

**THE FUNDAMENTALS AND BENEFITS OF CI/CD TO ACHIEVE,
BUILD, AND DEPLOY AUTOMATION FOR CLOUD-BASED
SOFTWARE PRODUCTS**

A PROJECT PROPOSAL WRITTEN

BY

ROBERT CHUKWUEMEKA

7TH JULY, 2022

THE FUNDAMENTALS AND BENEFITS OF CI/CD TO ACHIEVE, BUILD, AND DEPLOY AUTOMATION FOR CLOUD-BASED SOFTWARE PRODUCTS

Fundamentals of CI/CD

- **Continuous Integration:** The practice of merging all developers' working copies to a shared mainline several times a day.
- **Continuous Delivery:** An engineering practice in which teams produce and release value in short cycles.
- **Continuous Deployment:** A software engineering approach in which the value is delivered frequently through automated deployments.

With CI/CD, you can visualise the development cycle from commit to production, continuously merge code, and repair detected bugs. You can release software to production multiple times after thorough testing without compromising quality parameters. CI/CD always keeps the code in a release-ready state as changing code frequently is more of a routine now in modern software development.

Developers release code into a shared repository in small batches, which enables them to conduct parallel testing. Rather than working in isolation, they share their builds with the entire team frequently. Teams collaborate to identify critical bugs, which ensure that bad code does not make it to production. Thus CI/CD implementation drives business growth by providing high-quality releases that have fewer errors and bugs.

The CI/CD Pipeline: The elements of the CI/CD Pipeline include; build, test, analyse, deploy, verify, and promote as shown in the figure 1 below.

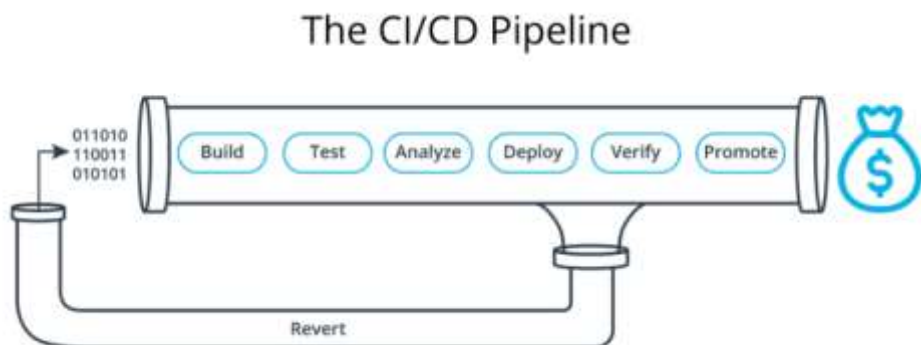


Figure 1: The Elements of CI/CD Pipeline

THE BUSINESS BENEFITS OF CI/CD

So, what are the business benefits of continuous integration and continuous delivery? We shall explore them below.

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.

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.

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.

CI/CD Plays A Crucial Role In Shortening Time To Value.

CI/CD fosters a culture of innovation, allowing developers to experiment with new technologies and try out new ideas. Teams can test different features with real users in parallel and use their findings to ensure

Frequent updates, new feature releases, rapid response to feedback, and quick bug fixes all play a major role in customer satisfaction and long-term loyalty. By using CI/CD, organizations can continuously build on applications and enhance the experience without the risk of downtime or interruptions.

Allows Developers to Deliver Products that Consumers Want Now

Over the last couple of years, customer-centricity has been a core focus for businesses. CI/CD enables organizations to respond to consumer needs as they evolve. Teams have the flexibility to update applications and build and deploy new ones in response to emerging trends, new markets, and evolving expectations. That way, early feedback and usability issues can be addressed without the need for major time-consuming refactors to change the direction later on.”

Boosts DevOps Efficiency

Without CI/CD, developers and engineering teams deal with more pressure in their day-to-day—service interruptions, outages, and bad deploys can put their jobs at risk. CI/CD automation can eliminate manual tasks, prevent coding errors, and detect problems before deployment, allowing teams to work faster without compromising quality. Because it takes less time for development teams to find and fix problems during the production process, CI/CD can dramatically accelerate release rates. What’s more, organizations can support recurring releases if code is developed in an automated testing pipeline.

With CI/CD, teams can also implement a standardized delivery mechanism that automatically merge codes, run tests, and deploy changes to multiple production environments to ensure that code is always in a release-ready state.

CI/CD Improves App Quality

One of the biggest concerns organizations have about implementing CI/CD is that they’re giving up quality in favour of speed. CI/CD helps organizations maintain quality standards as apps expand and evolve over time. Applications with large feature sets can get too big to feasibly run proper code review, meaning you may see a drop in quality over time. CI/CD automation allows developers to deploy code more frequently and make incremental

improvements to the code. Teams can automate regression testing and parallel tests, which improves test coverage and ensures that the code is bug-free and works across multiple environments.

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.

Supports Cloud-Based App Development

The rise of CI/CD is, in part, due to the rise in cloud adoption. Technically, you could use traditional development techniques to build cloud-based applications, but it's not really practical. CI/CD enables benefits like scalability, elasticity, and improved performance characteristic of cloud-native applications.

Conclusion

CI/CD will ultimately become imperative for your business – the benefits are simply too many. Migrating from the legacy software release cycles to CI/CD won't just accelerate the time-to-market, but also result in a productive and satisfied development team.