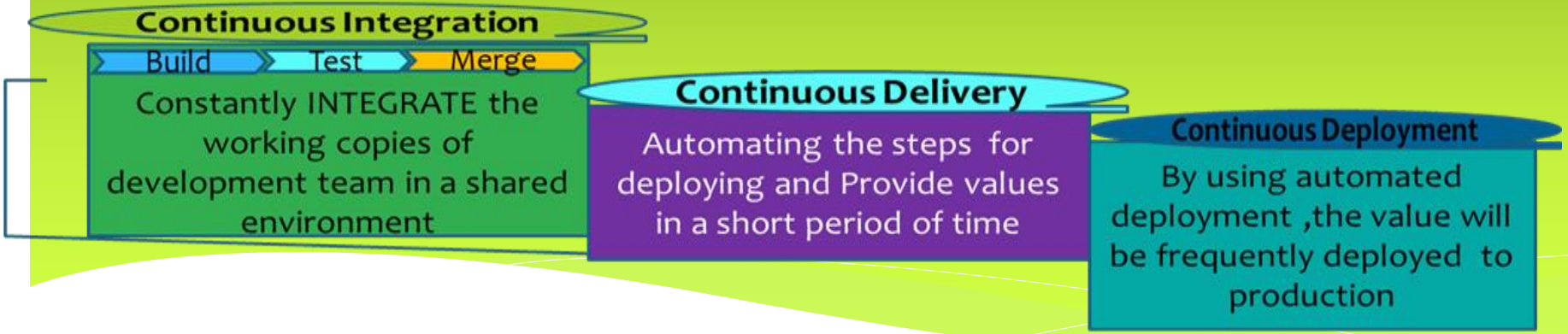
Fundamentals and Benefits of CI/CD to Achieve, Build, and Deploy Automation for Cloud-Based Software Products



Proposal Submitted By: UdaPeople Team

Overview

* The fundamentals of CI/CD
* The benefits of CI/CD
* The situational analysis of the udaPeople
* How UdaPeople benefit from the implementation of CI/CD

Fundamentals of CI/CD

CI/CD stands for a Continuous Integration , Continuous Delivery and Continuous Deployment.

Continuous Integration:

- It is a software development practice where developers regularly merge their code changes into a shared mainline several times a day, after which automated builds and tests are run.
- Continuous integration most often refers to the build or integration stage of the software release process

Continuous Delivery:

- It is a software development practice where code changes are automatically prepared for a release to production with the presence of a manual approval to update to production.

Continuous Deployment:

- It is an approach that automatically releasing a developer's updates from the repository to production, where it is usable by the end users faster and more cost-effectively by managing small, incremental software changes.
- In general CI/CD allows for the development teams to deliver code changes more frequently and reliably without compromising on quality.

Benefits of CI/CD

- Efficient, effective, maintainable, and reorganized workflows
- Reduced costs with increased quality
- Reduce risks and delivering confidently
- Enhanced teamwork and collaboration
- Require less manual effort as a result provide ample time to creativity and frequent release
- Less uncertainty; rollback is easier and better code quality
- Easily repeatable; lesser time-to -market

Why UdaPeople needs CI/CD

* The UdaPeople product is a revolutionary concept and it is also a large application, therefore ,maintenance will might be an issue if the team followed the manual/traditional deployment processes.
* It is difficult for developers to understanding the entire application; this creates a high-pressure, and high-stress situation for UdaPeople developers;
* The application will be very slow and error prone;
* The team spending more time on debugging issues than releasing of new features;
* Leaving broken code might be in place;
* The unit test might be neglected;
* Undocumented changes (configuration drifts) might be resulted;
* Due to a buggy and multi-day deployments, low motivation and high turnover will be resulted;
* Shipping a buggy product can easily result in losing of clients;
* Roll-backs might be impossible and inspections are painful.

*** Therefore, as a solution, the UdaPeople team recommended the application of CI/CD over the manual deployment which contribute to cost savings, profitability, and a higher-quality end-products.

Why Udapeople needs CI/CD Cont'd

Using CI/CD:

No	Outcome	Implication
1	Maximize developer time, minimize release risk, and empower stakeholders to release new features and become creative.	Reduce Cost
2	Teams spend less time on testing and bug fixes	Avoid Cost
3	Early detection and resolving of vulnerabilities	Avoid Cost
4	Generating extensive logs	Reduce Cost
5	It helps significantly reduce the team's burnout and establish a sustainable release of new features and leads high production; decreased employee dissatisfaction; low turnover and high retention of experienced employees	Avoid Cost

Why UdaPeople needs CI/CD Cont'd

Using CI/CD:

No	Outcome	Implication
1	Code quality will be higher and features are delivered faster	Increase Revenue
2	Enable to deliver high-quality UdaPeople products to our clients faster and respond to feedback as it comes in; referral rates will be higher	Protect Revenue
3	Increase collaboration among team members as well as across a whole range of functions; it provides an opportunity to free up individuals' time so they can focus on innovation	Increase Revenue
4	Easily roll back changes quickly	Protect Revenue