FUNDAMENTALS OF CI/CD

CI/CD is a core part of DevOps. More than a methodology, it's an **approach that introduces automation and monitoring** throughout the different stages of an app's lifecycle. This practice aims at empowering development teams to accelerate software release cycles, with increased reliability and security.

Continuous Integration is the practice of merging a developer's work in a single shared repository regularly, using different versions for each update. This way, all developers are working from a common code base and all development efforts have a shared set of foundations.

Continuous Delivery or **Continuous Deployment** used interchangeably refers to the automatic preparation of code and code changes for release in production, and the automation of testing and code release.

Continuous Deployment eliminates the need for human intervention and for on-demand release to production: instead, **code moves automatically from integration to release into a** production environment, and testing is integrated into the process.

BENEFITS OF CI/CD

Reduced business risk and cost

Without CI/CD finding and fixing bugs late in the development process is expensive and time-consuming. This is especially true when there are issues with features that have already been released to production.

With a CI/CD pipeline, you can test and deploy code more frequently, giving testers the ability to detect issues as soon as they occur and to fix them immediately. You are essentially mitigating risks in real time.

This continuous flow of small iterations also means problems have less impact and are easier to remediate which in turn **saves cost** considerably.

Less Manual Efforts, More Automation!

With CI/CD we expend less manual efforts, automation is used right from the start.

Once we build features and check in code, tests are automatically triggered to make sure that the new code does not break existing features and that the new features are working correctly.
After the tests run, the code gets deployed to different environments, including QA, staging and production.

This ensures our application at every point is available and does not have minimal downtime, hence **protecting revenues** of our business**.**

Increased Speed of Delivery

Since CI/CD is automated, our application features can be released multiple times a day.

Faster deployment of higher quality code results in fewer interruptions to the business as a result of broken code, and an overall increase in productivity.

The faster we ship tickets would also go a long way in **driving our revenues up.**

CONCLUSION

Continuous integration and continuous delivery provide an ideal scenario for our organization’s application teams. Our developers simply push code to a repository. This code will be integrated, tested, deployed, tested again, merged with infrastructure, go through security and quality reviews, and be ready to deploy with extremely high confidence.

CI/CD would help our business a great deal. Its automation means our developers would spend more time building priority features/issues, and less time on tracking bugs/errors from failed deployments.