Benefits of CI/CD

CI: Continuous integration
CD: Continuous Deployment/Delivery

CI is the practice of merging all developers' working copies to a shared mainline several times a day.

Currently Our development team wait a week or even more to deploy their changes and wait for feedback from users to start addressing bugs.

Imagine if our team can start pushing small changes multiple times a day and start receiving feedback quickly.

Implementing CI can help us avoid the following issues:

- Going through integration hell every time we finish a feature
- Code gets lost because of botched merges
- Unit test suite hasn't been green in ages
- No more neglecting the unit tests
- No more leaving broken code in place

These points are only a sample of the gains we might achieve if we start implementing Continuous Integration.

Continuous Delivery is an engineering practice in which teams produce and release value in short cycles.

Continuous Deployment is a software engineering approach in which the value is delivered frequently through automated deployments.

By adopting both practices we can achieve automatic deployment of our code which will decrease our release window significantly by making fast releases of our product.

Implementing CD can help us avoid the following issues:

- Deployments contribute to schedule slip
- Only one engineer can deploy a system
- Deployments are not cause for celebration
- No more manual deploying to environments

Adopting both CI/CD also has some requirements which can be:

- No more manual deploying to environments
- No more modifying environment settings in GUI's
- No more neglecting the unit tests
- No more leaving broken code in place
- Requires a high level of discipline
- Requires additional skills to maintain and extend automation