UdaPeople

PROPOSAL OF CI/CD IMPLEMENTATION FOR INCREASED REVENUE RETURNS

WHAT IS CI/CD?

Continuous Integration

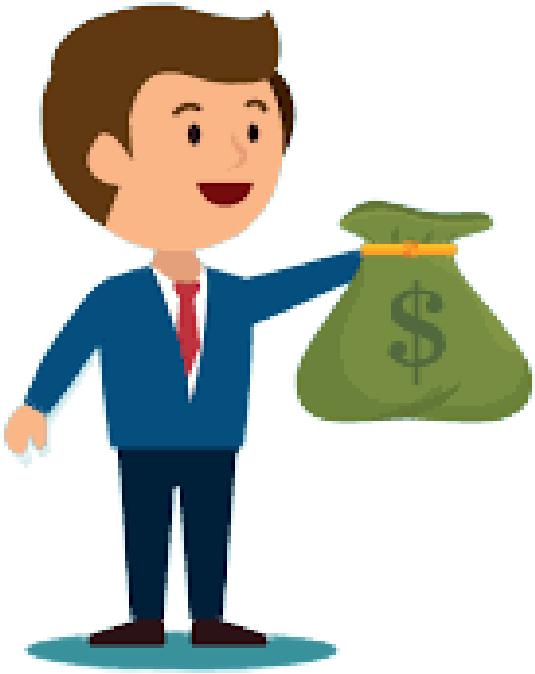
Continuous integration is the process of regularly merging all developers' working copies to one shared mainline several times a day.

Continuous Development

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

BENEFITS OF CI/CD

- Create Revenue
- Protect Revenue
- Control cost
- Reduce cost

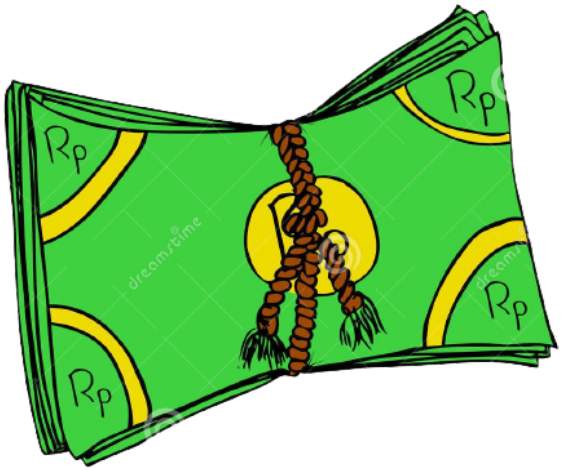


Create Revenue

Teams have the flexibility to update applications and build and deploy new ones in response to emerging trends, new markets, and evolving expectations. This way, early feedback and usability issues can be addressed without the need for major time-consuming refactors to change the direction later on.

Protect Revenue

Long-term downtime does not only influence our revenue negatively but also undermines Udapeople's credibility. With CI/CD pipeline adequately automated, product deployment and release process are more reliable and developers are not wasting hours on bug-fixing. Our clients trust us to deliver because we become more reliable.



Control Cost

Recently The American Airlines scheduling glitch allowed pilots to drop thousands of July flights. A lot of flight trips were affected which made them lose money. The Goal of Good unit test practices in a CI/CD environment helps to identify and resolve as many issues as possible before the product is released to production. This saves significant amounts of project time, and dollars and substantially improves the overall quality of Udapeople.

Reduce Costs

CI/CD is also good for the bottom line. It standardizes deployment processes across all projects, and, done right, it enables Udapeople teams to systematically test every change made to the source code. As a result, this process stands to dramatically reduce the likelihood that any bugs or errors slip through the cracks and cause problems down the line. Done right, this practice can lower development costs by eliminating many of the costs incurred while building and testing code changes.