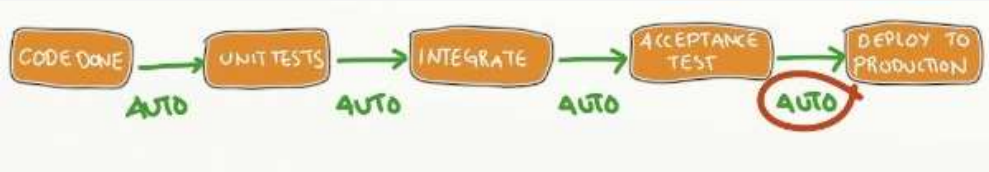
Proposal for the adoption of Continuous Integration/ Continuous Deployment

What is CI, what is CD?

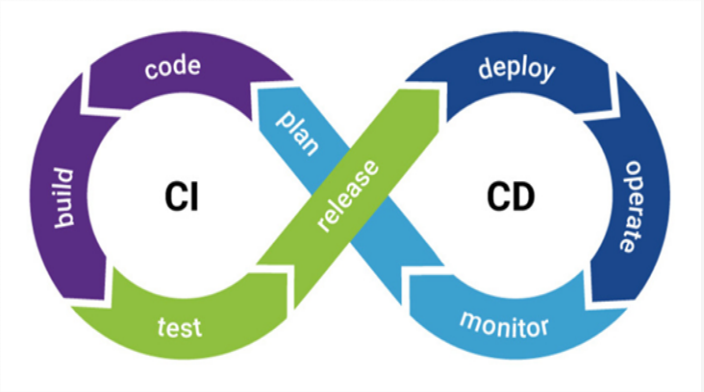
CI, or Continuous Integration, simply means the process of streamlining incremental code changes/commits from our different developers and automating it in such a way that eliminates/reduces merge conflicts through automated tests that check for bugs and errors without any need for human interaction.



CD, or Continuous Deployment, automates the deployment of production-ready code (code that has passed all tests) directly into production with the most minimum amount of downtime.

Adopting a Continuous Integration/Continuous Deployment Approach

Adopting Continuous Integration and Continuous Deployment into our operations will introduce some level of automation into how our software is delivered right from our code repository onto our production environment and made available to our end users.



A CI/CD approach will optimize the way we ship code from our staging environment to production, reduce human errors in the deployment process, and ensure we get our codes in the hands of the end-users faster; thus reducing our time-to-market.

Why CI/CD?

Some of the benefits of adopting a CI/CD approach are:

- Continuous Integration will improve collaboration between the development teams.
- Due to continuous and automated tests that follows every merge into the master branch, the quality of our software will be improved because bugs will be detected earlier in the testing phase, rather than after deployment.
- With Continuous Deployment, we can roll out updates, patches, and fixes to our software product in real-time without any downtime, thus increasing customer retention and satisfaction.

Why CI/CD?

- CI/CD will drastically reduce the amount of deployment failures or downtimes experienced due to bad code that was pushed into production.
- With Automated testing and deployment, our developers can actually spend more time on “writing codes”; thereby improving their efficiency.
- With CI/CD, we can easily scale-up/scale-down our infrastructure through automated deployment and automated configuration; thereby eliminating the cost of idle resources.
- CI/CD makes it easy to roll back the state to a previous iteration in the event of bugs or inconsistencies in the deployed code.