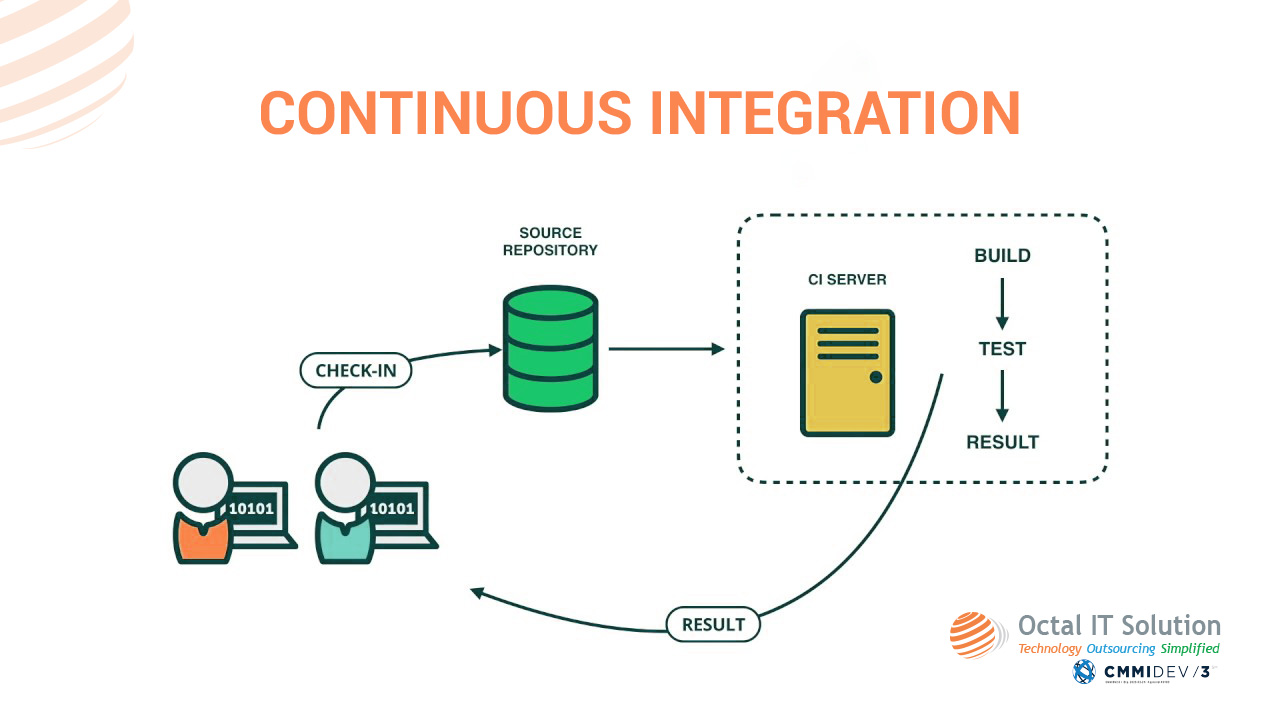
**Implementation of Efficient CI/CD Pipeline**

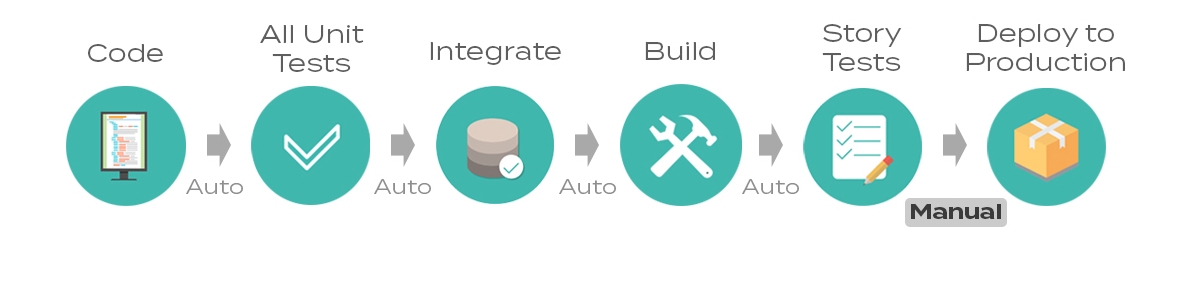
When you think of combining a set of operations in the building, testing, and deployment of applications, CI/CD is the ultimate technique. CI refers to the Continuous Integrations and CD refers to Continuous Delivery or Continuous Deployment. It enables a work culture where different sets of operations and multiple activities are coordinated in a way where code change becomes more smooth and flawless. To understand it more precisely, we have shared a detailed account of both activities.

