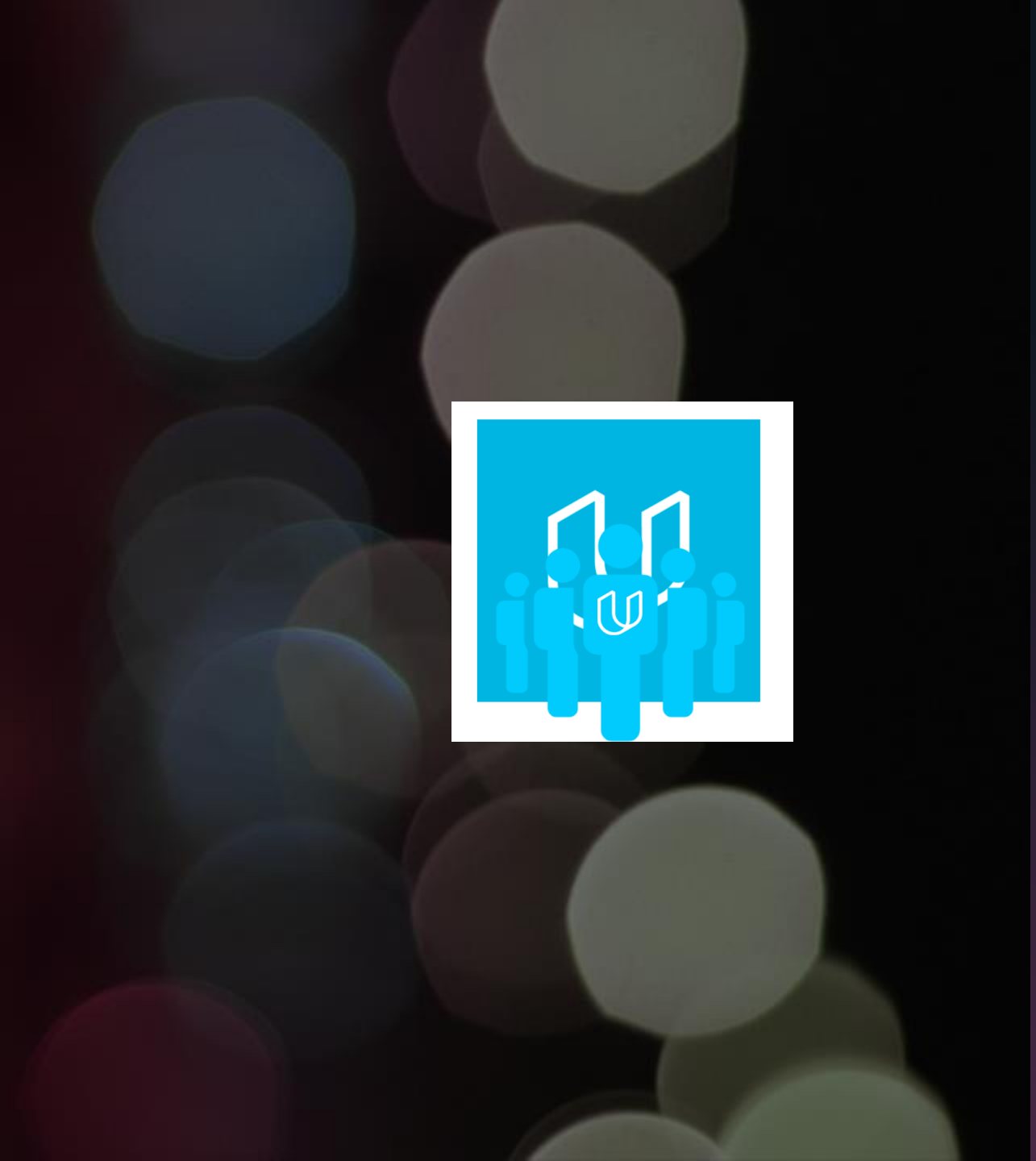


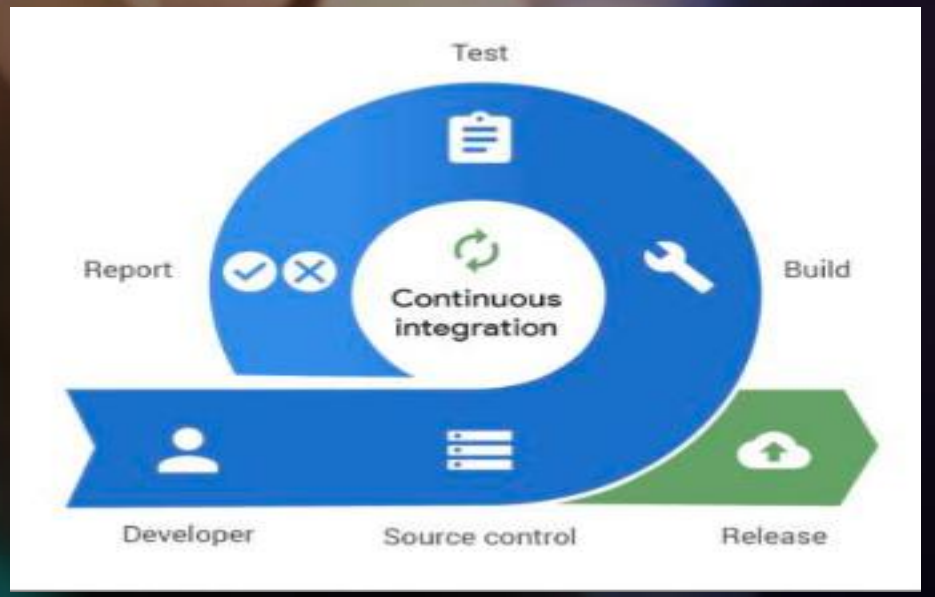
CI/CD Transform Business

Chemparidhi - UdaPeople

Continuous Integration

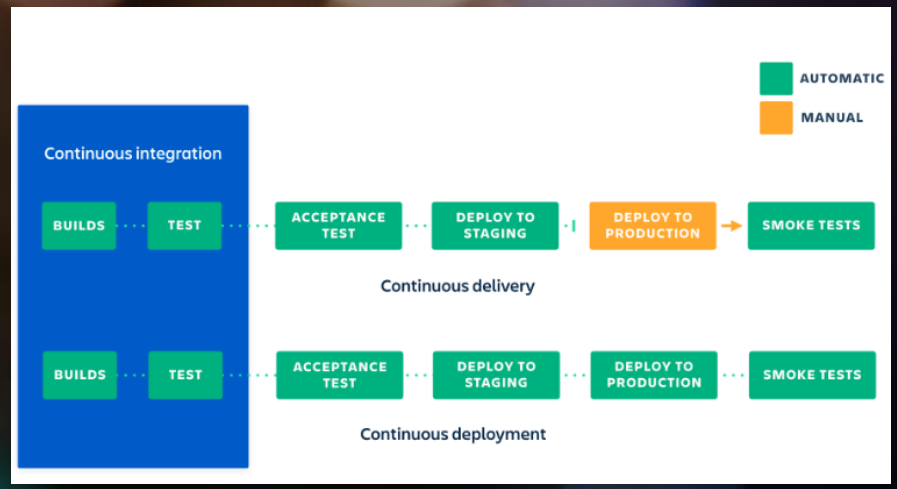
- Continuous integration is the practice of integrating all your code changes into the main branch of a shared source code repository early and often, automatically testing each change when you commit or merge them, and automatically kicking off a build. With continuous integration, errors and security issues can be identified and fixed more easily, and much earlier in the development process.

- By merging changes frequently and triggering automatic testing and validation processes, you minimize the possibility of code conflict, even with multiple developers working on the same application. A secondary advantage is that you don't have to wait long for answers and can, if necessary, fix bugs and security issues while the topic is still fresh in your mind.



