What is DevOps?

DevOps is a set of practices that combines software development (Dev) and IT operations (Ops).
It aims to shorten the systems development life cycle and provide continuous delivery with high software quality.
DevOps is complementary with Agile software development; several DevOps aspects came from the Agile methodology.

The goals of DevOps span the entire delivery pipeline. The ROI benefits include:

- Improved deployment frequency;
- Faster time to market;
- Lower failure rate of new releases;
- Shortened lead time between fixes;
- Faster mean time to recovery (in the event of a new release crashing or otherwise disabling the current system).

What is CI/CD?

CI/CD bridges the gaps between development and operation activities and teams by enforcing automation in building, testing and deployment of applications. Modern day DevOps practices involve continuous development, continuous testing, continuous integration, continuous deployment and continuous monitoring of software applications throughout its development life cycle. The CI/CD practice or CI/CD pipeline forms the backbone of modern day DevOps operations. [*]

Continuous integration (CI) is the practice of merging all developers' working copies to a shared mainline several times a day. [*]

Continuous delivery (CD) is a software engineering approach in which teams produce software in short cycles, ensuring that the software can be reliably released at any time and, when releasing the software, doing so manually.[1][2] It aims at building, testing, and releasing software with greater speed and frequency. The approach helps reduce the cost, time, and risk of delivering changes by allowing for more incremental updates to applications in production. [*]

Continuous deployment is a software engineering approach in which software functionalities are delivered frequently through automated deployments.[1][2][3] CD contrasts with continuous delivery, a similar approach in which software functionalities are also frequently delivered and deemed to be potentially capable of being deployed but are actually not deployed. [*]

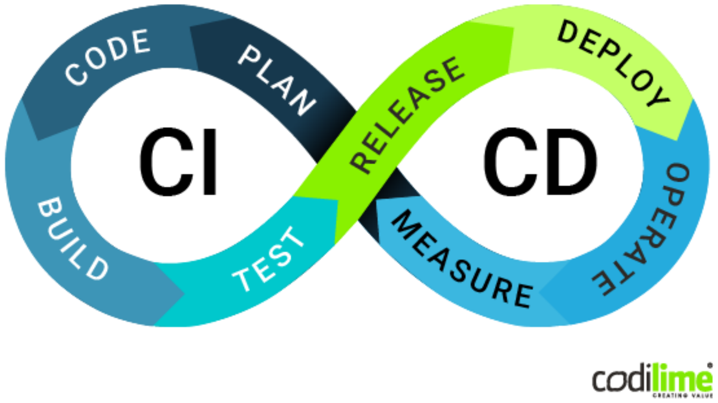
[*] From Wikipedia, the free encyclopedia

Business Benefits adopting CI/CD

Shorter time-to-market of new features
If code changes are small, you can release software builds faster—essentially, on demand. This improves flexibility and the ability to ship new functionalities. Now when users ask for new features or competitors introduce enhancements, your reaction can be faster.

Automation to reduce costs and save time
Automation is one of the most important reasons to implement a CI/CD pipeline. Build automation, automated testing and deployment not only make the life of the DevOps team easier, but also slash costs. Automated actions are also less susceptible to human error and easier to manage. Ideally, every deployment into the production environment should be performed without human intervention. This is what Continuous Deployment is about.

Increased speed of innovation and ability to compete in the marketplace
Two identical companies: One implements CI/CD technology and the other doesn’t. Who do you think deploys applications faster? While this seems like a silly comparison, because of course the company with more automation deploys faster, there are organizations out there still convinced they don’t need CI/CD because they’re not looking at their competition. Organizations that understand the importance of CI/CD are setting the pace of innovation for everyone else.



Conclusion

The ultimate goal of every CI/CD implementation is to make customers/clients happy. Buggy software can harm a company's reputation, sometimes irreparably. Fast and frequent releases, new features shipped regularly, bugs fixed promptly and immediate reaction to feedback—these are the major factors that will make end-users happy to pay for your software. They can be achieved thanks to embracing CI/CD in your company.