CI/CD BENEFITS

UDAPEOPLE

OVERVIEW

- CI/CD explained
- What are the current obstacles facing our company?
- How CI/CD helps overcome those obstacles.
- The challenges we are likely to face.

CI/CD EXPLAINED

1. What is Continuous Integration (CI)?

Continuous integration implies that new functionalities/ user stories are built, tested, and integrated into the already running system.

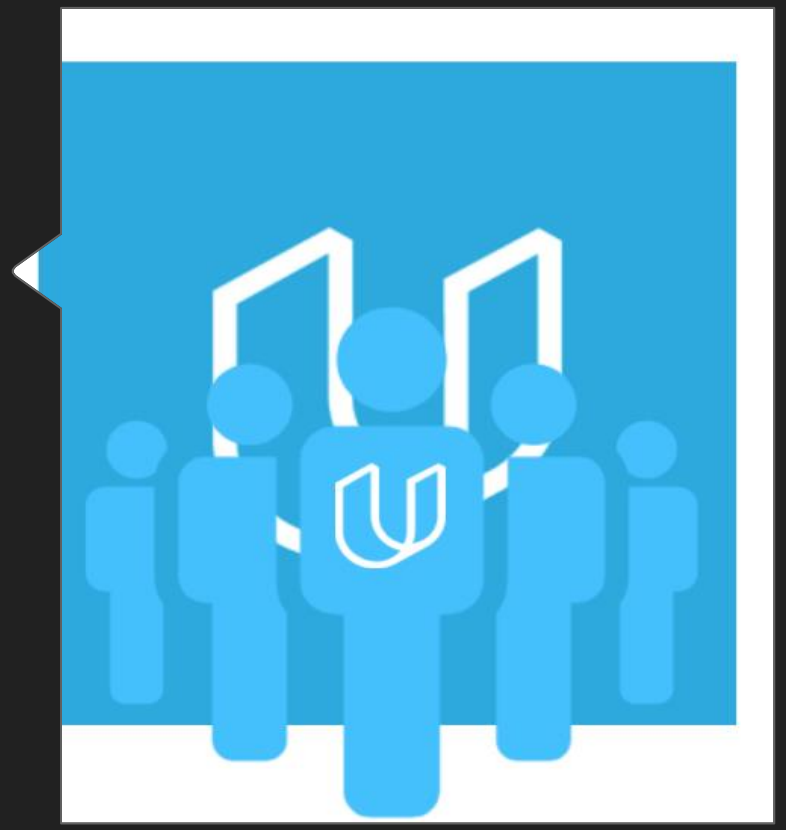
In the CI/CD development process, the developer team can submit the code changes more, while continuous testing speeds up issue fixes and functionality, which leads to better teamwork and higher-quality product in the end.

2. What is Continuous Delivery (CD)?

Continuous delivery makes ensuring that new user stories are automatically tested, sent back for issue fixes if necessary, tested again automatically, and finally sent to a central repository. The additional functionality can then be made available in the production environment.

3. What is Continuous Deployment (CD)?

Continuous deployment denotes the automatic release of new user stories in a shared repository, indicating that recently introduced features are accessible to end users.

