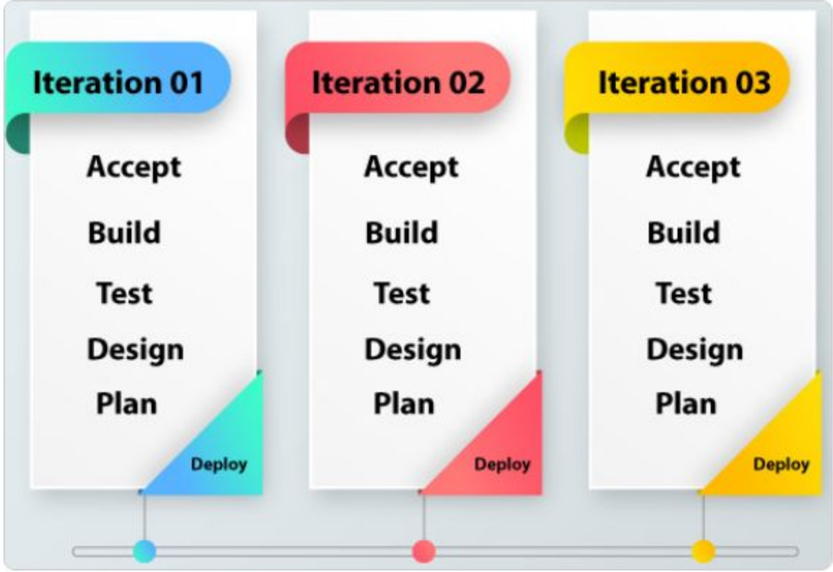
Downsides of Waterfall Model

- It is highly uncertain and risky
- There is a lack of visibility on the current progression
- It is not preferable when the requirements constantly modify
- It is strenuous to make changes in the products when they are in the testing phase
- You can find the end product only after the complete end of the cycle
- It is not recommended for complex and larger products

Agile Methodology

It is an iteratively based software development approach where the software project is broken into different **sprints and iterations**. All the iteration consists of the phases that are found in the waterfall model namely – gathering, requirements, designs, testing, development, and maintenance. The total span of every iteration is 2-8 weeks.

Process of the Agile Model



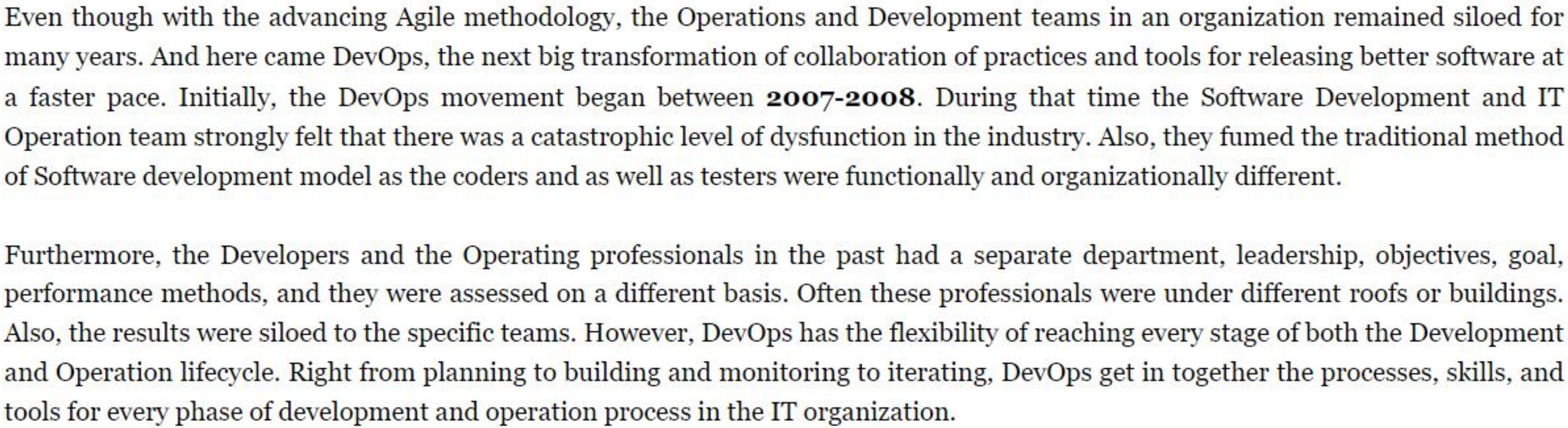
- The Agile company launches the application with high-priority features in its first iteration.
- Once the release is over, the customers or the end-users provide feedback on the performance of an application
- Also, we can make the required changes that are found in the application with features and applications that are launched again with the second edition
- You could repeat the complete process until you have reached the specific software quality

