

BENEFITS OF CICD TO THE COMPANY

DATE: 18TH JULY 2022

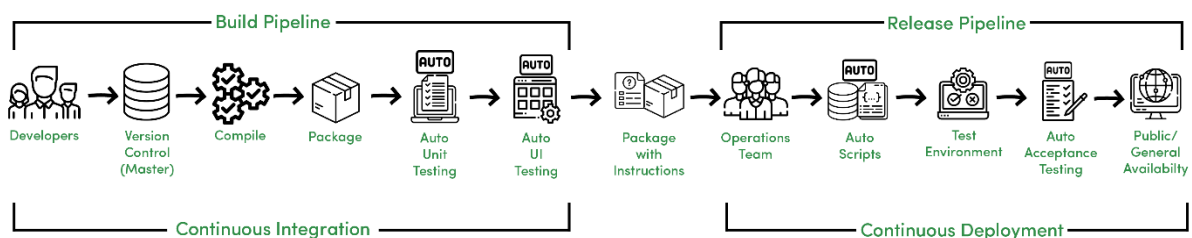
What exactly is CICD, you may ask? CICD, Continuous Integration and Continuous Deployment, in its simplest form, is a tool that takes a code from our team and processes it to make it ready for the market. The processes CICD goes through to make our product ready for the client are; building of code, testing of build to find flaws, analyzing, deployment, verification then finally promoting the product. CICD lists out some practices (best practices) or instructions to be followed to get your code out the door in a timely and safe manner for the satisfaction of our users and to ultimately rake in revenue.

Continuous integration is a development practice that encourages our developers to integrate into a shared repository several times consistently and continuously in a day. CI provides an automated process of building, packaging, and testing new software. Since CI is an automated tool, you are sure to get consistent results each time which is difficult to achieve if done manually.

When using CI, our developers would commit code changes into a shared repository that would trigger an automatic build and test sequence. These changes are always small and provide quick feedback to the developers and alert the developers about errors that arise.

The end product of CI is used as a raw product by CD (Continuous Deployment). CD enables developers to deploy regular software updates to different environments and end-users/clients whenever the need arises.

To get the best out of CICD, smaller and more frequent software updates are preferred as they are less disruptive and are easier to roll back in the case of an error. The team also has the avenue to quickly deliver new products or features which allows the company to meet one of its fundamental goals, satisfying the customer.



Based on the diagram above, once our developers commit code to the Version Control service (GitHub), the trigger is set in the CI tool (Circleci) to pull the commit and build it. Upon successful build with no hiccups, the code is pushed to production and if there is an error in our new build, a simple roll back to the previous version can be made so quickly that the user may never notice. CICD helps to reduce downtime. Downtime in this case refers to an application or server going offline

when not intended to. This can cause the organization major losses in terms of revenue and clients. CICD would allow the organization to avoid downtime while reducing the risk of exposing users to breaking changes when deploying applications.

CICD practices should matter to this organization as it helps get continuous feedback not only from our customers but also from our team. The use of CICD in this Organization would have extraordinary benefits, both financial and technical. Some of the benefits of implementing CICD pipelines in our development processes are:

1. CICD helps to improve the efficiency of the DevOps team. Without CICD, developers and engineers deal with a ton of pressure in their day-to-day activities and bad deploys can put their jobs at risk. Devs are required to write deployment scripts, documents, procedures, and so on for the ops team to execute them. All this human intervention makes the process very inefficient, laborious, and often risky.
CICD, when used properly eliminates manual tasks, prevents coding errors, and detects problems before deployment, hence relieving the team to work faster without compromising quality. Because it takes a shorter time for development teams to find and fix problems during the production process, CICD can dramatically accelerate release rates which would translate to generating more revenue for the organization.
2. CICD improves app quality. One of our biggest concerns this organization may have is that implementing CICD would imply giving up quality in favor of speed. When done right, CICD means we do not need to prioritize quality over speed, instead, it offers the best of both worlds by enhancing cohesive collaboration between developers and operations teams allowing problems to be identified and fixed faster and early in the development lifecycle. This allows teams to do what they do best, writing code that solves real problems rather than worrying about what's going on in the production environment. Additionally, CICD would help our organization maintain quality standards hence maintaining the trust of our clients and also bringing in more business and revenue.
3. CICD reduces costs and boosts profits. CICD serves as a good bottom line. It standardizes deployment processes across all projects and if done right, it enables teams to systematically test every change made to the source code. As a result, the likelihood that any bugs slip through the cracks and cause problems is drastically reduced. This process can also lower development costs by eliminating many of the costs incurred while building and testing code changes.
Teams also spend less time on testing and bug fixes, so organizations do not have to spend money on tasks that do not provide any value to the organization or its clients.
4. CICD would enable our organization to bring products to the market faster. Should our organization effectively implement CICD, we would be able to bring new products and features to market faster than normal and effectively start generating revenue from the features we deploy rather than waiting for the entire app to be completed before we launch. We would always have confidence in our code because we would have automated the testing process, and also, continuous deployment means the code is automated provided it meets the predefined criteria. In the case, an issue arises and something isn't working as it should, we can simply roll back to what was working.

In conclusion, CICD is a system that was made to make life easier for our developers. It allows for quick and seamless system updates, easier problem solving, and a better customer experience. These valuable benefits make CICD an amazing tool to integrate into the business to add efficiency to our business and increase revenue.

References

Diagram credit:

<https://media.geeksforgeeks.org/wp->

[content/uploads/20210109235224/GFGTechnicalScripterImg3Black01.png](https://media.geeksforgeeks.org/wp-content/uploads/20210109235224/GFGTechnicalScripterImg3Black01.png)