



Problem Definition:

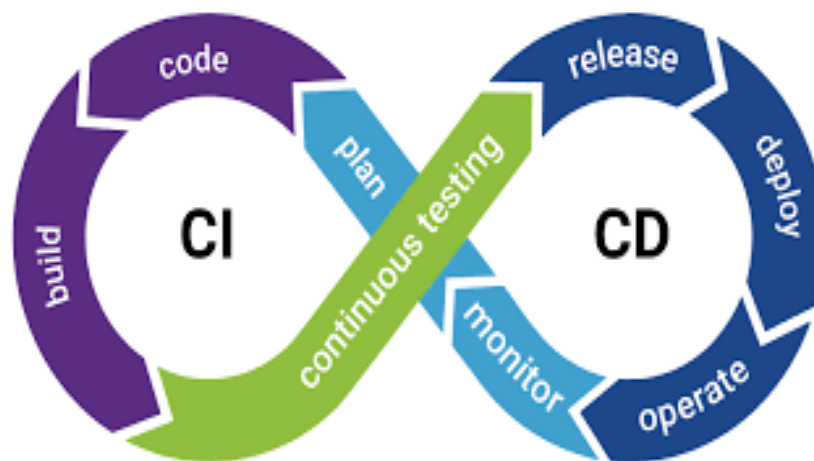


Using a manual way to produce our products faces many difficulties which are

1. our manual way as usual has errors that cause delays in production deployments which cause a high-cost process and Customer dissatisfaction
2. there is not enough time for quality analysis
3. complexity of the deployment process which needs many people to do, resulting in a high-cost process

4. complexity of the deployment process which depends on few people
5. there is no enough time for smoke tests that may result in problems while delivering our product and embarrassing situations with customers
6. poor rollback mechanism causing a delay in delivering our product

what does CI/CD stand for? and why it is the key solution for the previous problems?



What does CI/CD stand for:

- CI/CD is a **method** to frequently deliver apps to customers by introducing automation into the stages of app development. The main concepts attributed to CI/CD are **continuous integration**, **continuous delivery**, and **continuous deployment**. CI/CD is a solution to the problems integrating new code can cause for development and operations teams.

Continuous integration:

Continuous integration (CI) helps developers merge their code changes back to a shared branch, or “trunk,” more frequently—sometimes even daily. Once a developer’s changes to an application are merged, those changes are validated by automatically building the application and running different levels of automated testing

Continuous delivery

Continuous delivery automates the release of that validated code to a repository. So, to have an effective continuous delivery process, CI must be already built into your development pipeline. The goal of continuous delivery is to have a codebase that is always ready for deployment in a production environment.

Continuous deployment

Continuous deployment means that a developer’s change to a cloud application could go live within minutes of writing it (assuming it passes automated testing). This makes it much easier to continuously receive and incorporate user feedback

Why **CI/CD** is the key solution for the previous problems?



- *Change fail rate KPI or metric*

It helps you reduce errors on production that helps to solve 1,2,5 problems

- *Increases web development productivity*

One of the CI CD benefits is the increase of productivity. If you have a review process like deploying code to dev, testing, production, entering several commands across different environments

- *Helps you automate your testing phase*

Using or leveraging CI-CD principles. Well, you can automate your testing, such as unit testing, regression testing, exploratory testing, or even your security testing.

- *Customer Satisfaction*

Keep your customers happy with fast turnaround of new features and bug fixes. Utilizing a CI/CD approach also keeps your product up-to-date with the latest technology and allows you to gain new customers who will select you over the competition through word-of-mouth and positive reviews.

- *Reduce Costs*

Automation in the CI/CD pipeline reduces the number of errors that can take place in the many repetitive steps of CI and CD. Doing so also frees up developer time that could be spent on product development as there aren't as many code changes to fix down the road if the error is caught quickly. Another thing to keep in mind: increasing code quality with automation also increases your ROI.