## What is CI/CD?

CI/CD is an approach to app delivery that integrates automation into app-development stages.

CI/CD stands for *continuous integration/continuous delivery or deployment and* grew out of

agile-development practices aimed at reducing app development times and increasing the

number of releases. CI/CD is an umbrella term that includes a collection of operating principles

and practices that empower DevOps teams to more quickly and easily implement code changes

to apps.

Altogether, CI/CD produces a system where developers can rely on a high degree of automation

and monitoring to improve app development.

# Why is CI/CD Important?

**CI/CD helps you build a desirable team developers want to join**

A working CI/CD methodology shows prospective hires that you are dedicated to supporting them, and that you have the right tools and processes in place to promote a better working experience with fewer development hurdles.

**CI/CD improves collaboration**

CI/CD eliminates much of communication complexity, automating many responsibilities and allowing developers to focus on their individual tasks. Continuous integration into a shared version-control repository helps ensure that as individuals within the team perform their assigned jobs, the overall structure and function of the app remains sound.

**CI/CD maximizes reliability, automation, and efficiency**

The continuous nature of CI/CD means more apps and application fixes, delivered more quickly. But CI/CD doesn't provide improved quantity at the expense of quality; automated tools constantly test new code changes, ensuring desired functionality before sending the changes to production or deployment.

Fault-detection and isolation tools pinpoint the root causes of any errors. Continuous-reliability tests focus on snapshots of code as they are introduced to the system. Non-critical defects are dealt with as they arise, reducing your team's backlog. The end result is an automated approach

that allows developers to produce more-complete software solutions at lower cost and in less time.

**Business Benefits of CI/CD**

Getting code out of the door with the risk of discovering showstoppers like critical bugs found late, infrequent releases, too much manual work, or late feedback has spawned a need for adopting Continuous Integration and Continuous Deployment/Delivery (CI/CD) technologies. For any competent software house, CI/CD is evolving as the modus operandi for rapid software delivery.

**Superior Quality Code**

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 ensures 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.

**Accelerated Release Cycles**

With CI/CD, you can visualize 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.

**Loyal Customer Base**

The on-demand delivery experience that customers insist on leaves little room for errors and bugs. Matching them with the best fulfillment is critical to capturing market share. Implementing CI/CD reduces both lead time and response time, which in turn minimizes release risk and maximizes developer time. A stable operating environment and faster resolution to production requirements enable you to cement customer loyalty and create great experiences.

**Cost Deduction**

As per one of the recent Forbes Insights surveys, "Three out of four executives agree that the amount of time, money, and resources spent on ongoing maintenance and management—versus new project development or new initiatives—is affecting the overall competitiveness of their organization."

Delivering high quality tailor-made solutions to address business-specific challenges requires a way to meet the rapid time constraints imposed by rivals. Embracing CI/CD is the perfect fix to shorten the time to finish a project and market new features. The shorter the development cycle, the higher the chances are to meet ambitious time-to-market goals.

**Improved Mean Time To Resolution (MTTR)**

With small code changes and quick fault isolations, adopting CI/CD plays a crucial role in reducing the MTTR, thus keeping deployment failures to a bare minimum and recovering

quickly from failures is a critical business risk assurance for those embarking on the CI/CD

journey.