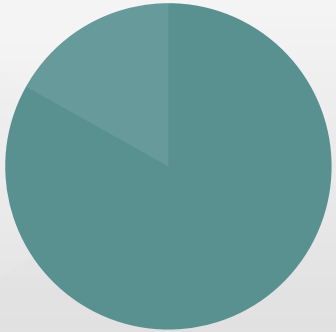
What is CI/CD? Continuous Integration & Continuous Delivery



Continuous Integration

The practice of merging all developers' working copies to a shared mainline several times a day. It's the process of "**Making**". Everything related to the code fits here, and it all culminates in the ultimate goal of CI: a high quality, deployable artifact! Some common CI-related phases might include:

- Compile
- Unit Test
- Static Analysis
- Dependency vulnerability testing
- Store artifact

Continuous Deployment

A software engineering approach in which the value is delivered frequently through automated deployments. Everything related to deploying the artifact fits here. It's the process of "**Moving**" the artifact from the shelf to the spotlight. Some common CD-related phases might include:

- Creating infrastructure
- Provisioning servers
- Copying files
- Promoting to production
- Smoke Testing
- Rollbacks

Benefits of CI/CD and Reduce Costs

1. Smaller Code Changes

One technical advantage of continuous integration and continuous delivery is that it allows us 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, allowing our developers to recognize a problem before too much work is completed afterward.

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.

3. 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

Benefits of CI/CD and Reduce Costs

4. 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.

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 our product is a make-or-break-it moment.
Utilizing a CI/CD approach also keeps our product up-to-date with the latest technology and allows us to gain new customers who will select us over the competition through word-of-mouth and positive reviews.

9. 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.