CI/CD

A brief presentation about the business impact of this practice

Israel Andrade - 16/12/2021

What is CI/CD?

What does it stand for and how it applies.

- **Continuous Integration (CI):** The practice of merging all developers working copies to a shared mainline several times a day. The goal is to speed up the release process by enabling teams to identify and fix problems early in the development lifecycle. Also, it ensures that the code follows predefined rules, (like eslint rules, for example), which increases code general quality.

- **Continuous Deployment (CD):** A software engineering approach in which the value is delivered frequently through automated deployments. Follows the practice of getting all updates, fixes, features, and configuration changes either into production or into the hands of end-users as quickly (and safely) as possible. CD aims to ensure that the code is always in a deployable state.

What are the business benefits of CI/CD?

Let’s take a look at a few benefits in the next sections

Increased Revenue

By implementing CI/CD, we can reduce the time our products take to go to market. With that, we can start to generate revenue immediately with the features those products deploy rather than waiting for the whole product to be completed, tested and checked manually.

Not only will our product be pushed to market earlier, but we will also be able to push new features and updates to that product more often. Since we have faster, safer, and more frequent production deployments, we increase our revenue with value-generating features being released more quickly.

Reduced Costs

Through CI/CD, we can catch compile errors after merge. That means if something breaks, our developers will know exactly what broke, or at least where to find it. Done right, this practice will eliminate many of the costs incurred in building and testing code changes. This also ensures that our developers are working to solve real problems and worrying less about production issues and bugs, so our organization will spend less money on tasks that don’t generate value at all.

One more way to reduce our costs is to automate our infrastructure cleanup. By doing that, we ensure that we are not getting any invoices from unused environments, resources and services.

Avoid Costs

We can also avoid costs with CI/CD. How is that? Well, since we can implement a strong pipeline with armored unit testing routines, when those fail, we have probably captured what would become a bug in production! So that makes us have less bugs in production and spend less time testing things.

And what about if we implement security checks to capture vulnerabilities? Besides protecting us against the embarrassment of having a security breach in production, it avoids our cost to have someone working on fixing it.

One more thing that we should take in mind to avoid costs is automating our infrastructure creation. That way we would have less human errors, since we only have to hit a correct infrastructure once and translate it to a script, and have faster deployments.

Protect Revenue

Lastly, we can protect our revenue by automating our smoke tests. That would reduce the downtime in any deploy related crashes or some major bug in production.

Along with that, we can have automated rollbacks triggered by job failures, that means if somethings goes wrong, we can quickly undo the changes and get back to a working state.

Technically speaking those would be some ways to protect our revenue with CI/CD. But take in mind that all those practices also increase client trust in our company, so we can protect our relationship with that client and consequently the revenue we get from it.