**What is CI (Continuous Integration)?**



According to the concept, the developer’s working copies are shared with this devoted mainstream multiple times a day. The [dedicated developers](https://www.octalsoftware.com/hire-dedicated-resources) take note of the current code base to work upon. Meanwhile, the other developer submits the changed codes to the source code repository. The process facilitates existing code changes but allows to add of new codes, libraries, etc.

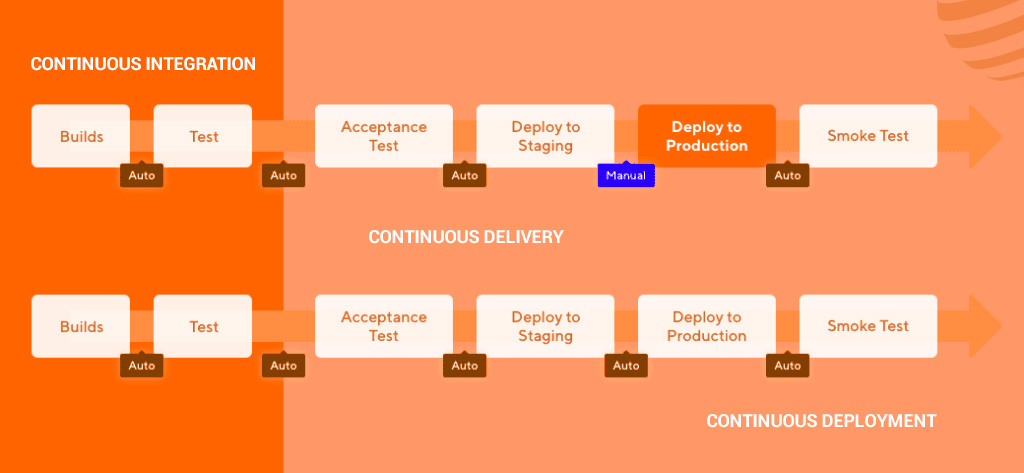
**What is CD (Continuous Delivery)?**



Now, this process involves producing software in a comparatively short period. The CD technique ensures the reliable release of the software at any point in time. The main game is to build, test, and release the software with high speed and frequency. Moreover, the benefits include just cutting down on Cost, Time, and Risk of Delivering Changes with incremental updates.

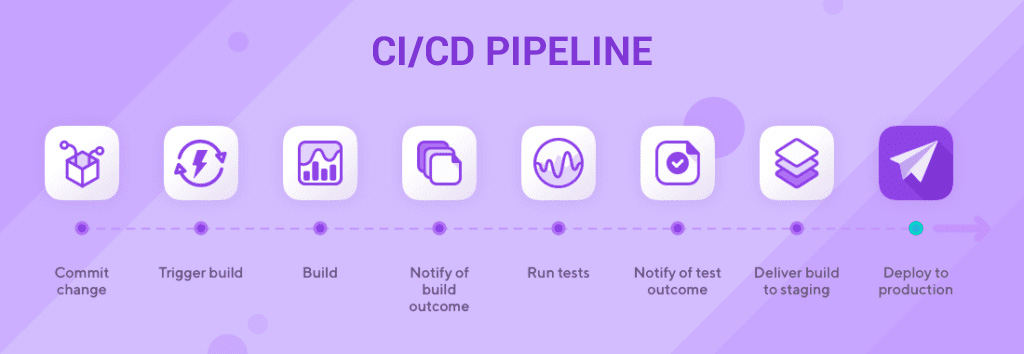
**Now, What is Continuous Deployment?**

This is a software engineering approach to formulate the automated deployment of functionalities in an application. It contrasts with Continuous Delivery as the software is also produced in a short cycle but involves automation not that manual operations like continuous delivery.



