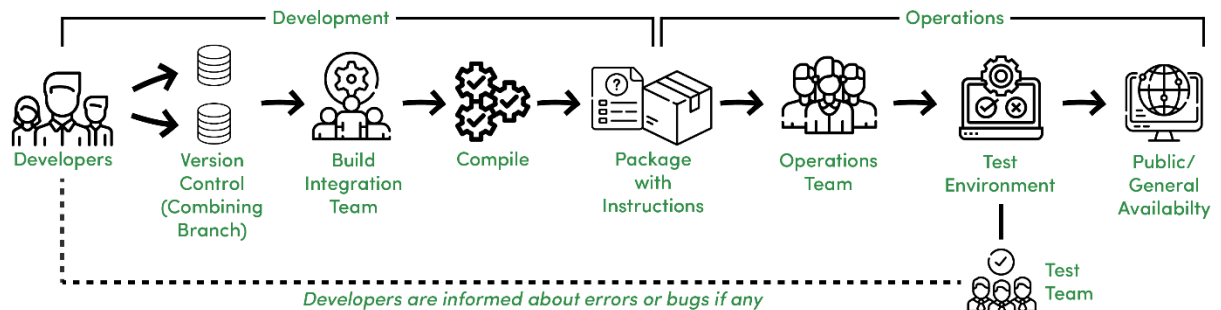


CI/CD

Continuous Integration and Continuous Delivery (or Continuous Deployment)

1. **Planning** –Documenting and determining project structure and schedule.
2. **Analysis** –Analyzing project requirements and collecting resources.
3. **Design** –Designing software model with architecture and Interface designing.
4. **Implementation** –Developing an actual product by considering design and requirements.
5. **Testing** –Testing the built, fixing errors & bugs, and refactoring the code.
6. **Deploying and Maintenance** –The software is deployed and monitored for further enhancements.



Features –

An overview of a Traditional [SDLC](#) shown in the above image works in the following manner.

- Different Development teams maintain 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 for Production Release.

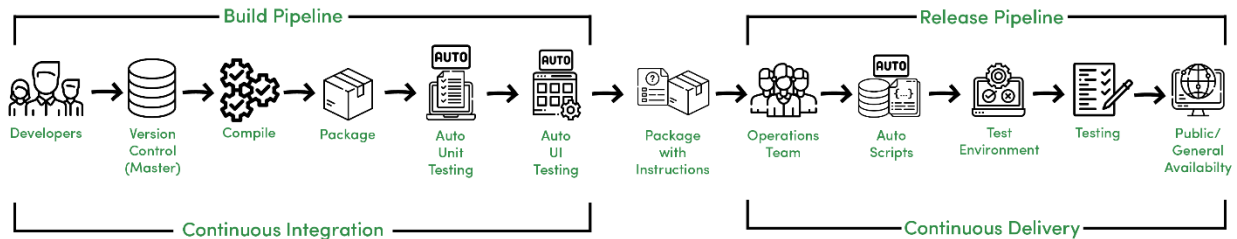
However, it is not valuable in this fastly adapting world to practice the above process and models like the Waterfall Model wherein we have to wait till the end to receive a functional product, generally delivered at the end of a cycle.

Continuous Integration :

Continuous Integration is about how developers integrate code using a shared repository multiple times a day with the help of automation.

Continuous Delivery :

Continuous Delivery is about automatically releasing software to the test or production environment.



Features –

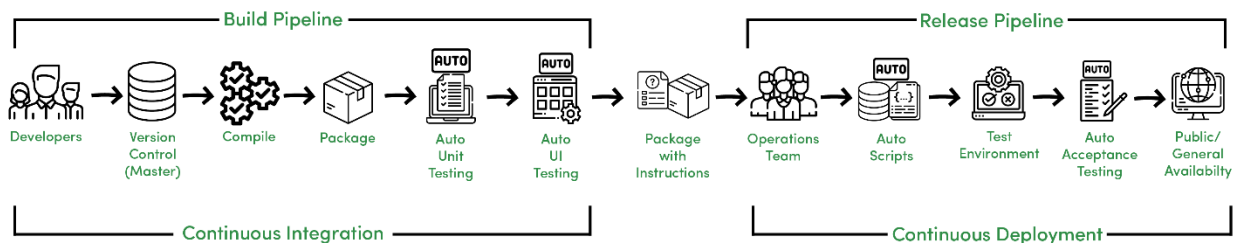
The above image shows Continuous Integration and Continuous Delivery process that differ from traditional SDLC due to the following reasons,

- Developers manage code on a shared repository.
- After compilation, Automated Unit and UI testing are performed.
- The Operations Team needs to take care of Automated scripts that further go to the test environment.
- Testing is performed and after approval, the software is sent to production.

Here Continuous Integration can be referred to as Build Pipeline and Continuous Delivery can be referred to as Release Pipeline.

Continuous Deployment :

Continuous Deployment is about releasing software to production automatically without human intervention.



Continuous Deployment is an advancement and it reduces manual work of the Operations and Testing team to deploy the software to production as it is automatically deployed to production after Automated Acceptance Testing.

Features –

CI/CD helps for rapid and reliable software development in the following manner.

- Automated and quick code integration.
- Error detection and response are fast due to the short cycle and iterations.
- Less prone to human errors.
- Faster than following manual instructions due to automated scripting.
- Deployment can be done several times a day.

Moreover, CI/CD in [DevOps](#) is about unifying process and automation, it helps to act as a bridge between Development and Operations team to make the process more reliable and adaptive.