Upsides of the Agile Model

- This response to the needs and changes more adaptively and suitably
- Also, fixing the errors at the early phase of development helps the process to be more cost-efficient
- It enhances the quality of the product and thus makes them error-free
- It permits direct communication among the people who are involved in the Software Development project
- It is highly recommended for the big and long-term projects
- Also, the Agile model needs only minimum resources and this is very easy to handle

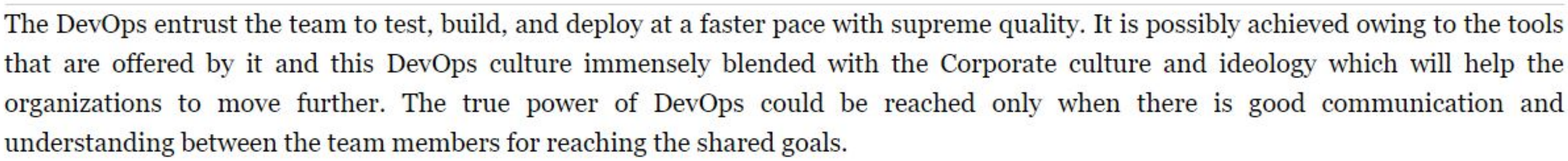
Downsides of the Agile Model

- It is highly dependent on the customer needs
- It is not easier to predict the effort and time of the bigger projects
- It is not preferable for complex projects
- It struggles more with the document efficiency
- It helps in maintaining the risks



Even though with the advancing Agile methodology, the Operations and Development teams in an organization remained siloed for many years. And here came DevOps, the next big transformation of collaboration of practices and tools for releasing better software at a faster pace. Initially, the DevOps movement began between **2007-2008**. During that time the Software Development and IT Operation team strongly felt that there was a catastrophic level of dysfunction in the industry. Also, they fumed the traditional method of Software development model as the coders and as well as testers were functionally and organizationally different.

Furthermore, the Developers and the Operating professionals in the past had a separate department, leadership, objectives, goal, performance methods, and they were assessed on a different basis. Often these professionals were under different roofs or buildings. Also, the results were siloed to the specific teams. However, DevOps has the flexibility of reaching every stage of both the Development and Operation lifecycle. Right from planning to building and monitoring to iterating, DevOps get in together the processes, skills, and tools for every phase of development and operation process in the IT organization.



The DevOps entrust the team to test, build, and deploy at a faster pace with supreme quality. It is possibly achieved owing to the tools that are offered by it and this DevOps culture immensely blended with the Corporate culture and ideology which will help the organizations to move further. The true power of DevOps could be reached only when there is good communication and understanding between the team members for reaching the shared goals.

Applications of DevOps

DevOps is not only used by the developers and operators. Rather this is used by the administrators such as project managers, test engineers, and in the different segments by the administrators. There are numerous practices that permit the organization to offer faster and more reliable updates for its customers. The DevOps core aspect revolves around the Agile principle, which is the significant influencer of DevOps concept creation. Below we have listed down the applications,

Microservices: It is a well-planned approach that supports building single apps as a package for small services. All the services in these applications are capable of communicating through well-defined interfaces. It utilizes the lightweight mechanism which is mostly the HTTP-oriented API.

