Continuity is the heart of the DevOps lifecycle and people often get confused between the terms Continuous Delivery and Continuous Deployment. In this blog on Continuous Delivery vs Continuous Deployment, I’m going to compare these two highly conflicting terms in the industry.

So, before I deep dive into differentiating these terms, let me brief you about DevOps first.

**What is DevOps?**

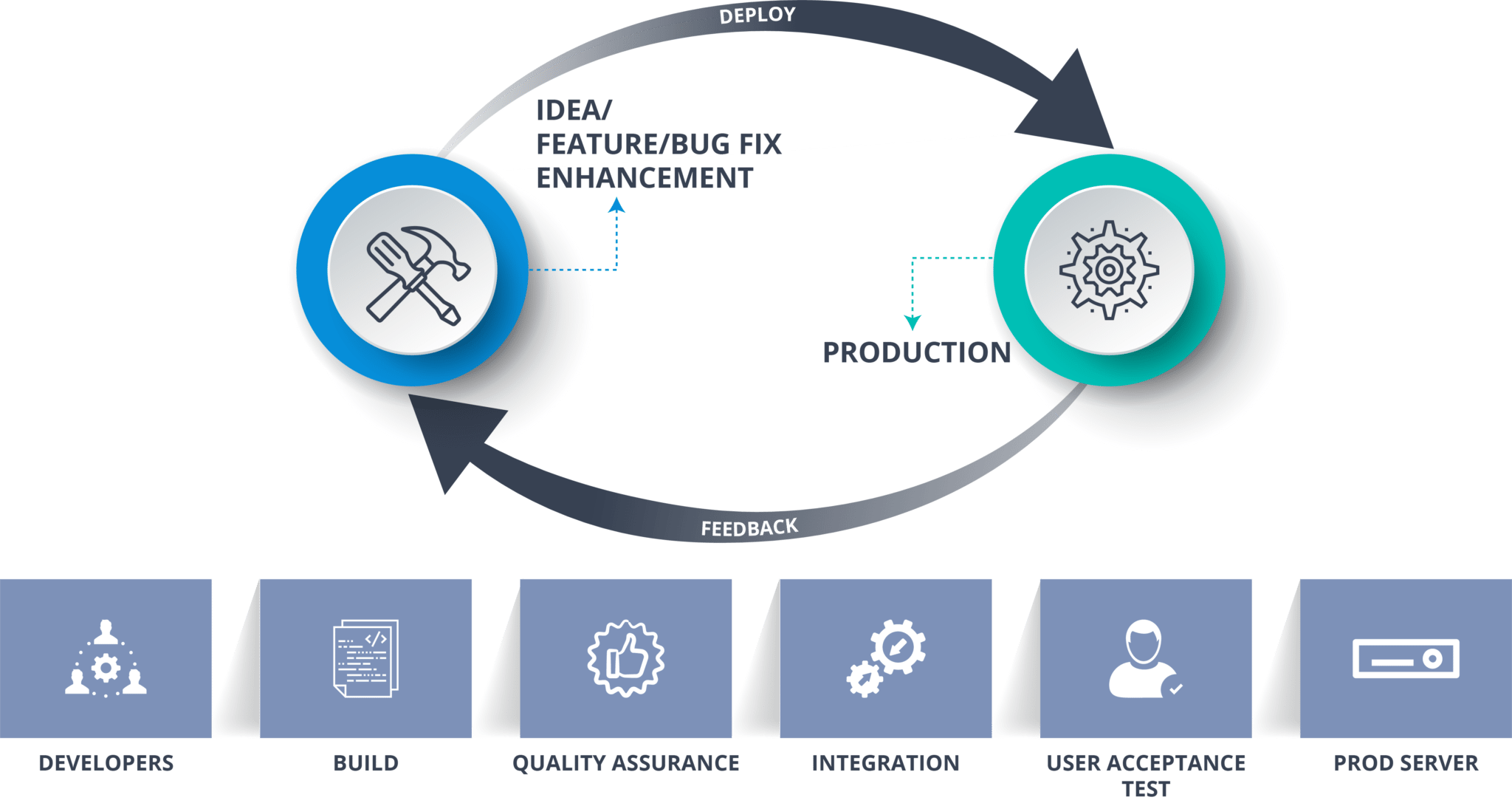
[***DevOps***](https://www.edureka.co/blog/what-is-devops/) is basically a software development strategy which bridges the gap between the dev side and the ops side of the company. In simple terms, you can say that DevOps is, how a developer gets a new feature, an enhancement request, or a change out to production, so that,  when the customers give feedback, the developers can improve based on that feedback.

