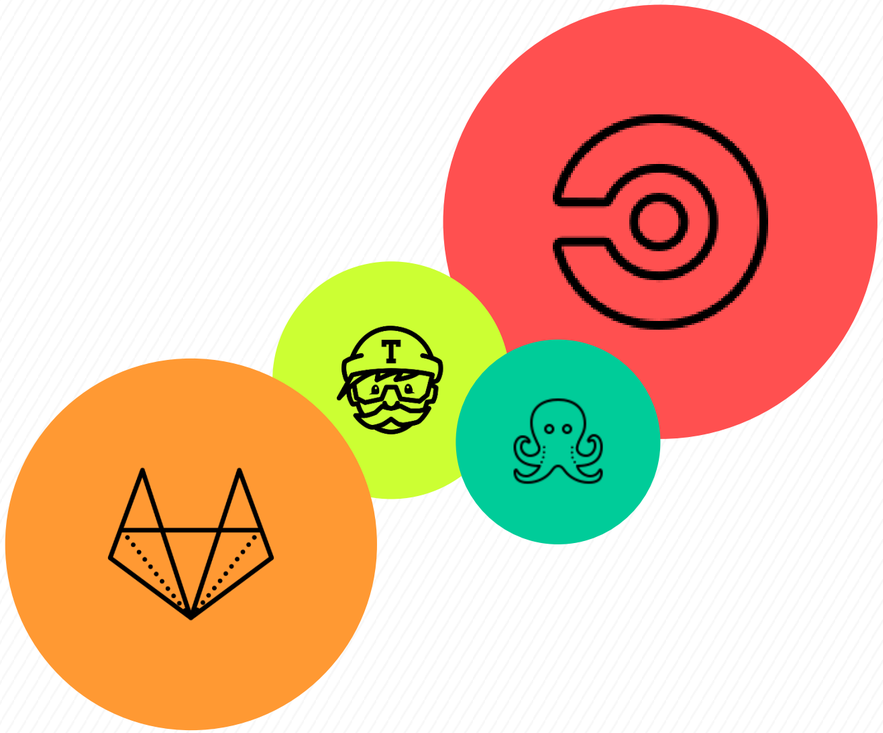
SLIDE 1

WHAT IS CI/CD



CI

Continuous integration, or CI as it's often known, is the practice of having everyone working on the same software project share their changes to the codebase regularly and then checking that the code still works as it should after each change. Continuous integration forms a key part of the DevOps approach to building and releasing software, which promotes collaboration, automation and short feedback cycles.

CD

Continuous deployment takes the DevOps practice of automating build, test and deployment steps to its logical extreme. If a change to the code successfully passes all previous stages of the pipeline, that change is automatically deployed to production without any manual intervention. Adopting continuous deployment means you can get new features to your users as fast as possible, without compromising on quality.

SLIDE 2

WHY CI/CD IS IMPORTANT

FASTER TIME TO MARKET

The primary goal of a CI/CD pipeline is to deliver working software to users quickly and frequently.

BETTER CODE QUALITY

Testing your code's behavior is an essential step in the software release process but doing it thoroughly can also be extremely time consuming.

MAXIMIZED CREATIVITY

As we've seen, building a CI/CD pipeline eliminates waste and helps create a leaner, more efficient software development and release process.

SLIDE 3

FASTER TIME TO MARKET

Reduced risk

Having a shorter time to market doesn't just help you keep up with the competition. Rapid releases provide an opportunity for product managers and marketing professionals to engage more closely with the development process.

Shorter review time

With continuous integration, developers are encouraged to commit their code changes more frequently – at least once a day as a rule of thumb. Sharing code with the rest of the team regularly not only ensures everyone is building on the same foundation, but also results in faster code reviews and makes it easier to integrate changes.

SLIDE 4

BETTER CODE QUALITY

1

Smoother path to production

As deploying changes more frequently you'll identify pain points and steps in your current process that slow you down.

2

Faster bug fixes

Each release to production will contain a relatively small number of code changes, making it much easier to identify the cause of an issue.

3

Efficient infrastructure

Once you've established a solid CI foundation, the next stage is to automate deployment of your build to test and staging environments.

4

Measurable progress

Consistently extending your code test coverage, reducing your defect rate, or increasing your release frequency all belong on the team's collective brag sheet as signs of great working culture.

5

Tighter feedback loops

It starts with automated build and test steps to inform you of immediate problems, helping you to work more efficiently and effectively than if there is a long delay between the original work and the results.

6

Collaboration and communication

DevOps is as much about building a collaborative culture as it is about new processes and tools. In order to get started with CI/CD you need to start breaking down barriers between teams and encouraging more communication.

MAXIMIZED CREATIVITY

Automated process

By using computers to perform repetitive tasks, an automated process also frees up individuals to be creative. Instead of following manual test scripts, refreshing environments, or deploying updates, you can focus on solving problems and experimenting with solutions.

More creative

Having the scope to be more creative and add value to what you do leads to improved job satisfaction, which encourages people to contribute more, attracts more talent, and improves staff retention. In turn, that benefits your organization, your product, your users, and ultimately your bottom line.