Infrastructure as Code: This is the practice on which the Software Development codes and techniques are built and the infrastructure is handled. Usually, the developers and system administrators shall communicate through the API-driven models of the Cloud. Rather than configuring and setting up them manually, the IaC communicates with the infrastructure programmatically.

Monitor and Logging: Mostly the enterprises inspect the logs and metrics to check the roots and the application's performance to measure the end-user experience. Also, active monitoring is crucial to ensure that the services are there 24/7 without any interruptions.

Continuous Integration: It constantly indicates the repeated testing and merging. Continuous integration intends to identify the bugs and defects at an earlier stage to enhance performance. Also, it significantly reduces the time that is taken for validating and releasing the updates on the software.

Collaboration and Communication: One of the prime goals of DevOps is to promote better collaboration and communication. The automation and tooling of the software render the process by bringing in the workforce under one roof of operation and development. It helps in boosting the communication among the departments. Also, teamwork permits the potent accomplishment of any assigned tasks.

DevOps in Networking: The doctrine of the DevOps concept is alluring and it is harvesting unprecedented popularity in handling Networking services. With the aid of the vendor hardware, deployment modes, automation of the network functions, devices, and configuration tools it has just become an easier job for the professionals to deploy it.

DevOps in Data Science: The companies are persistently working hard to become more buoyant. And thus more organizations are switching to DevOps to deploy the codes robustly and efficiently. And it makes use of the integrated method and successively deploys the plan of the **Data Science** in the production. Also, it uses the perfect direction for robust implementations.

DevOps in Testing: Based on a survey report gathered from the **RightScale**, it is stated that numerous companies have preferred DevOps for testing. It is done for reaching agility and speed as it is more essential to automate the complete process of the configuration and testing. The complete function of DevOps is entrusted on the **"Agile Manifesto".** And yet the root of the proficient strategy is called the **"DevOps Trinity"** and it is:

DevOps in Testing: Based on a survey report gathered from the **RightScale**, it is stated that numerous companies have preferred DevOps for testing. It is done for reaching agility and speed as it is more essential to automate the complete process of the configuration and testing. The complete function of DevOps is entrusted on the **"Agile Manifesto".** And yet the root of the proficient strategy is called the **"DevOps Trinity"** and it is:

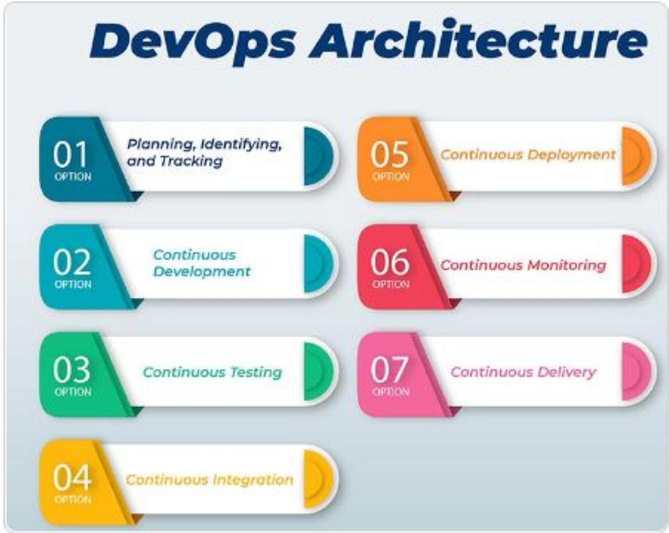
People and Culture: When you adopt DevOps, it helps in eliminating all the differences among the teams. Also, they mainly work on a common goal. The main purpose of DevOps is to gain quality software.

Tools and Technologies: DevOps is one of the sustainable and adaptable models with a range of technology and tools. It allows the complete process of the operation and development to be a much easier one.