**But, what are the factors that developers can work on?**

Well, developers can work on mainly 3 important factors:

* The software which is delivered.
* The efficiency and performance of environment to which the software is delivered.
* Making the process of delivering software more efficient, capable and faster at a lower cost.

So, when you deliver software it’s not just delivering it to the production, but,  there’s an entire software delivery lifecycle involved with it.



DevOps Methodology – Continuous Delivery vs Continuous Deployment

As you can refer to the diagram, the developers build the software and store it in a software configuration management or version control system. After that, the QA environments assure the quality, and the system integration test, user acceptance test is performed. Finally, when the software passes through all these stages, it reaches the production where the software actually runs and customers interact with it.

Now, the environments at which developers are working might be the same as the environments, customers are working in, but the configurations may differ. So to match these configurations, automated deployment is required.

Automated deployment is the ability to get software deployed in any environment at any given time and Continuous Delivery is that capability to deploy this software to any particular environment at any given time.

Now, you might have heard about large web companies deploying changes every day, all the way onto their prod servers.

**How do they do deploy so many times so frequently?**

Basically, any change that a developer makes, gets deployed all the way to production and this is nothing but known as Continuous Deployment.

Want to explore more about DevOps?

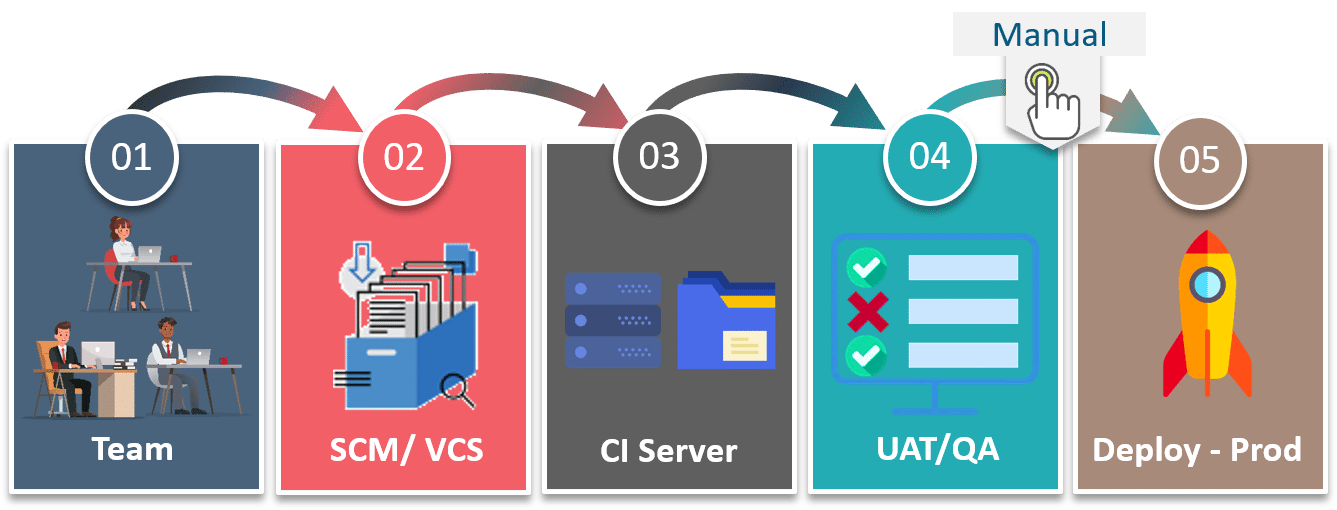
**Continuous Delivery vs Continuous Deployment**

Now that you have a basic understanding of both the terms let me define Continuous Delivery and Continuous Deployment for you.

**Continuous Delivery**

Continuous Delivery is a software development practice where you build software in such a way that the software can be released to the production at any time.

You achieve Continuous Delivery by continuously integrating the products built by the development team, running automated tests on those built products to detect problems and then push those files into production-like environments to ensure that the software works in production.



