CI/CD stands for continuous integration and continuous delivery/deployment which is a set of practices that speed up the software life cycle. CI/CD involves the automation of the processes of the software development life cycle which ensures speeding up the entire process and also guarantees only quality products are released to the market.

Listed below are some of the major benefits of the implementation of CI/CD:

Superior code quality:

From the beginning of the CI-CD process, the product's quality is given very important and should not be affected by any means. The quality of the product gets more effective because of continuous testing and building. This helps in identifying the flaws that might arise in the future.

Because CI/CD pipelines offer test automation and developers can know about code problems nearly in real-time. That concept of “failing fast” means teams aren’t wasting time or resources with buggy code, and devs aren’t occupied with endless “fix” requests when they’ve moved on to other projects. Time and money are saved. This could also give an advantage over competitors.

This aspect helps Udapeople to **SAVE COST** because there will be no costly security holes

Deliver faster with an accelerated release rate:

The implementation of CI/CD will boost the rate of software releases. The faster code is released, the more new code can be developed, and then released. The automation process of CI/CD ensures it provides early warning of any potential integration problems rather than waiting till the end of the project to see if all of the software will integrate successfully. This helps ensure there is no delay in the release of the product to the target market.

Delivering to production faster **increases your revenue** because we will be deploying new features faster and more quickly.

CI/CD pipelines: Automation reduces the cost

Eliminating human intervention from the software development process helps save time and money. CI/CD automates the handoffs, the source code management, the version control system, the deployment mechanisms, and, of course, so much of the testing. Testing has been identified as the number one reason releases were delayed. Not only do delayed releases impact the business from a cost, branding, public relations, and even a reputation perspective, they are deadly to businesses relying on speedy time-to-market. Historically software testing was manual and incredibly time-consuming, which is why companies only released new code once or twice a year. In today’s world, companies have to release all the time, and automated software testing is critical to making that possible.

Automation helps **reduce cost** as we do not spin up unneeded infrastructure after deployment switch of resources.

Fault isolation and Continuous feedback:

Before DevOps and CI/CD became common in software development, development teams would know there was an issue with code but would struggle to know exactly where the problem was happening. CI/CD and its automated testing have changed that. Developers can easily identify and then isolate code faults, dramatically improving productivity. By implementing CI/CD, whatever happens, you will get feedback to learn from and move forward. So you will be getting assured that the end product would be the effective one. Whenever any test fails, the feedback is sent to the authorized team of the same and directs them to look into the failure. It then gives out a more effective product. This eliminates the possibility of downtime and loss of productive time.

Automated smoke tests protect your revenue because there will be a reduction in downtime from deployment-related issues or bugs