BENEFITS OF CONTINUOUS INTEGRATION AND CONTINUOUS DELIVERY/DEPLOYMENT

Continuous delivery is the culture that amalgam the practices of continuous integration and continuous deployment. Continuous Integration (CI) is the constant merging of code changes from various developers into a central branch as often as possible. On the other hand, Continuous Deployment (CD) is where all changes that have gone through the CI stage successfully are made available to customers automatically with no human interference. Continuous Delivery, when implemented, will enable developers to constantly have a working valuable artifact and enhance the consistent and frequent delivery of features to our cherished customers.

Continuous Delivery relies on some principles, namely, repeatable reliable processes – the ability of the process for releasing of software to be trustworthy and capable of producing the same results all the time, automation – for the process to be reliable will need it to be void of any human interference, version control of code – thus everything done with the code must be kept in version control like git and GitHub, bring the pain forward, built in quality - our processes will be designed to include quality checks, "Done" means released, shared responsibility – thus everyone is responsible for ensuring the release of the product and then continuous improvement – being intentional with small daily improvements.

Fundamentals of CI/CD

The basic building blocks of a CI/CD pipeline are Continuous Integration and Continuous Deployment.

Continuous Integration: This is known as the process of "making," everything relating to code is done here. It is the practice of the continuous merging of every developer's working code into a shared repository to achieve a high-quality working artifact. It comprises of the following stages:

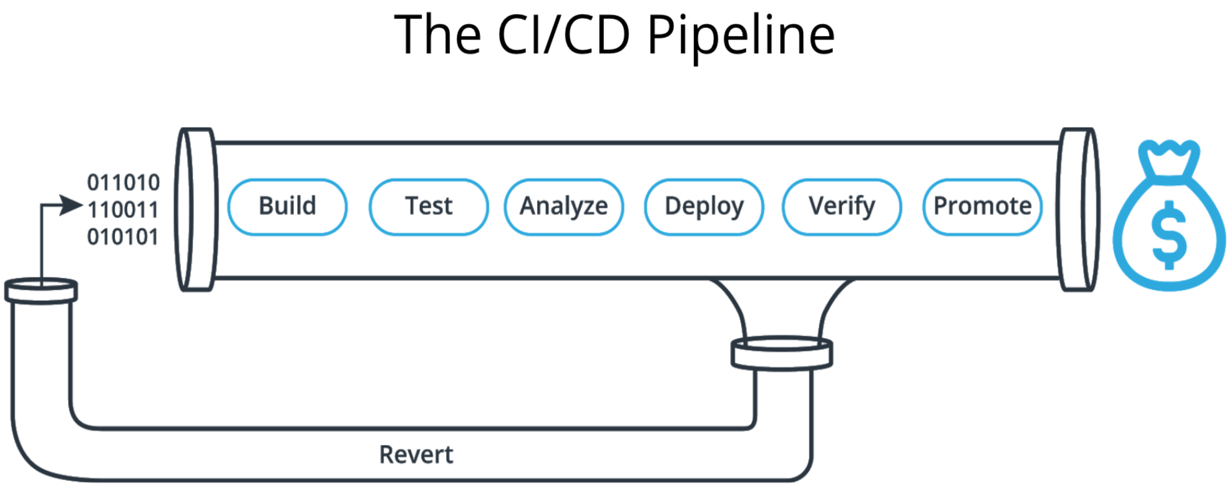
1. Compile – This is where the code is built to produce an artifact- a working piece that can be deployed.
2. Unit Test – This is where testing is run on the code to ensure the quality of the artifact produced.
3. Static Analysis – Further code analysis is being run at this stage to further ensure the quality of artifact being produced.
4. Dependence vulnerability testing and Storage of artifact.

Continuous Integration comes along with some benefits for the business. With proper Continuous Integration in place, less bugs are sent to production as they are caught early at the test stage and rectified resulting in revenue for the company as quality is improved. The cost of testing is totally reduced, which is also a gain for the company in terms of revenue.

Continuous Deployment: This is where the valuable working artifact is shipped to customers seamlessly and consistently through automated deployments. This is also known as the “moving” phase – moving the working artifact from the development house to the public. It comprises the following stages:

1. Creating and provisioning of infrastructure and servers – This is where we deploy the necessary infrastructure and servers needed to run the application using infrastructure as code.
2. Copying files – This is the copying of necessary files on to the provisioned servers.
3. Promotion to Production – This is where the working application is set rolling on the provisioned server.
4. Smoke Testing – This is where we confirm the efficiency of the servers and application.
5. Roll Back – This is where a roll back is performed should something go wrong in the deployment stage.

Continuous Deployment also comes along with its own business benefits. With proper Continuous Deployment in place, there is a continuous stream of improvements for customers leading to customer satisfaction and an increase in quality which increases revenue as a happy customer equals a successful business.



The phases of a CI/CD pipeline

In all a well-established CI/CD pipeline will add the following values to the business:

1. Help catch errors during compile time at merge thereby reducing developer time on issues from the new code which in turn reduces cost on the side of the business.
2. Detect security vulnerabilities there by preventing costly security holes hence avoiding cost for the business.
3. Faster and more frequent releases, new value features are enrolled quickly there by increasing revenue for the business.
4. Less time to market as deployments are automated and hence increase in revenue for the business.
5. Quicker and faster roll back to get production back into a working state, hence protecting revenue.