After gaining an understanding of both of the crucial terms, another important aspect associated with these terms is CI/CD Implementation. Now, the coming paragraphs will elaborate on the topic more broadly.

**What is the CI/CD Pipeline?**



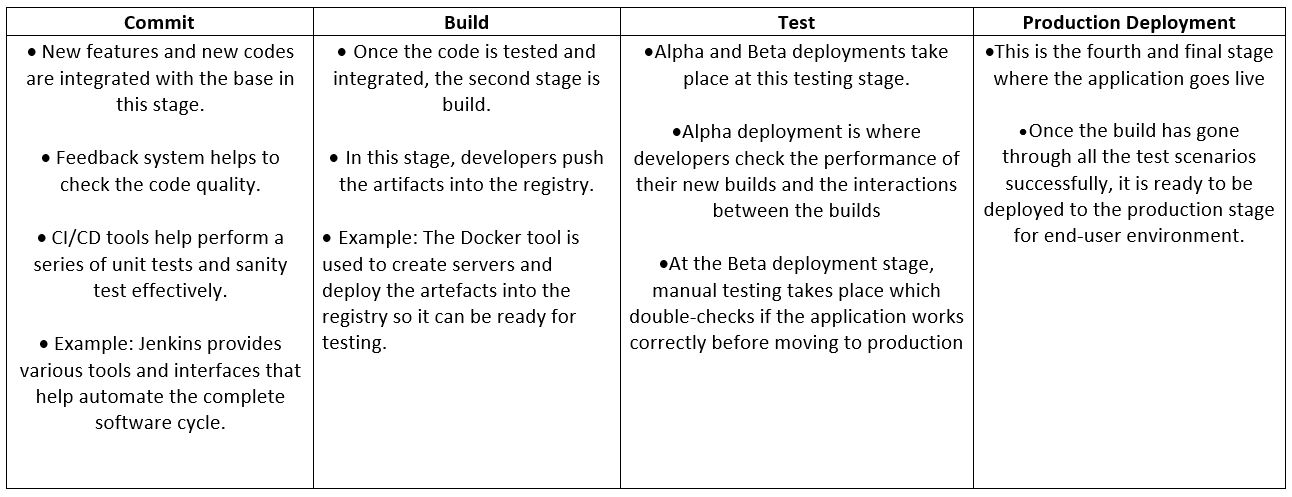
The systematic procedure that automates the [software development](https://www.octalsoftware.com/) process is termed as CI/CD Pipeline. The process accompanies building codes, running tests i.e. CI, and then safely deploying the latest version of the application i.e. CD. Generally, these pipelines abolish manual errors, signals feedback loops to the development team, and fasten the production.

**What Is the CI/CD Implementation Plan?**

The implementation of CI/CD depends upon a few stages. It can be done either with a [DevOps](https://www.octalsoftware.com/devops-solutions) or SRE (Site Reliability Engineering) approach. The approach is mainly utilized in the integration and testing phase as well as during the delivery and deployment phase.

### [**DevOps - How it works and What are the application areas?**](https://www.octalsoftware.com/blog/devops-software-development)

There are few elements that affect the CI/CD Implementations. These elements go hand in hand during the production of the latest version of an app and any failure in these steps will automatically trigger an error notification to the concerned team.



### **Source Stage**

As we all know, with the help of CI, a source code repository gets created. Any change in the code or establishing a new code is notified to the associated pipeline. In this way, any error happening like automated scheduling or user initiating any workflow, or results from other pipelines are notified.