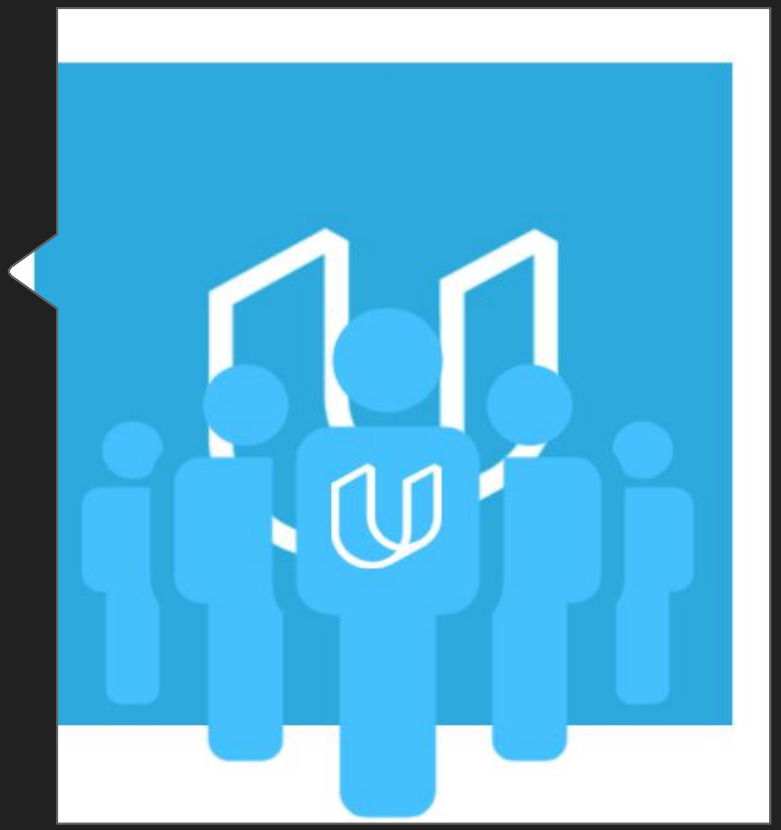


What are the current obstacles facing our company?

1. The release process is now manual, which makes it prone to errors and causes disruptions in production.

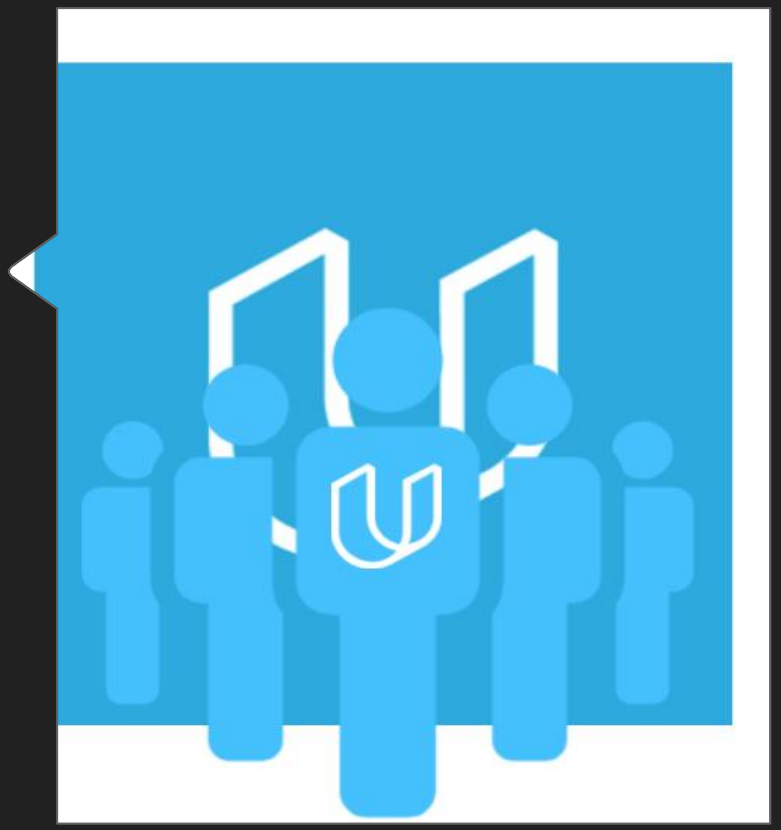
2. This costs time and affects software quality.

3. Deployments get complex, the process needs helpers scripts and automated testing and rollback mechanisms.



How CI/CD helps overcome those obstacles.

1. Using automation, the DevOps team, which combines development and operations, can concentrate on what they do best. Teams also won't need to wait for human approvals from one another in order to execute integration, delivery, and deployment.
2. Better code quality with code being added to the central repository almost daily. These everyday additions make continuous integration and delivery easier as there are small code fragments to deal with, thus less probability of emerging anti-patterns and bugs.
3. Ensures faster time to market, because the key objective of CI/CD is to shorten the years-long time to market that was previously required because of flawed procedures and insufficient communication between development and operations.



The challenges we are likely to face.

1. Introducing CI/CD might be overwhelming at first.
2. Possible high initial learning cost for our team.
3. Team Communication; Delivering CI/CD needs constant support and maintenance.
4. If not done correctly, software's performance issues could affect software and overall scalability.

The challenges need to be made aware of, but they could be overcome with good planning and execution. On the long run, costs could be dramatically reduced.