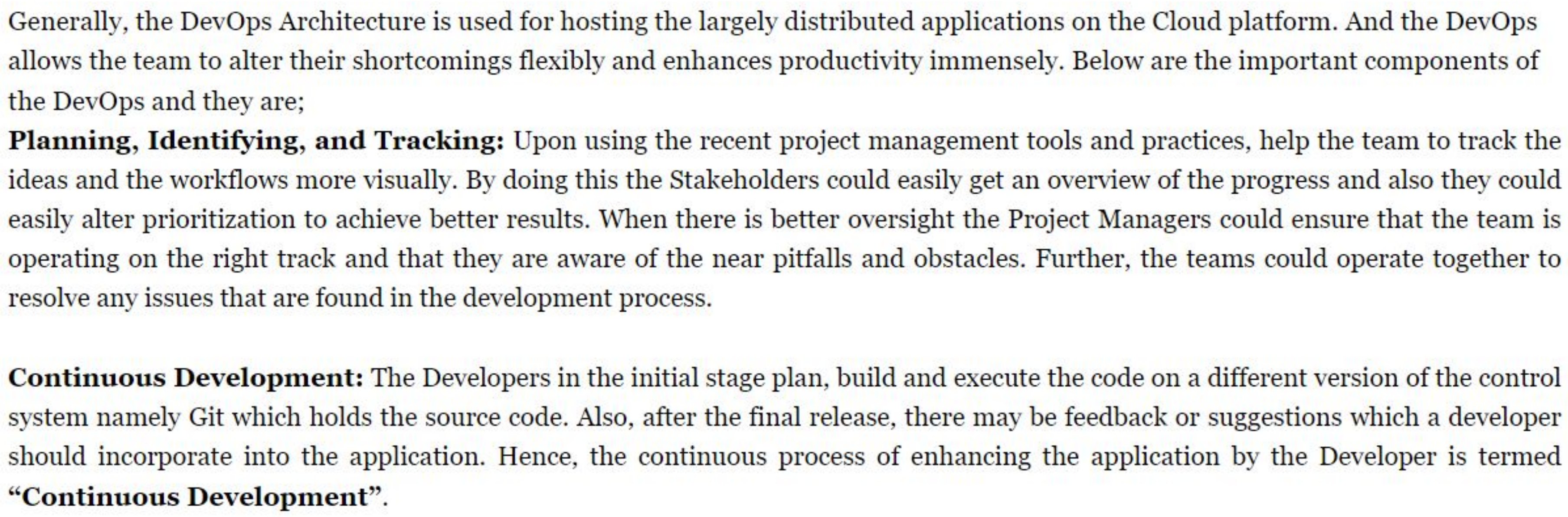
Processes and Practices: Both the DevOps and the Agile go together. Thus, on the deployment of Agile, **Scrum**, Kanban, or Plus automation the organizations were able to streamline the processes in a replay.

DevOps Architecture

In the arena of Software Engineering, both the Development and the Operation team occupy a vital role in application delivery. Generally, the Development team constitutes the administrative services, processes, and support of the software. When both the Development and the Operation teams are joined together for collaborating, the DevOps architecture comes into play. The DevOps concept is the only way to bridge the gap between the Development and Operation teams so that the delivery of the software could be achieved rapidly with lesser issues.



Generally, the DevOps Architecture is used for hosting the largely distributed applications on the Cloud platform. And the DevOps allows the team to alter their shortcomings flexibly and enhances productivity immensely. Below are the important components of the DevOps and they are;



Generally, the DevOps Architecture is used for hosting the largely distributed applications on the Cloud platform. And the DevOps allows the team to alter their shortcomings flexibly and enhances productivity immensely. Below are the important components of the DevOps and they are;

Planning, Identifying, and Tracking: Upon using the recent project management tools and practices, help the team to track the ideas and the workflows more visually. By doing this the Stakeholders could easily get an overview of the progress and also they could easily alter prioritization to achieve better results. When there is better oversight the Project Managers could ensure that the team is operating on the right track and that they are aware of the near pitfalls and obstacles. Further, the teams could operate together to resolve any issues that are found in the development process.

