CI/CD for Udapeople

Reliability, Automation and Smoothness for profitability.

Our Process

Integration

Testing

Security Analysis

Build

Deployment

Maintenance

The problem

Integration: Code reconciliation takes time and requires a waiting period to aggregate and batch code to be tested and built. This often results to time wasting as it prevent early finishers from moving to the next task and this leads to waste of revenue.

Testing: Tests are usually done by dedicated personnel who spend bulk of their time on these tests which can be automated. Automating these test will let these personnel focus on other productive task which will increase our revenue.

Security Analysis: Time spent on manual testing could be further used in developing the next important feature which will reduce time spent on developing a solution and hence reduce cost of production.

Build and Deployment: Our build and deployment process can be automated.

Maintenance: Maintenance is currently hard to do and it is difficult to catch errors, foresee a possible downtime and and prevent service failure due to lack of metrics from monitoring. CI/CD can help us with monitoring and prediction which will help us avoid costly downtime, increase customers’ trust and save revenue .

The Solution

CI/CD

Continuous Integration + Continuous Deployment = Continuous Delivery

Continuous Integration / Continuous Deployment is the new solution to the problems stated earlier. This practice helps us automate all our processes in a set of tasks/jobs executed one after another to ensure all requirements are met and our customers get the changes made to our product more frequently and reliably.

What will we benefit?

Stage	Benefit	Financial Benefit
Code Integration	Less time is wasted on fixing bugs and waiting for others.	Reduce cost
Code Testing	Bugs are detected early and do not make it into production	Avoid cost
Security Analysis	Security risks and loop holes are detected early	Avoid potential cost of lawsuit and fixing damages.
Deployment	Less human error, faster deployments.Lesser time to shipping new features to market. Reduced downtime from crashes and cyber attacks.	Avoid cost, Protect revenue and image
Roll back	Easy to revert back to previous state where things were working perfectly.	Protect revenue.

What next?

There are several resources on adopting CI/CD and creating a great CI/CD pipeline.

I would be happy to share a few more on request. Here are some good places to start:

Quick links to brief explanations:

https://www.redhat.com/en/topics/devops/what-is-ci-cd

https://www.synopsys.com/glossary/what-is-cicd.html

Short Video:

https://www.youtube.com/watch?v=scEDHsr3APg