

# CI/CD- BENEFITS OF AUTOMATION FOR HIGHLY EFFECTIVE SERVICES

# CONTINUOUS INTEGRATION (CI)

- The practice of automating the integration of code changes from multiple contributors into a single software project.
- The developer's changes are validated by creating a build and running automated test against the build.
- CI puts a great emphasis on testing automation to check that the application is not broken whenever new commits are integrated into the main branch.

# 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.
- Continuous deployment is an excellent way to accelerate the feedback loop with your customers and take pressure off the team as there isn't a "release day" anymore.

# 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.
- In theory, you can decide to release daily, weekly, fortnightly, or whatever suits your business requirements. However, if you truly want to get the benefits of continuous delivery, you should deploy to production as early as possible to make sure that you release small batches that are easy to troubleshoot in case of a problem.

## BENEFITS OF CI/CD

- The deployment work can be pack into 1 action can be trigger by click on 1 button from anyone who has permission, so it will reduce cost for technical employee, reduce the mistake when execute deploy.
- With CD, your changes will reach to user faster and more often, so it will increase user experience and increase income.
- With shorter time to wait the changes, the dependent works could be fast feedback and shorter time-to-market.
- With the right CI/CD pipeline in place, organization scan automate repetitive and mundane tasks that can cause employee burnout. This can lead to happier and more productive employees, resulting in higher quality code and, ultimately, increased revenue.