What is CI/CD?

- **Continuous Integration** – Continuous integration refers to all of the tasks revolving around the *code*. CI is used to seamlessly integrate changes from all developers, and usually culminates in an *artifact*

- **Continuous Deployment** – Continuous deployment refers to the act of (optionally) spinning up the infrastructure to run your application on, performing environment set up and deploying your application to that infrastructure

- **Continuous Delivery** – Continuous delivery refers to the concept that we should be able to continuously deliver smaller updates and features to the client in a timely fashion, while also ensuring the quality of our product and code

Current Status

- Our release process is error-prone and leads difficulty maintaining timely production deployments.

- Since we don't perform checks or quality analysis on a regular basis, this leads to deterioration of the quality of the codebase.

- Deployment is in the hands of a single expert. This results in a single point of failure, as well as key person risk!

- We essentially lack the capability to run automated smoke tests and rollbacks – this results in a large operational risk due to potential downtime.

- We are unable to flexibly respond to problems or requests from our business counterparts in a timely fashion.

Udapeople – CI/CD Value Proposition

- Problem: Our release process is error-prone and leads difficulty maintaining timely production deployments.

- Solution: Implement continuous integration and continuous deployment:

  - Create CI server, and run automated compilation, testing, code analysis and regression tests.

  - Automate infrastructure creation with the CI server, as well as smoke tests and rollback functions.

- Benefits:

  - Reduced costs due to botched deployments.

  - Maintain revenue by reducing downtime.

  - Improved reputation/brand by improving confidence in our product.

Udapeople – CI/CD Value Proposition

- Problem: Low responsiveness to new client requests and new business ideas.

- Solution: Implement automated smoke tests, an end to end testing suite, and rollback mechanism:

  - Implement a form of test driven development

  - Curate an end to end testing suite

  - Improve code architecture to facilitate testing

- Benefits:

  - Improve revenues and reputation by implementing newer features faster.

  - Higher confidence by business and stakeholders in the product.

  - Improved competitive edge due to lower time to market.

Udapeople – CI/CD Value Proposition

- Problem: Untenable operational risk due to key person risk.

- Solution: Implement CI/CD server with automated unit tests, as well as deployment and promotion jobs.

  - Utilize CI/CD server for all steps of code checking, deployment and promotion to production.

  - Prevent developers from making manual changes to deployment environment.

  - Democratize access and education about all steps from code testing to promotion to production.

- Benefits:

  - Reduction in operational risk due to removal of key person risk – this reduction in risk cannot be overstated!!

  - Revenues protected due to improved response time to failing deployments due to higher overall education of the testing/deployment process.

  - Improved confidence into product due to improved visibility into development and deployment process.