Continuous Development: The Developers in the initial stage plan, build and execute the code on a different version of the control system namely Git which holds the source code. Also, after the final release, there may be feedback or suggestions which a developer should incorporate into the application. Hence, the continuous process of enhancing the application by the Developer is termed **“Continuous Development”**.

Continuous Testing: Once the code is uploaded into the Source code platform it undergoes the testing phase. Here in this phase, every time the codes are tested and the necessary changes are implemented on those codes before pushing them to the production team.

Continuous Integration: When you complete one stage of the DevOps lifecycle, the application code should be moved immediately to the next stage. This could take place with the aid of the integration tool. The Development practice persistently harmonizes the code right from the first stage to the next level with the support of tools and they are called continuous integration.

Continuous Deployment: After the addition of every feature on the application, it would need a few modifications to the application environment. It is called Configuration Management. To attain this, we should make use of more deployment tools. The process of continuous change on the application environment based on the recent addition of the attribute is termed Continuous Deployment.

Continuous Monitoring: After every testing and planning the bugs may identify their means to the production. We could keep track of those bugs or other inappropriate system behavior. Also, we could keep track of the feature request and monitor the tool persistently to check when and how the application goes through the updates.

Continuous Delivery: Last, but not least, the DevOps architecture is developed on the motto of Continuous Delivery. This means that any practice that is set on the play shall foster collaboration and communication among the teams and it should function toward the constant and routine delivery of the tested software. It could be automated just like continuous deployment as mentioned above.

Advantages of DevOps Architecture

The properly implemented DevOps approach comes with more benefits. It includes the following and they are

Cost Reduction: One of the primary concerns for any business would be the operational cost. The DevOps aids the organizations to keep their cost or expenses at a lower range. Owing to efficiency it gets the boost with the DevOps practice and the Software production enhances the business performance to foresee the overall decrease in the cost of production.

Improved Productivity and Release Time: With the curtailed streamlined processes and development cycles, the teams become more productive and the software is deployed robustly.

Efficient and Time-Saving: The DevOps eases the lifecycle with the earlier iterations that have been growing complex over time. However, with DevOps, the organizations could gather all the requirements at ease. The thing to be noted here is that, in the DevOps, the process of collecting the requirements is streamlined and a culture of collaboration, accountability, transparency meets the requirements in a smooth sail with team efforts. And with this practice, an organization could achieve anything.

Customer Satisfaction: The User Experience and the User feedback is the most important accept of the DevOps culture. By collecting all the details from the clients and then acting based on it helps to ensure that the client’s requirements and needs are fulfilled completely to reach new heights in achievements.

Principles and Workflow of DevOps

DevOps was earlier called the mindset and the culture which strongly withholds the collaborative bond among the infrastructure operations and the software development teams. This culture is fundamentally built on the below principles

Gradual Changes: The utilization of the gradual rollouts permits the delivery teams to release the product of the users to have the opportunity to use the updates and the rollbacks when something goes wrong.

Constant Communication and Collaboration: This is the building block of the DevOps concepts ever since its inception. Both the Operation and Development teams should function cohesively and collaboratively to comprehend the requirements and expectations of all the members of the organization.

Sharing of end-to-end Responsibility: All the members of the team should march towards one specific goal and this is responsible equally for a project right from the very beginning till the end that is to facilitate and aid with the need of other member's tasks.

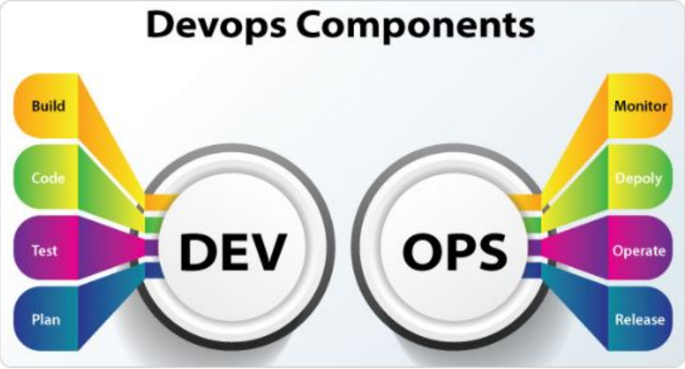
Ease of Problem-solving: DevOps would require the tasks that have to be performed in the early stage of the project lifecycle. So, DevOps mostly concentrate on the tasks of these types and lays the efforts to address these issues more quickly.