### **Build Stage**

In this step, the application is compiled by putting together the codes and related dependencies. It helps in building a runnable entity of the product. Further, the runnable product or instance is shipped to the end-users. A small but extra effort ie.. compilation has to be taken when the instance is produced in Java, C/C++, or GO language. But when the product is developed in [Ruby](https://www.octalsoftware.com/ror-development-services), [Python](https://www.octalsoftware.com/hire-dedicated-python-developer), and [JavaScript programs](https://www.octalsoftware.com/java-script-development-company) then you need not follow such compilations.

### **Testing Stage**

The most crucial stage of all is the Testing phase. Automated testing is done to verify the ascertain the behavior of the product and exactness of the codes so implemented. It becomes a safety trap that stops reproducible bugs from reaching the end-user.

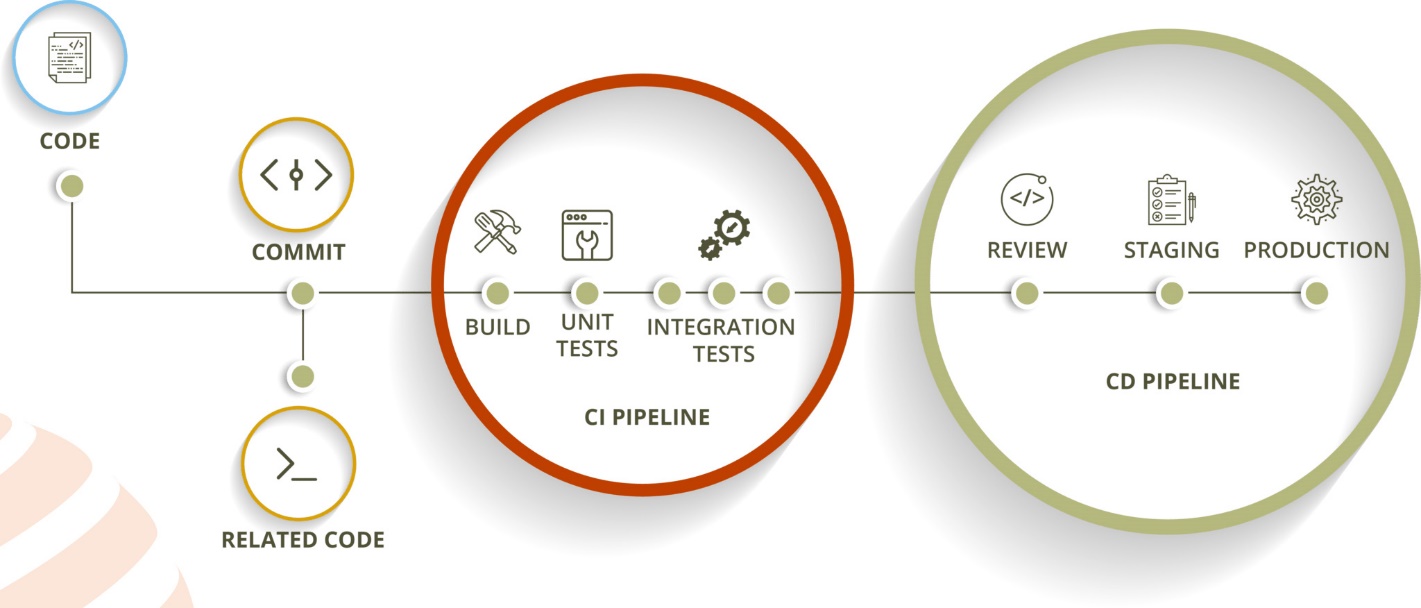
The duration of testing depends upon the type of project and the skills required to validate the product. It might take seconds to hours or days to complete the testing of the product. Failure in the testing exposes the problems with the codes. Thereafter the developers are responsible for re-writing or reversing the codes.

