UdaPeople with CI/CD benefits

By

Nagy Ahmed Ezzo

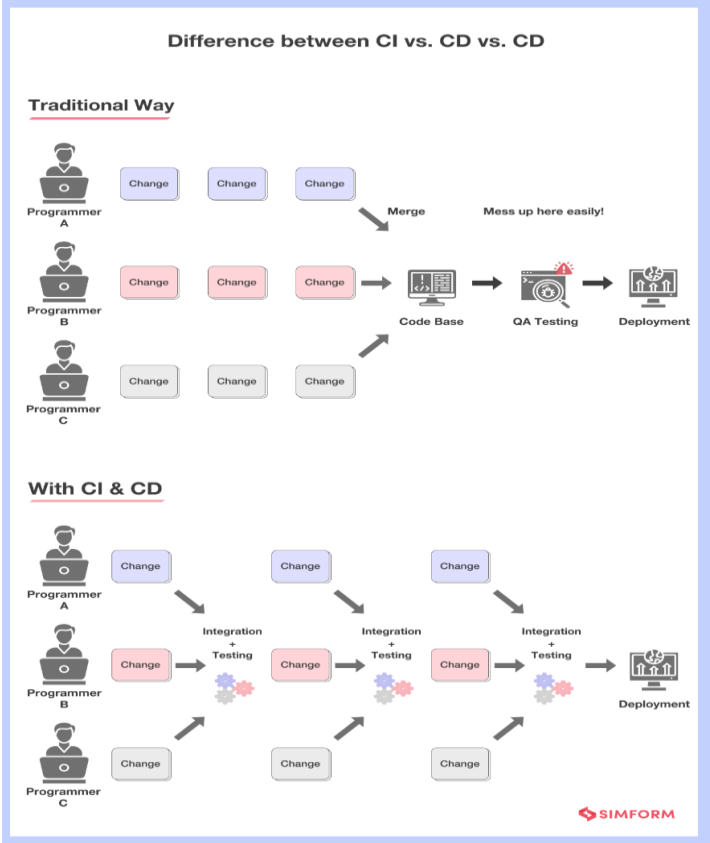
What is CI/CD?

- **Continuous Integration** is a DevOps software development practice where developers regularly merge their code changes into a central repository, after which automated builds and tests are run. Continuous integration most often refers to the build or integration stage of the software release process and entails both an automation component (e.g. a CI or build service) and a cultural component (e.g. learning to integrate frequently). The key goals of continuous integration are to find and address bugs quicker, improve software quality, and reduce the time it takes to validate and release new software updates. (AWS)
- **Continuous Delivery** is a software development practice where code changes are automatically prepared for a release to production. A pillar of modern application development, continuous delivery expands upon continuous integration by deploying all code changes to a testing environment and/or a production environment after the build stage. When properly implemented, developers will always have a deployment-ready build artifact that has passed through a standardized test process. (AWS)
- **Continuous Deployment** is an extends continuous delivery in such a way that it allows frequent automated deployments without any human interaction. typical phases in continuous deployment are infrastructure provisioning, smoke testing, production deployments, and automated rollbacks.

Continuous Delivery vs. Continuous Deployment

With continuous delivery, every code change is built, tested, and then pushed to a non-production testing or staging environment. There can be multiple, parallel test stages before a production deployment. The difference between continuous delivery and continuous deployment is the presence of a manual approval to update to production. With continuous deployment, production happens automatically without explicit approval.

Continuous Integration Benefits	Continuous Delivery Benefits
Improve Developer Productivity	Automate the Software Release Process
Find and Address Bugs Quicker	Improve Developer Productivity
Deliver Updates Faster	Find and Address Bugs Quicker
	Deliver Updates Faster



Benefits your company can get out of an effective CI/CD pipeline

1- Deploy Faster: A CI/CD pipeline is like a turbo engine when it comes to accelerating the code deployment pace and time-to-market of the final product. It enables you to ship code changes not just every week, but every day and even hourly. As testing is automated, the code is automatically deployed if it meets the predefined criteria. This enables you to release software to production several times at a rapid pace while keeping quality intact. Moreover, the automation the CI/CD pipeline brings to the software development lifecycle reduces manual labor and time required for creating and maintaining deployment scripts and tools.

2- Avoid outages: A successful CI/CD pipeline enables your DevOps teams to continuously integrate small batches of code instead of the entire application. This approach helps the developers to easily identify the anomalies and fix them. So, you can avoid significant outages and other key issues by flagging bugs and vulnerabilities before they make it to production and disrupt the entire application.

3- Improve visibility across development: Before the introduction of DevOps methodologies and CI/CD approaches in software development, the developers would easily know there was an issue with code, but they used to struggle to know where exactly the problem was happening. Now, the automated testing practices of the CI/CD pipeline improved the visibility across the software development lifecycle. Developers can easily spot and isolate code issues. This, in turn, significantly improved productivity.

4- Reduce costs of delivery: CI/CD pipeline reduces human intervention across the DevOps lifecycle by automating the handoffs, version controlling, source code management, deployment processes, and testing, among others. This significantly saves the time and money required to develop and deliver high-quality software. Moreover, with a successful CI/CD pipeline in play, the development teams aren't plagued with endless 'code fix' requests, so they can keenly focus on the next projects, maximizing the overall ROI for the company.

5- Improve mean time to resolution (MTTR): A CI/CD pipeline provides high visibility across the SDLC, enabling DevOps teams to identify code issues quickly and fix them at a faster pace. This ability to rapidly resolve the issues improves the key development metric, i.e., mean time to resolution (MTTR). The better the MTTR, the more productively the DevOps teams can work, and the faster software can be delivered. Simply put, the better the MTTR, the better the ROI.