Measuring KPIs (Key Performance Indicator): Usually, the Decision-making process must be powered with the factual information in the first stance. It is important to keep track of the progress and activities that make up the DevOps workflow in order to achieve optimum efficiency. To measure the different metrics of the system you should first allow the system to have an understanding of which goes well with the system and what else could be done to enhance the performance.

Automation of Processes: The golden rule of DevOps is to automate as many things as possible like testings, configurations, deployment procedures, and developments. It permits the Specialists to overcome the time-consuming and repetitive work and thus to focus on the other essential activities that could not be automated by nature.

Sharing: The DevOps philosophy in actual highlights the common English phrase **"Sharing is Caring"**. The DevOps culture highlights nothing but the significance of the collaboration. It is a crucial aspect of any work to share feedback. We can also say that it is one of the best practices to widen your knowledge and skills among your teams and this eventually promotes transparency that develops more collective intelligence and excludes constraints significantly. Also, on adopting the DevOps process, you need not just stop a development process just because only a single person could handle the task efficiently and that person is out of work due to some reasons.

DevOps Process :



Plan: It is the part of the development process where you would organize the schedules, tasks, and setting up of your project management tools. The primary idea is to plan the tasks by using the user story of the process right from the agile methodology. We can also write the tickets in the method of user store for permitting the Operational engineers and Developers to comprehend the development requirements and why it is done.

Code: In this stage, the developer performs coding and reviews the code completely. When the code is ready you can merge them easily. Also, in the DevOps practice is it more important to share the code among the Developer and the Operating engineers.

Build: It is the first where one moves towards automation. The aim here is to build the source code in one desired format, testing, compiling, deploying in the specific place of an infrastructure. Once the setup is done then CI and CD tools could be verified with the support of the Source Code Management and build them.

Test: On performing the Continuous testing process, the organization could easily reduce the risk. The Automatic test makes sure that there are no bugs that are implemented in the production stage. We can implement those testing tools in the workflow to ensure that the best development of quality software is produced.

Release: Every code has to be passed in the testing process only then it is considered to be ready for deployment.

Deploy: The Operational Team deploys the new feature on the production. However, automation is one of the major principles of DevOps and this is possible only to set up with continuous deployment.

Configure/Operate Infrastructure: The Operation team develops or maintains the scalable infrastructure and the infrastructure as code to evaluate the log management and security control issues.

Monitor: This is the most important step in the DevOps that permits fixing all the incidents which have issues at a faster pace. This eventually enables the users to have a better user- experience.

DevOps Automation

Automation is the key factor that occupies an important role in DevOps. The question here is how you can in reality place the automation in the practice to advance your goals in DevOps. Here in this **DevOps Tutorial** session, we have explained in-depth what automation indicates here and the context of automation in DevOps, and the different practices that could be automated to reach the DevOps Automation.

With the never-ending evolution in the technology field, Software Development teams are always under pressure to cope with the growing demand and customer expectation of the business applications. The general expectations are:

- Enhanced Performance
- Extended functionality
- Offer guaranteed uptime and availability

Also, all the Traditional Software Development processes have shifted themselves with the cloud-based applications with the advent of technology. The present paradigm is focused more on developing the software more as an ongoing service instead of creating them simply for specific customer requirements. Software development has come a long way from monolithic to agile structure, where it is possible to develop the software constantly.

Automation

In DevOps, the term automation means getting rid of the need of human engineers for intruding physically to facilitate DevOps practices. In the conceptual aspect, we could perform the DevOps processes namely **Continuous Integration(CI), Continuous Delivery(CD),** and log in the analytics manually. In doing so you may need a bigger team, a huge time, and a high level of coordination and interaction among the team members that are more relevant to the situation prevailing in the organization. However, with automation, you could perform all these processes by using a predefined set of tools and configurations.

