CI/CD

A better way to ship products to consumers.

CI/CD aids achievement of build and deploy automation within the product lifecycle and faster release to consumer culture.

Objectives

- The primary goal of a CI/CD pipeline is to deliver working software to users quickly and frequently.
- Protecting the revenue of the organization through early failure detection mechanisms offered by CI/CD and thereby ensuring defective products never leaves the production line; the company does not incur huge costs of payments for product recalls.

- There is also the need to prevent embarrassing and costly security holes and vulnerabilities, which can cause severe loss in revenue for the organization.

Benefits of CI/CD

These are some of the many benefits of implementing CI/CD process;

Shorter review time

With continuous integration, our developers are encouraged to commit their code changes more frequently – at least once a day as a rule of thumb. Sharing code with the rest of the team regularly not only ensures everyone is building on the same foundation, but also results in faster code reviews and makes it easier to integrate changes.

Better Code Quality

Once we have invested in a first layer of automated tests, we can save plenty of time on running those tests manually and focus on developing additional layers of automated tests – such as end-to-end or performance tests – and on manual exploratory testing.

Automating tests ensures they are performed consistently, making the results more reliable. Because automated tests are quicker to run then

their manual equivalents, it becomes feasible to test much more frequently.

Testing your code regularly and thoroughly means you'll discover bugs sooner, making it easier to fix them as less functionality has been built on top of them. Over time this results in better quality code.

Measurable Progress

Many of the tools available to support an automated CI/CD pipeline also instrument the process, providing us with a whole host of metrics from build times to test coverage, defect rates to test fix times. Armed with this data we can identify areas that might need attention so we can keep improving your pipeline. Slower builds may indicate a need to increase capacity while an increase in mean fix times might be a sign of a process or cultural issue.

Conversely, metrics can also provide reasons to celebrate, and so they should; consistently extending our code test coverage, reducing our defect rate or increasing our release frequency all belong on the team's collective brag sheet as signs of a great working culture. Being able to measure how your

CI/CD pipeline is supporting our organization’s goals is another advantage of the practice.

Summary

CI/CD techniques have proven to have significant amount of benefits for our organization in comparison to the negatives and I believe that with the few stated benefits, among many others, that will prompt us to make the right call for the immediate and long-term benefits for the organization.