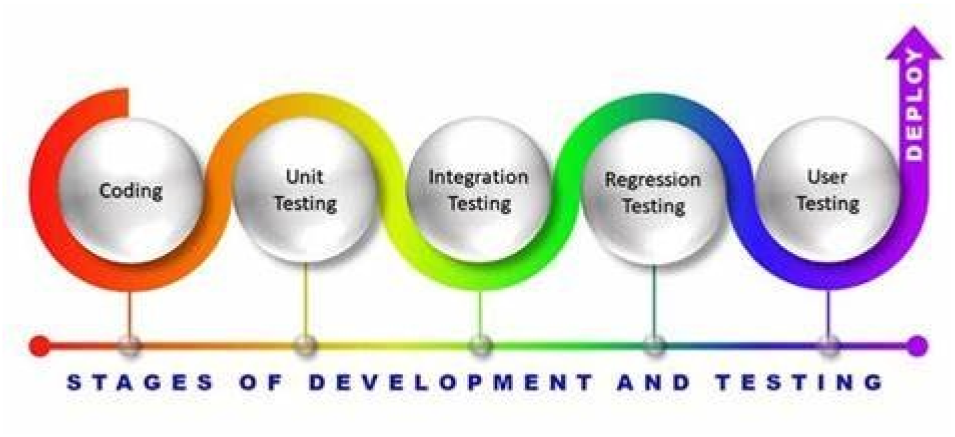
UDAPEOPLE CI/CD



A presentation by Dalia Abdulmonsif on how CI/CD can help us make more money and save cost.

What is CI/CD?

CI and CD stand for continuous integration and continuous delivery/continuous deployment.

In very simple terms, CI is a modern software development practice in which incremental code changes are made frequently and reliably. Automated build-and-test steps triggered by CI ensure that code changes being merged into the repository are reliable.

The code is then delivered quickly and seamlessly as a part of the CD process. In the software world, the CI/CD pipeline refers to the automation that enables incremental code changes from developers’ desktops to be delivered quickly and reliably to production.

What is the difference between CI and CD?

Continuous integration (CI) is practice that involves developers making small changes and checks to their code. Due to the scale of requirements and the number of steps involved, this process is automated to ensure that teams can build, test, and package their applications in a reliable and repeatable way. CI helps streamline code changes, thereby increasing time for developers to make changes and contribute to improved software.

Continuous delivery (CD) is the automated delivery of completed code to environments like testing and development. CD provides an automated and consistent way for code to be delivered to these environments.

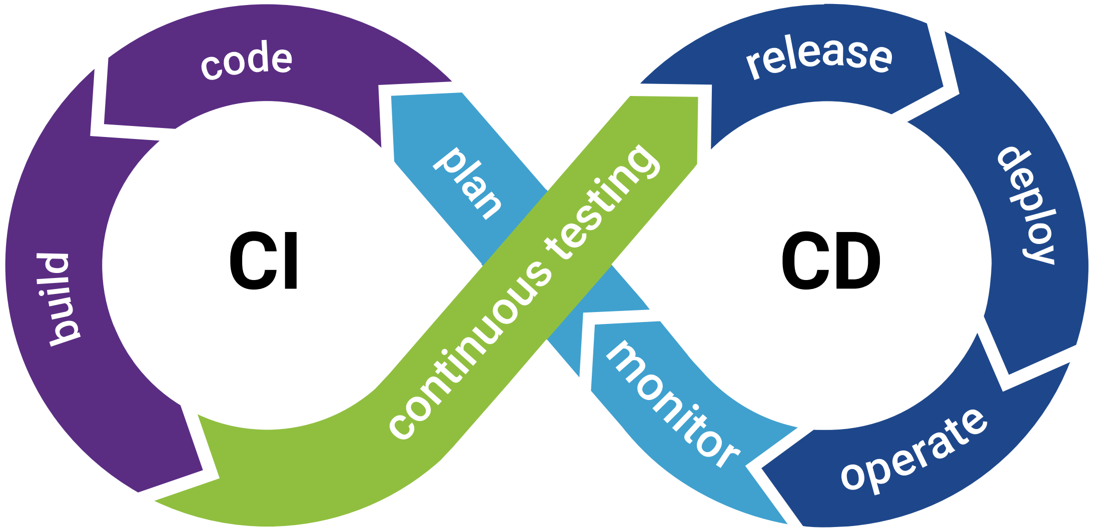
Continuous deployment is the next step of continuous delivery. Every change that passes the automated tests is automatically placed in production, resulting in many production deployments.

Continuous deployment should be the goal of most companies that are not constrained by regulatory or other requirements.

In short, CI is a set of practices performed as developers are writing code, and CD is a set of practices

performed after the code is completed.

What is the difference between CI and CD?



How will this help us save cost ?

- Software developers spend a lot of time **debugging** faulty code and performing manual tests to find bugs. Even after this, this is a very high chance that errors will slip into the production environment. Errors in the production environment also mean that the customer care team of Udapeople will have to do a lot of work appeasing angry

  With a correctly set up CI/CD process, only code that has passed the tests will make it to production. Also, there will be little need for the customer service team to be engaged with appeasing customers which will save cost on the number of staff and resources needed.
- Reducing the number of errors encountered can also mean that developers now have more time to work on the actual project, which saves costs on hiring more people to work on the same project that was originally taking so many resources because of debugging.
- Downtimes are periods where our services are not usable by the customers. When customers experience lots of downtimes in our services, the next action is to switch to our competitors. This in turn can cost a ton of marketing funds to get more customers or get back the ones who have lost. CI/CD processes will greatly **reduce the chances of having downtimes.**

How this will help us make more money ?

- With time taken from fixing bugs, developers can now focus on more features which will make the customers happier and keep them patronizing our services and products.
- When software developers are not tied down by fixing bugs, they become more productive. This improves their motivation and helps them become more creative. A creative software development team makes for more robust software which means more money.
- Having excellent services means more satisfied customers which means more referrals and more money. The CI/CD process ensures the quality of code which means great service and happy customers.
- The ability to release new features rapidly which is what CI/CD enables means that we can get customer feedback early and implement new features that satisfy their needs.