What is DevOps Automation?

DevOps Automation is the method of automating the repetitive and mundane DevOps tasks that could be executed without any intervention on humans. The Automation could be practiced in the entire wheel of the DevOps lifecycle and they are

- Software Deployment & Release
- Design & Development
- Monitoring

The main intent of DevOps Automation is to standardize the DevOps cycle by reducing the manual workload. Thus, the automation results in more key improvements,

Advantages of Automation in DevOps

The Automation renders an array of benefits that helps to reach the goal of the DevOps at ease.

Consistency

Usually, the processes are highly automated and are persistently predictable. The Software Automation tool would always perform a similar thing until they are configured again to do the same thing. However, this is not applicable in case if a human is working there.

Scalability

Generally, Automation is considered to be the mother of scalability. Also, these processes are more flexible to handle numerous processes when compared to the manual way of scalability. To brief, consider the instance when you are working manually, you will be able to deploy the new releases only when you are dealing with one specific application or environment. However, when your team handles different applications and that the application is being deployed to a different environment that is more than one Cloud or OS, you can release the newer codes rapidly and consistently.

Speed

The automation here means the processes such as Code Integration and thus the Application deployment occurs at a rapid pace. To adhere to the above statement let us consider some scenarios. When you have automation deployed, you may not wait for the required person to process it, you can just simply deploy the new release or update irrespective of the time and dependency of a person. With the aid of automation tools, you could easily overcome the delay factor.

Secondly, with the in-built automated processes, you could execute the assigned work more rapidly. Over here when you have an Engineer employed, the Engineers should imply some of the criteria like checking the environment, typing out the configuration, and physically checking whether the latest version was deployed successfully. In contradiction, the automation tool could perform the operations more instantly.

The Things to be Prioritized for the DevOps Automation

Many processes and practices are found in the DevOps and it shall differ from one enterprise to the other. Here in this **DevOps tutorial**, we have jotted down some of the common processes that could be prioritized for the automation process.

Software Testing: In the testing process, before releasing the Software it has to undergo some testing process. However, when you perform this manually you would need more time and workforce. You can overcome this obstacle with the aid of automation test tools like **Appium** and **Selenium** With these tools, the **Software Testing** process is way easier and the test could be performed in a proper routine.

CI/CD: Rapid application development and delivery is the core or central theme of the DevOps concept. Also, it is much more difficult to reach the goal when you don't automate the CI – Continuous Integration, CD – Continuous Delivery process.

Monitoring: There is one major hindrance in the DevOps environment is that to keep the track of all the components in the rapidly moving environment. The automation tools could be used for checking the performance, availability, and security issues that generate the alerts based on the ability to resolve an issue.

Log Management: The total amount of the log data is developed by the DevOps environment and that it is more widespread. The process of gathering and analyzing every data by hand is not possible for many teams. Rather, we can rely on the log management solution which could robustly cumulate and analyze the log data.

The Popular DevOps Automation tools

In the case of automation, more software options are available. Both the open-source and the licensed tools support the complete automation of the DevOps pipeline. Among them, the most widely used type is the CI/CD tools.

The Chef and Puppet are used for cross-platform configuration management. These tools primarily handle the deployment, configuration, infrastructure management, automation, and management of the infrastructure.

TeamCity, Jenkins, and Bamboo are the popular CI/CD software that automates the tasks right from the beginning of the pipeline till the deployment stage. Apart from this, there are specialized tools and software that aims at a single function and this is the most crucial aspect of the DevOps pipeline

Infrastructure Provisioning: Terraform, Vagrant, Ansible

Containerized Applications: Docker, Kubernetes

Source code management: CVS, Subversion, and Git

Application/ Infrastructure Monitoring: QuerySurge, and Nagios

Security Monitoring: Splunk, Suricata, Snort

Log Management: Datadog, SolarWinds Log Analyzer, and Splunk