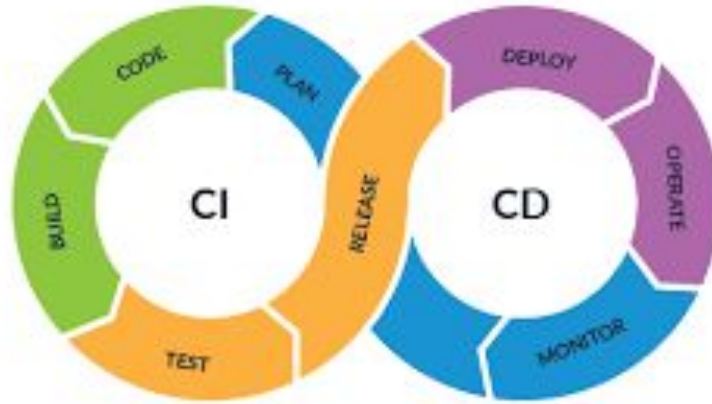# CI/CD Implmentation

Introduction

CI/CD is a method to frequently deliver apps to customers by introducing automation into the stages of app development.

Phases

1. The *Plan* phase often combines practices from Scrum and Agile to enable frequent microincremental releases.
2. The *Code* phase focuses on core development tasks from within IDEs and appropriate sandboxing and frameworks.
3. The *Build* phase rapidly and incrementally merges code commits with some testing and security validation.
4. The *Test* phase focuses on automated verification of enhancements, often incorporating test-driven deployment practices. Testing is sometimes incorporated as part of the Build phase, and generally extends in some way to all phases of the CI/CD process to ensure continuous feedback and improvement.
5. The *Release* phase is centered around repository commits and adequate documentation of the changes.
6. The *Deploy* phase is the actual update to the codebase, with special thought given to issue and error avoidance.
7. The *Operate* phase occurs once the code is made live, and consists of monitoring and orchestration.
8. The *Monitor & Optimize* phase takes place parallel to the Operate phase and consists of data collection, analysis, and feedback to the start of the pipeline and to other phases as needed.

Advantages

Save money

With CI/CD, you can save money because a lot of tasks that were once manual and time-consuming can now be automated. You consequently have to pay extra red dollars as a result of this (time spent by staff members releasing to production.

Save time

Because everything will be automated and you can view the build process in CI/CD, you will be able to save time by being informed as soon as failures occur. It has also pipeline reuse for each release, automatic rollbacks for when things go wrong, and automated infrastructure building and cleanup to reduce human mistake and keep things neat and orderly while saving you money.

Constant releases

Minor features are regularly released to consumers with every commit to GitHub, there is no need to wait for a single large release of a feature.