



# CI/CD

## What, Why & How

**Author : Rohit Sachdev**

# Our Software Delivery Process

## SDLC

Today our developers follow a Software Development Life Cycle process/model for developing Software Projects.

At the outset the process involves :

- Different Development teams with different code repositories.
- Build and Integration team integrates the code.
- Integrated code is compiled as a package.
- A package with instructions is sent to the Operations team.
- The Operations team pushes the package into a test environment, the package is tested by the Testing team.
- The Test team informs the Development team about any possible issues else gives a green signal to operations team for a Production Release.

## SDLC Processes





## CI/CD

### What is it?

While SDLC have been serving technology owners and makers since the beginning of it all, it is no more valuable in this fastly adapting world to practice SDLC Waterfall Model wherein we have to wait till the end to receive a functional product, generally delivered at the end of a cycle.

That's where CI/CD comes to the rescue. It is more of a culture and practice than tools, but what is CI/CD?

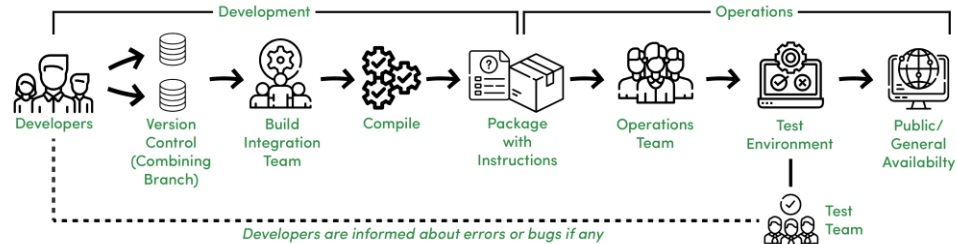
CI/CD stands for Continuous Integration and Continuous Delivery (or Continuous Deployment), it is about how an Integrated code on a shared repository is used to release software to production multiple times a day with the help of automation.

Continuous Integration refers to the practice of merging all developers' working copies to a shared mainline several times a day. Everything related to the code fits here, and it all culminates in the ultimate goal of CI: a high quality, deployable artifact

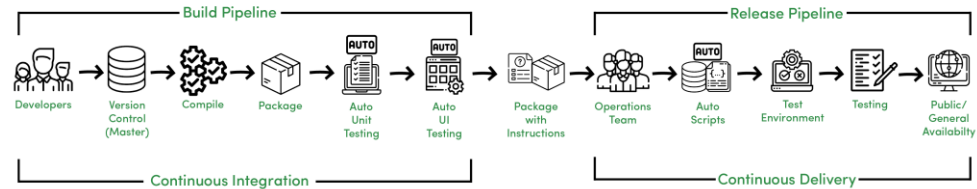
Continuous Deployment is a software engineering approach in which the value is delivered frequently through automated deployments. It's the process of deploying/delivering the artifact from the packaged builds to production.

# CI/CD Why Needed ?

## Traditional SDLC Approach



## CI/CD Approach



## CI/CD Why Needed ?

**Faster Software Builds** - Integrating CI and CD results in faster builds and is known to deliver quicker results. With deployments running in continuous cycles, it enables to track the project and provide feedback in real-time, as well as fix shortcomings whenever necessary. With consistent reviews, the product refinement strengthens and helps meet end-users' expectations.

**Time-to-Market** - By deploying apps to the market in time, we will not only engage with our customers better, but will also achieve assured profits, support pricing, and advance market goals. With significantly improved time-to-market it will for certain have a positive affect on our ROI.

**Improvements to Code Quality** - It helps deliver quality products. CI allows developers to integrate their code into a common repository. With the help of this repository, developers can share their builds multiple times a day rather than working in isolation. In turn, this will help reduce integration costs, as developers can share their builds in a more frequent manner.

**Efficient Development** - By incorporating CI, our development team becomes more bankable and the cost of our software projects is minimized. CI aims at refining the integration process, thus breaking it down into simple, daily development tasks, to help minimize the build costs and detect defects early in the software lifecycle.



## CI/CD Process

