

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

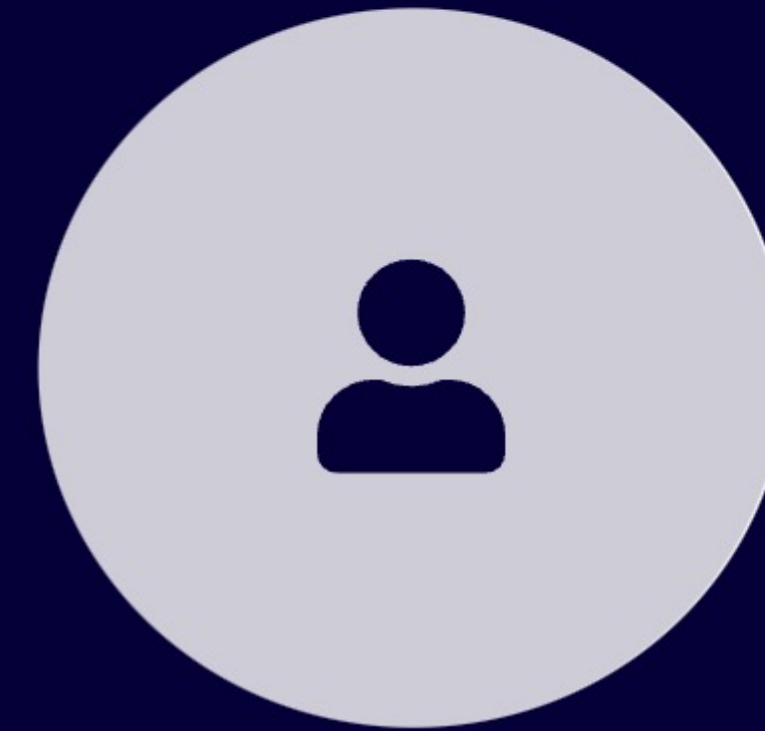
Needs to
ensure a
quality
Product and
faster Delivery.



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

Business Benefits of CI/CD

Reduce Costs and Boost Profits

CI/CD is also good for the bottom line. It standardizes deployment processes across all projects, and, done right, it enables 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.

Teams spend less time on testing and bug fixes, meaning organizations spend less money on tasks that don't provide any value to the business or its customers.

And because CI/CD also makes it easier to deliver high-quality products to market faster and respond to feedback as it comes in, organizations stand to see an increase in profits. Customers stick around longer and will likely recommend your products to others in their network.

Bring Products to Market Faster

Organizations that have effectively implemented CI/CD can bring new products and features to market faster and immediately start generating revenue from the features they deploy rather than waiting for the entire app to be completed (and checked manually) before they can launch.

Instead, teams already know the code is in good shape because they've automated testing, and continuous delivery means code is automatically deployed if it meets predefined criteria. Back-to-back releases are easier and less time-consuming, and, if something isn't working, they can pull features with a single click.

Gain Real-Time Visibility of the Development Process

CI/CD also brings real-time visibility into the development cycle. Reviewing test results helps everyone on the team identify the project status and immediately understand which code changes caused problems or improved on what came before.

Stakeholders can easily see where a project stands at any given moment—spot bottlenecks, inefficiencies, etc., and use those insights to optimize the process.

You can also see the history of deployments and success rates, which can inform the direction of future projects, help teams plan ahead, and keep everyone focused on the tasks that create the most value.

CI/CD Significantly helps out down on

- **Deeply expensive downtime.** Long-term downtime does not only influence your revenue negatively but also undermines your company's credibility.
- **Troubleshooting costs.** With CI/CD pipeline properly automated, the process of the product deployment and release is more reliable and developers are not wasting hours on bug-fixing.
- **Switching/onboarding costs.** Automated CI/CD let new members of the team get down to business right away without having to deal with an exhausting learning curve.
- **Staff turnover costs.** Happy employees – more successful company! When developers are freed from manual, repetitive tasks, they focus on creating new features and code and continuous quality improvement, isn't that right? Sure, it is. With them knowing they can rely on their system, employees become more relaxed and satisfied and, as a result, they stay with you, work harder and become more and more in love with the company.

<https://itsvit.com/blog/how-ci-cd-automation-saves-money-and-time/>

<https://www.3pillarglobal.com/insights/the-business-impact-benefits-of-ci-cd/>

<https://udacity.com>