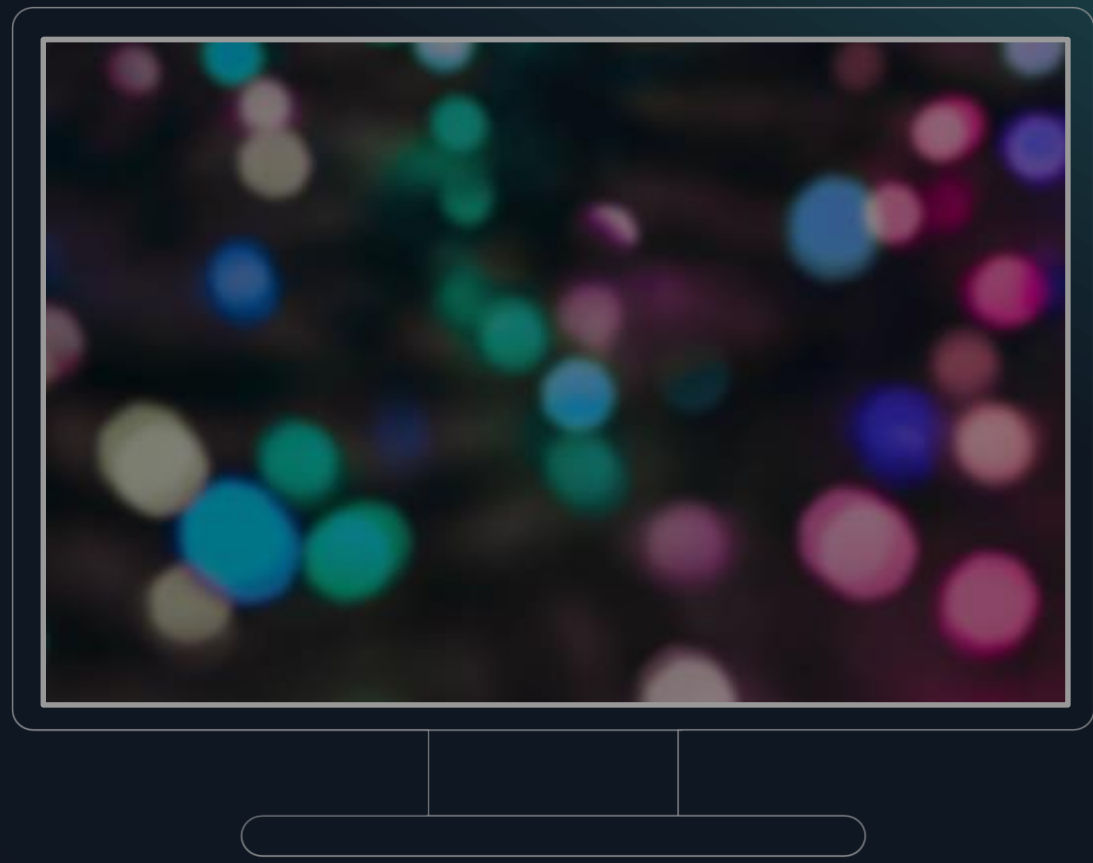
Continuous Delivery

- Continuous delivery is a software development practice that works in conjunction with CI to automate the infrastructure provisioning and application release process.
- Once code has been tested and built as part of the CI process, CD takes over during the final stages to ensure it's packaged with everything it needs to deploy to any environment at any time. CD can cover everything from provisioning the infrastructure to deploying the application to the testing or production environment.
- With CD, the software is built so that it can be deployed to production at any time. Then you can trigger the deployments manually or move to continuous deployment, where deployments are automated as well.



CI/CD Fundamentals

- A single source repository
- Frequent check-ins to main branch
- Automated builds
- Self-testing builds
- Frequent iterations
- Stable testing environments
- Maximum visibility
- Predictable deployments anytime



Benefits of CI/CD

Companies and organizations that adopt CI/CD tend to notice a lot of positive changes. Here are some of the benefits you can look forward to as you implement CI/CD:

Happier users and customers

Fewer bugs and errors make it into production, so your users and customers have a better experience. This leads to improved levels of customer satisfaction, improved customer confidence, and a better reputation for your organization

Accelerated time-to-value

When you can deploy any time, you can bring products and new features to market faster. Your development costs are lower, and a faster turnaround frees your team for other work. Customers get results faster, giving your company a competitive edge.

Free up developers' time

With more of the deployment process automated, the team has time for more rewarding projects. It's estimated that developers spend between 35% and 50% of their time testing, validating, and debugging code. By automating these processes, developers significantly improve their productivity.

Less fire fighting

Testing code more often, in smaller batches, and earlier in the development cycle can seriously cut down on fire drills. This results in a smoother development cycle and less team stress. Results are more predictable, and it's easier to find and fix bugs.

Hit dates more reliably

Removing deployment bottlenecks and making deployments predictable can remove a lot of the uncertainty around hitting key dates. Breaking work into smaller, manageable bites means it's easier to complete each stage on time and track progress. This approach gives plenty of time to monitor overall progress and determine completion dates more accurately.

Less context switching

Getting real-time feedback on their code makes it easier for developers to work on one thing at a time and minimizes their cognitive load. By working with small sections of code that are automatically tested, developers can debug code quickly while their minds are still fresh from programming. Finding bugs is easier because there's less code to review.

THANK YOU!