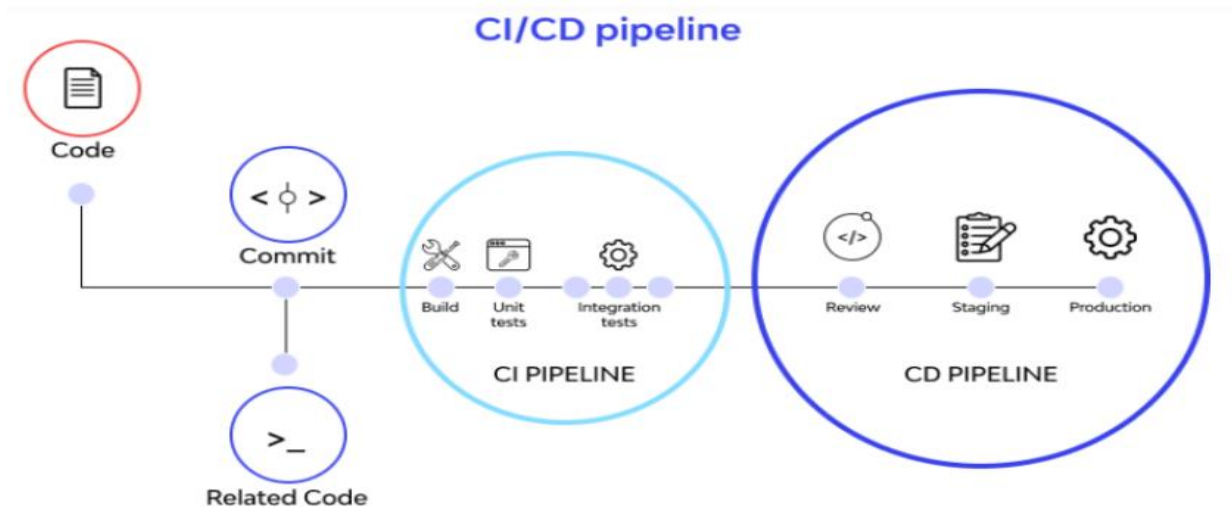


## What is CI/CD?



**CI/CD** A continuous integration and continuous deployment ([CI/CD](#)) is a series of steps that must be performed in order to deliver a new version of software. CI/CD pipelines are a practice focused on improving software delivery throughout the software development life cycle via automation.

By automating CI/CD throughout development, testing, production, and monitoring phases of the software development lifecycle, organizations are able to develop higher quality code, faster. Although it's possible to manually execute each of the steps of a CI/CD pipeline, the true value of CI/CD pipelines is realized through automation

### Continuous Integration (CI)

**CI** is a software development practice where development teams frequently commit application code changes into a shared repository. Typically, this would happen at least once or then several times a day (depending on the number of code commits) and this practice encourages committing small changes more often over committing large changes infrequently. These commit/changes automatically trigger new builds which are then validated by automated testing to ensure that they do not break any functionality.

### Continuous Delivery (CD)

**CD** is an extension of that process. CD is all about the ability to continuously deliver integrated code, be it bug fixes or new features, to production. It's the automation of the release process so that new code is deployed to target environments - typically to test or staging environments - in a repeatable and automated fashion.

It means that your “green builds” are ready to go in one click, should you wish to release them.

**Continuous Deployment** focuses on the automation process to release what is now a fully functional build into production. That means Continuous Deployment goes one step further as it allows you to automatically deploy live every main branch change that passes the CI. However, you might have business reasons for not doing so automatically or you might need to do testing that can't be automated.

## What is a CI/CD pipeline?

A pipeline is a process that drives software development through a path of building, testing, and deploying code, also known as CI/CD. By automating the process, the objective is to minimize human error and maintain a consistent process for how software is released. Tools that are included in the pipeline could include compiling code, unit tests, code analysis, security, and binaries creation. For containerized environments, this pipeline would also include packaging the code into a container image to be deployed across a hybrid cloud.

CI/CD is the backbone of a DevOps methodology, bringing developers and IT operations teams together to deploy software. As custom applications become key to how companies differentiate, the rate at which code can be released has become a competitive differentiator.

## What are the Business Benefits of CI/CD?

So, what are the business benefits of continuous integration and continuous delivery? In these next few sections, we'll take a look at CI/CD's business value from a few different angles.

- **Bring Products to Market Faster**

Organizations that have effectively implemented CI/CD can bring new products and features to market faster and immediately start generating revenue from the features they deploy rather than waiting for the entire app to be completed (and checked manually) before they can launch.

- **Allows Developers to Deliver Products Consumers Want Now.**

Over the last couple of years, customer-centricity has been a core focus for businesses.

- **CI/CD enables organizations to respond to consumer needs as they evolve.**

Teams have the flexibility to update applications and build and deploy new ones in response to emerging trends, new markets, and evolving expectations. According to Rodolfo Carmona, "good CI/CD implementation starts early in the production process by making the core functionality available to end-users right away. That way, early feedback and usability issues can be addressed without the need for major time-consuming refactors to change the direction later on."

- **CI/CD plays a crucial role in shortening time to value.**

According to 3Pillar Software Engineer, Paul Estrada, the strategy “fosters a culture of innovation, allowing developers to experiment with new technologies and try out new ideas. Teams can test different features with real users in parallel and use their findings to ensure.

- **Boosts DevOps efficiency**

Without CI/CD, developers and engineering teams deal with more pressure in their day-to-day—service interruptions, outages, and bad deploys can put their jobs at risk.

- **CI/CD Improves App Quality**

One puts it, “quality shouldn’t be tied to timelines. If you sacrifice quality in the CI/CD automation allows developers to deploy code more frequently and make incremental improvements to the code. Teams can automate regression testing and parallel tests, which improves test coverage and ensures that the code is bug-free and works across multiple environments.

What’s more, it’s much easier to roll back to the previous version without causing damage to other features and components in the case of failure.

- **Reduce Costs and Boost Profits**

CI/CD is also good for the bottom line. It standardizes deployment processes across all projects, and, done right, it enables teams to systematically test every change made to the source code.

As a result, this process stands to dramatically reduce the likelihood that any bugs or errors slip through the cracks and cause problems down the line. Done right, this practice can lower development costs by eliminating many of the costs incurred while building and testing code changes.

**In a nutshell, by adopting the CI/CD approach, your business stands to gain the following:**

- Detect issues in software builds faster and, therefore, facilitating quicker resolutions
- End-user involvement and feedback during CD can lead to usability improvements
- Deliver software on time with faster time to market
- Showcase your capability in the market, as you adopt (CI/CD) for producing quality mobile apps
- Forgo the stale periods in the lives of products
- Back-to-back releases using CD are easy and less time-consuming
- Share visibility of the development process in real-time

## **Conclusion**

CI/CD are two DevOps best practices as they tackle the misalignment between developers and the operational team. With the presence of automation, developers can release changes and new features more frequently, while operation teams have better overall stability.

Therefore, CI/CD is integral in software building and deployment: smaller teams, constant changes, fast and real-time feedback, and app deployment. Not only do they provide clear benefits to businesses but also to stakeholders such as product owners, development teams, and end-users, just to name a few.

**Thanks ,  
Ali Ahmed**