### **Deployment Stage**

After all the qualify and production checks, the instance then becomes ready to deploy on the product. Depending on the project type, there can be multiple layers in the deployment of the version. It can be a beta or staging version that runs internally by the product team. Once the instance is sorted, the codes get automatically deployed on the product. In this way, these elements work for the deployment of a new version of an application.

## What Makes A Good CI/CD Pipeline?

After implementing the pipeline, what features make it a sustainable pipeline? Continuous Integration and Continuous Delivery is purely based upon steps upon steps. Each step depends on another step and then it flows into depending loops.



The implementation of the pipeline becomes when your development satisfies the below-given aspects. Yes, you created a pipeline but how will you detect the performance? It becomes possible when you consider these aspects.

### **1. Speed**

**Quick Feedback:**It is always required to obtain quick feedback on the exactness of the codes or of the work done. Pushing these codes to CI is equivalent to inevitable task switching which always leads to developer work less efficiently. So, it is better to get feedback from the right person on the right task and yes at the right time.

**Duration of Setting Up the Pipeline:**The implementation of the pipeline requires quick execution and more accuracy. This in the meantime makes any developer take more risk in deployment which lacks the uniqueness, company today following.

**Meeting the Scalable Development Standards:**On a very common note, a certain number of CI/CD Pipelines can run at a given point in time. As a well-understood result, other activities become idle for that particular time. On the other hand, developers wait for their CI/CD to become available for them as then it comes in the busiest hours of the day. Hence, a new way should be adopted to establish equal work-flow between manpower and ascertain quick development standards.

### **2. Reliability**

Reliability is the key to fast and sustainable version deployments. Any pipeline gives assurity that the outcome will always be the input provided and without any other uncertainty. Mostly pipeline the same outcomes but when the project and work-flow grows, there might be chances that reliability of the codes or even of the instance becomes poor.

### **3. Accuracy**

Automation can be a good start but it is not enough to compose a functional CI/CD Pipeline. It ensures accurate running and when combined with other tools, must be capable of handling the complex and simple workflows. The process should be done in a way where it leaves no manual errors when performing repetitive tasks. Accurate pipeline results in complete automation where no human interference is involved.

### **4. Comprehensive Feedback**

CI/CD pipelines are introduced to cover as many as possible facets of a flawless software delivery system. Proper and comprehensive feedback allows developers to improve the development process and execute the required code for scalable application deployment.

## What Are the Advantages of CI/CD Pipeline Implementations?

Taking an account of the details discussed in the article, there are many benefits of executing the CI/CD Pipelines in the new version deployment. The benefits can be:

* Boost Speed & Quality
* Adds Technical Know-How
* Adds Business Values

**Boost Speed & Quality:**As we have an encounter earlier that accuracy in the pipeline always boosts the quality and speed of the production. It leads to automation which does not need any human involvement. Simultaneously, robust the entire CI/CD process.

**Adds Technical Know-How:**It brings on all the resources together and allows them to showcase their skills on simple as well as complex points. It allows any developer to stay focused on what they are known for i.e. publishing codes.

Again the quality of the codes is analyzed by the Testing unit. The team takes its proper time and evaluates all the codes implemented from scratch to the top. One small change or bug can lead to re-analyze the entire written code. In this way, the entire team either developers or testers bind themselves up to deliver the latest versions.

**Adds Business Values:**Speaking about business values then productivity is considered first. Quick and accurate deployment of the instances on the real products always welcomes the desired productivity. Similarly, this adds business ethics to the team too that any delay will automatically lose the uniqueness of the product and eventually the efforts.

## Conclusion:

To wrap up on CI/CD pipeline, it is always better to implement the pipeline with a full passionate team. Skilled developers, smart testers, and a focused working environment ensured quick results and super success of the product in the market. Hence, Continuous Integration and Continuous Delivery (CI/CD) will be the new normal that will automate the human efforts in development soon.