Continuous Integration-Continuous Delivery

Why CI/CD

30/08/2022

Continuous Delivery is an approach that informs and enhances the practices of Continuous Integration and Continuous Delivery.

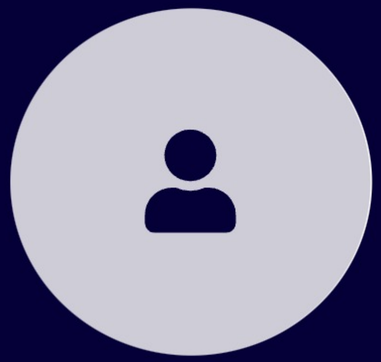
Continuous Integration The practice of merging all developers' working copies / source code to a shared main-stream.

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

Continuous Delivery = Continuous Integration + Continuous Deployment

In summary CI/CD is a set of development practices aimed at automating the building, testing, and deployment of applications.

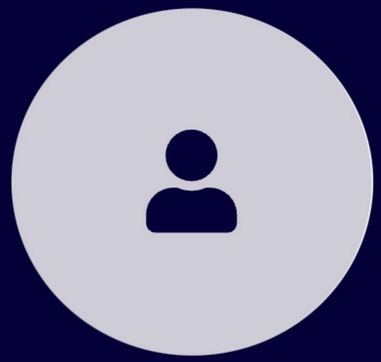
Team



Ade

Project Manager

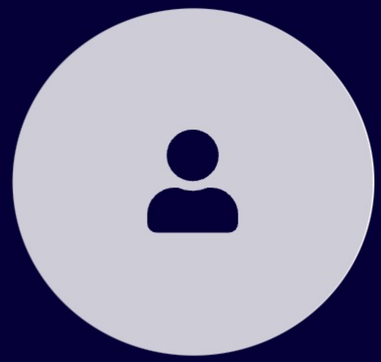
Needs to ensure a quality Product and faster Delivery.



Sam

Front-end Developer

Needs to have testing and build automated.



Ola

Back-end Developer

Needs to have testing and build automated.

Ayo

DevOps Engineer

Needs to prevent making configuration on every build

Exercises to Build Confidence and Move Closer to Continuous Delivery

- Expect an high level of collaborative and comprehensive grooming of features that include team and stakeholders
- Breaking down of features into smallest chunk of valuable increments
- Builds deep understanding of each feature's requirements and characteristics before coding starts
- Write comprehensive automated unit tests in front-end and back-end layers
- Shoot for high coverage from automated back-end integration tests
- Shoot for high feature critical-path coverage from end-to-end UI tests
- Include automated smoke tests that can be run on production-candidates
- Ensure all post-commit tasks and hand-offs must be automated in CI/CD
- Strive for quick, reliable rollback if smoke tests fail

Metrics

Performance Indicators

70%

Increases the and reliability of software

74%

A faster time to market/ launch. Due to a continuous Monitoring/Feedback.

38%

Via automation, CI/CD reduces the capital for the Engineering team.