What is CI/CD and CD

Continuous Integration:

Continuous Integration is a coding philosophy and set of practices that drive development teams to implement small changes and check in code to version control repositories frequently. Because most modern applications require developing code in different platforms and tools, the team needs a mechanism to integrate and validate its changes.With CI it is easier to identify defects and other software quality issues on smaller code differentials rather than larger ones developed over extensive period of times.

Continuous Delivery:

Continuous delivery is an extension of continuous integration since it automatically deploys all code changes to a testing and/or production environment after the build stage.
This means that on top of automated testing, you have an automated release process and you can deploy your application any time by clicking a button.

Continuous Deployment:

Continuous Deployment goes one step further than continuous delivery. With this practice, every change that passes all stages of your production pipeline is released to your customers. There's no human intervention, and only a failed test will prevent a new change to be deployed to production.

Business Benefits

- ## Improved code quality

One of the chief technical benefits of CI/CD is that it enhances the overall code quality. A CI/CD pipeline enables developers to integrate their code into a common repository in small batches. Through this repository, developers can share their builds with the entire team rather than working in isolation. Now, the whole team can collaborate for thorough detection and fixation of the most severe bugs. Additionally, this helps to share stable builds more frequently, free of any critical bug, and thus bad code rarely makes it to production. That's how CI/CD implementation simply diminishes the off-chance of buggy codes making it too far into production.

- ## Time to review

Code changes in such an environment are done at an atomic level, which reduces the risk of unintended consequences. Such changes are easy to handle compared to large hefty ones and are easy to fix in case of any issues. Using CI/CD, these code changes can be tested quickly once they are integrated. Such an approach is very beneficial when direct communication is a roadblock or when teams are spread across remote locations.

- ## Fast release cycles

CI/CD serves this purpose by continuously merging codes and deploying it to production-like systems regularly to keep the code in a release-ready state. This also enables the organization to establish a standardized delivery mechanism that runs repeated processes for each change, trusted by all.

- ## Enhanced Test Reliability

Using CI/CD, you can improve test reliability to a great extent. Since specific and atomic changes are introduced to the system, it allows developers or QAs to add more relevant positive and negative tests for the changes. This testing is also referred to as 'Continuous Reliability' within a CI/CD pipeline.

- ## Reduced Backlog

By implementing CI/CD into your development process, you have the chance to decrease the number of non-critical defects in your team's backlog. Such defects are often fixed before they become a critical issue. Any such defect is highlighted and is fixed before it makes it to production or impacts end-users.

- ## Cost Deduction

Using a CI/CD pipeline limits the potential impact and loss that a deployment problem can cause by allowing it to be deployed in non-critical business hours. Also, repeated automated deployments during the development phase help developers catch the errors early before causing any significant damage. Such a pipeline implementation increases code quality, thus increasing overall ROI for the organization.

- ## Enhanced Transparency & Accountability

The benefits of CI/CD are not just limited to testing, but it can also help you assist your team in a better way. CI/CD encourages transparency and accountability for any problem amongst the team members. Continuous integration focuses on providing this benefit, as the reports generated there list down all the build failures, code integration problems, architectural bottlenecks, etc. A steady flow of these reports allows the team to analyze and fix problems to make the product even more useful.

- ## Communication

For each pipeline run on a CI/CD environment, all the stakeholders are notified and kept on the same page for any changes being made and in case of any failures. This allows the product owners and developers to communicate effectively about the test results and take desired actions based on the failure's criticality.

- ## Frequent Updates & Maintenance

Regular maintenance and updates are the backbones of making a great product, and this is one of the great benefits of CI/CD. It ensures that release cycles are shorter and targeted, which blocks fewer features that aren't ready for release. In a CI/CD pipeline, maintenance is usually done during non-business hours, saving precious time for the entire team.

- ## Customer Satisfaction

CI/CD also provides technical advantages and incorporates the ones that are essential from a business's viewpoint. Customer-first is the foolproof business approach, and a CI/CD pipeline follows this approach. Once a product is released, it monitors the initial few actions of any new customer and keeps track of the results. This helps decide the impression of the product on the user. CI/CD implementation also enables end-user involvement and feedback during continuous development leading up to usability improvements.

- ## Performance improvement

Applications, in all cases, need to be monitored carefully after releases. This can be considered amongst one of the top benefits of the CI/CD pipeline as it allows you to monitor the health, performance, and reliability of an app. Monitoring these metrics help in creating actionable insight and taking necessary actions to improve the product.