Continuous delivery

We discuss the main features of automating the work in a very efficient way

What is continuous delivery?

In this part we introduced the concept of Continuous delivery or CD, which is a very innovative concept in software development. We also introduced some of the topics we were going to talk about during the presentation: advantages and disadvantages of CD, tools used in CD, etc. All key terms are mentioned with few details so that we do not repeat the same topics again and again.

The information found in this introduction was taken from the following links which are exposed in this document.

Continuous delivery and DevOps

Continuous delivery is a software engineering approach in which teams produce software in short cycles, DevOps isn't a software engineering approach, it's rather an approach to culture, automation, and platform design. Continuous delivery is often used in DevOps alongside Continuous Deployment (both are often referred as the CI/CD pipeline) with development and operations teams working together in an agile way. We took most information from RedHat webpage.

Pros and cons

Here we discuss the general features that make the continuous delivery methodology, which is great for working more efficiently.

In this situation, there are more benefits as the tests are the most important priority in order to reduce costs and enhance the maintainability through the stages of the project, when in the other part we face problems that could be easily found in every project and somehow don't depend that much while developing but also in the design, or marketing strategies that don't affect the scope of this methodology that directly.

Tools

There are many tools in the market that accomplish the different concepts of continuous delivery. In this topic we will talk about some of these tools (like Buddy or Eclipse) and their characteristics. Information was taken from this website, Buddy, Eclipse and Selenium.

Differences with continuous integration

Many people are confused with the differences between continuous integration and delivery. In this topic we defined those differences with information extracted from this website. An image in which the difference of those terms can be easily seen here.

Differences with continuous deployment

In continuous delivery, every part is automated instead of the release one, which is controlled manually, when in continous deployment this stage is automated (if there are no bugs).

It could be said that in the continuous deployment methodology they are also applying continuous delivery until that releasing part, with the drawback that if all the tests don't pass, the application won't be released, so with that approach the way of testing must be a little more exhaustive.

Deployment pipeline

Here we explain the pattern used in continuous development, which helps recognizing the bottlenecks in the delivery process. The idea is to automate those parts that are repetitive, making the work more efficient. Each time one of the automated parts runs, it works like a pipeline, consuming an input and returning an output in the form of errors found or metrics.

The stages of the pipeline vary from team to team, but the most common are:

- **Commit stage:** checks the system works at a technical level.
- **Acceptance tests stage:** asserts the system meets the needs of the client.
- **Manual tests stage:** checks the system is usable and fulfils the requirements.
- **Release stage:** delivers the system to the users, or deploys it to a staging environment.

The key purposes of this pattern are:

- **Visibility**: every member of the team can see all aspects of the system.
- **Feedback:** the team can be aware of problems as soon as they occur.
- **Continuous deployment:** the software is always ready to be released to any environment.

This information was collected from informit, Wikipedia, bmc and Hari's blog.

Strategies for overcoming adoption challenges

Lianping Chen, an independent researcher and consultant who currently works as chief software engineering expert at Huawei Technologies wrote an article presenting several strategies to overcome CD adoption challenges. Here we have the paper from we took the information. The points are:

- Selling CD as a painkiller
- Dedicated team with multidisciplinary members
- Continuous delivery of continuous delivery
- Starting with easy but important applications
- Visual CD pipeline skeleton
- Expert drop