Udapeople

What is CI/CD?

Brief:

- Continuous integration (CI) and continuous delivery (CD), also known as CI/CD, embodies a culture, operating principles, and a set of practices that application development teams use to deliver code changes more frequently and reliably.
- CI/CD is a best practice for devops teams. It is also a best practice in agile methodology. By automating integration and delivery, CI/CD lets software development teams focus on meeting business requirements while ensuring code quality and software security.

Continuous integration

- is a coding philosophy and set of practices that drive development teams to frequently implement small code changes and check them in to a version control repository. Most modern applications require developing code using a variety of platforms and tools, so teams need a consistent mechanism to integrate and validate changes. Continuous integration establishes an automated way to build, package, and test their applications. Having a consistent integration process encourages developers to commit code changes more frequently, which leads to better collaboration and code quality.

Continuous *delivery*

- *Continuous delivery* picks up where continuous integration ends, and automates application delivery to selected environments, including production, development, and testing environments. Continuous delivery is an automated way to push code changes to these environments.

Automating the CI/CD pipeline

- CI/CD tools help store the environment-specific parameters that must be packaged with each delivery. CI/CD automation then makes any necessary service calls to web servers, databases, and other services that need restarting. It can also execute other procedures following deployment.
- Because the objective is to deliver quality code and applications, CI/CD also requires continuous testing. In continuous testing, a set of automated regression, performance, and other tests are executed in the CI/CD pipeline.

Cont.

- A mature devops team with a robust CI/CD pipeline can also implement *continuous deployment*, where application changes run through the CI/CD pipeline and passing builds are deployed directly to the production environment. Some teams practicing continuous deployment elect to deploy daily or even hourly to production, though continuous deployment isn’t optimal for every business application.

Benefits of Implementing a CI/CD Pipeline

- **1. Smaller Code Changes**
- One technical advantage of continuous integration and continuous delivery is that it allows you to integrate small pieces of code at one time. These code changes are simpler and easier to handle than huge chunks of code and as such, have fewer issues that may need to be repaired at a later date.
- Using continuous testing, these small pieces can be tested as soon as they are integrated into the code repository, allowing developers to recognize a problem before too much work is completed afterward. This works really well for large development teams who work remotely as well as those in-house as communication between team members can be challenging.
- **2. Fault Isolations**
- Fault isolation refers to the practice of designing systems such that when an error occurs, the negative outcomes are limited in scope. Limiting the scope of problems reduces the potential for damage and makes systems easier to maintain.
- Designing your system with CI/CD ensures that fault isolations are faster to detect and easier to implement. Fault isolations combine monitoring the system, identifying when the fault occurred, and triggering its location. Thus, the consequences of bugs appearing in the application are limited in scope. Sudden breakdowns and other critical issues can be prevented from occurring with the ability to isolate the problem before it can cause damage to the entire system.

Cont.

- **3. 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.
- **4. Easy Maintenance and Updates**
- Maintenance and updates are a crucial part of making a great product. However, it’s important to note within a CI/CD process to perform maintenance during downtime periods, also known as the non-critical hour. Don’t take the system down during peak traffic times to update code changes.

Cont.

- **5. Customer Satisfaction**
- The advantages of CI/CD do not only fall into the technical aspect but also in an organization scope. The first few moments of a new customer trying out your product is a make-or-break-it moment.
- **6. Increase Team Transparency and Accountability**
- CI/CD is a great way to get continuous feedback not only from your customers but also from your own team. This increases the transparency of any problems in the team and encourages responsible accountability.

Cont.

- **7. Faster Release Rate**
- Failures are detected faster and as such, can be repaired faster, leading to increasing release rates. However, frequent releases are possible only if the code is developed in a continuously moving system.
- **8. Smaller Backlog**
- Incorporating CI/CD into your organization’s development process reduces the number of non-critical defects in your backlog. These small defects are detected prior to production and fixed before being released to end-users.

Cont.

- **9. Faster Mean Time To Resolution (MTTR)**
- MTTR measures the maintainability of repairable features and sets the average time to repair a broken feature. Basically, it helps you track the amount of time spent to recover from a failure.
- **10. More Test Reliability**
- Using CI/CD, test reliability improves due to the bite-size and specific changes introduced to the system, allowing for more accurate positive and negative tests to be conducted. Test reliability within CI/CD can also be considered Continuous Reliability. With the continuous merging and releasing of new products and features, knowing that quality was top of mind throughout the entire process assures stakeholders their investment is worthwhile.