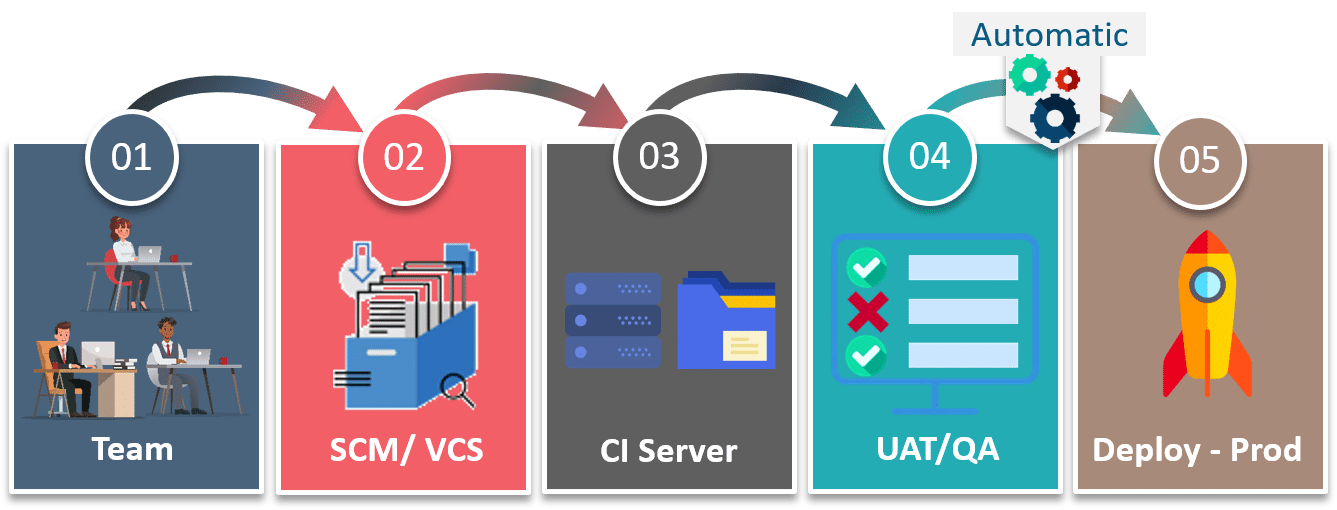
Continuous Delivery – Continuous Delivery vs Continuous Deployment

The benefit of continuous delivery lies in the fact that the code is ready to deploy at all times. So, as you can see here the Quality Assurance team tests if each feature is working or not, and then they manually deploy it to production based on the need of business increasing the quality and velocity of the product. So, each and every change is not deployed on to the production.

Now, let me tell how different Continuous Deployment from Continuous Delivery is.

**Continuous Deployment**

Continuous deployment means that every change that you make, goes through the pipeline, and if it passes all the tests, it automatically gets deployed into production. So, with this approach, the quality of the software release completely depends on the quality of the test suite as everything is automated.



Continuous Deployment – Continuous Delivery vs Continuous Deployment

For example, if you have a function to check various conditions in the test suite, then in Continuous Delivery a manual test can be performed to check the quality of the function. So, if anyone finds out that there could more cases included in that particular function, then it would not be deployed on to production.

But, in the case of Continuous Deployment there would be no approval required, so that function would be automatically deployed on to the prod servers.

So, if we have to summarize in a single line, then, in the world of DevOps using Continuous Deployment, there’s no release approval required. So, the code moves automatically from the developer site to the production site, which is not the case with continuous delivery.

It is always recommended that we should not use, Continuous Deployment as we need to consider many factors before releasing the software like marketing the product before it’s out to the world,  but we must do Continuous Delivery so that we have the capability to deliver the software to any given environment at any given time.

Have you ever wondered if there is any case of Continuous Deployment with Continuous Delivery?

**Continuous Delivery with Continuous Deployment**

Well, there are many cases where you use both of them. Let me show you one example:

Consider a situation where a customer finds out a bug in the software and sends a feedback to the Dev team.

The Dev team has to recreate that bug as quickly as possible and then fix it. So, in such emergency situations if the team uses continuous delivery then they have the ability to provide that environment which has the same configuration of that of the customer, to deploy the right version of the software to that environment and use automated testing to assist every change. Since the final steps are fully automated Continuous Deployment quickly solves the problem.

