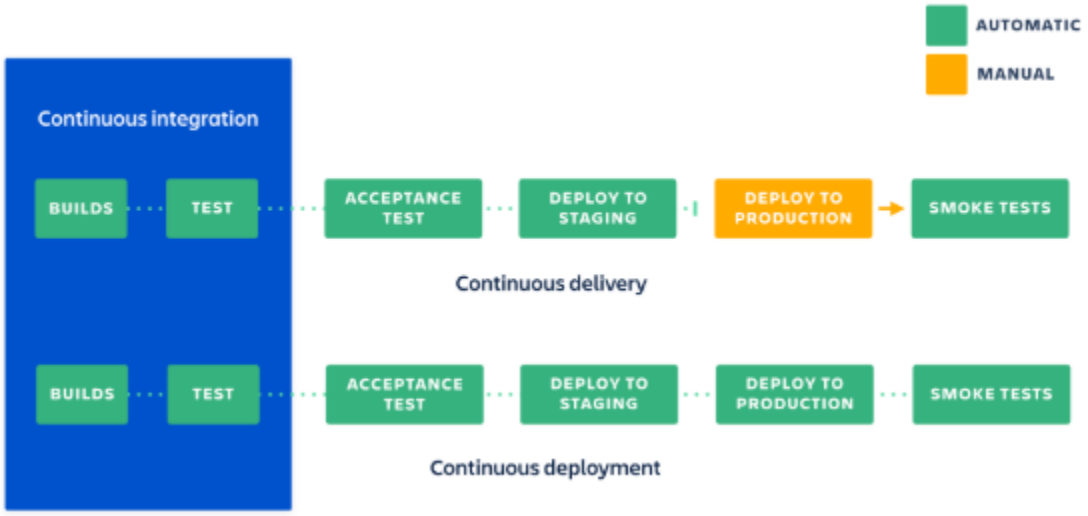
CD/CD

What is CI/CD?

- CI/CD is a method to frequently deliver apps to customers by introducing automation into the stages of app development. The main concepts attributed to CI/CD are continuous integration, continuous delivery, and continuous deployment.
- CI/CD is a solution to the problems integrating new code can cause for development and operations teams (AKA “integration hell”).
- Proponents of CI/CD claim benefits of accelerated time to market, faster customer feedback, improved developer productivity, reduced technical risk, improved quality, and improved customer satisfaction.
- Often, teams struggle to ship software into the customer’s hands due to lack of consistency and excessive manual labor. Continuous integration (CI) and continuous delivery (CD) deliver software to a production environment with speed, safety, and reliability.

Quick summary:-

- An assembly line in a factory produces consumer goods from raw materials in a fast, automated, reproducible manner. Similarly, a software delivery pipeline produces releases from source code in a fast, automated, and reproducible manner. The overall design for how this is done is called "continuous delivery."
- The process that kicks off the assembly line is referred to as "continuous integration."
- The process that ensures quality is called "continuous testing"
- The process that makes the product available to end users is called "continuous deployment."
- And the overall efficiency experts that make everything run smoothly and simply for everyone are known as "DevOps" practitioners.

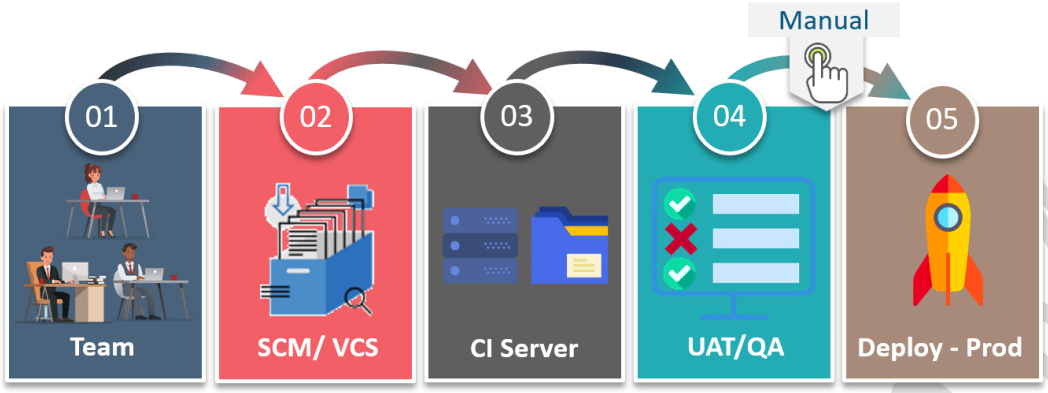


Continuous Delivery vs Continuous Deployment

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.

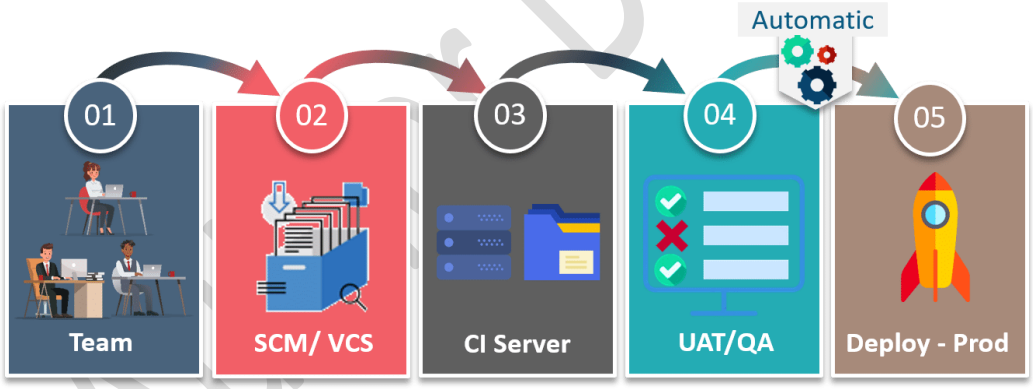


The benefit of continuous delivery lies in the fact that the code is ready to deploy at all the times. So, as you can see here the Quality Assurance team tests if each feature is working or not, and then they manually deploy it to production based on the need of business increasing the quality and velocity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



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 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. 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.