isep Instituto Superior de Engenharia do Porto
DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

Organização do Desenvolvimento de Software

MYRECIPE – TECHNICAL REPORT

EVARISTO FIGUEIREDO – 1010836

NELSON LOPES – 1160098

GROUP NO. 101 - 2016-2017

CONTEÚDO

THEORETICAL GLOBAL PLAN

ROADMAP OF THE PRODUCTS

The complete roadmap of the products is dependent of the dates that MyRecipe company will require us after the approval of the financing, after the event with the investors, at least in what concerns to the iOS application and to the Android application. However, in the next two images, it is possible to see our milestones proposal, which focus in an immediately starting point with the development of the OFBiz server entities, services and REST component, followed by the development of the Console "Mobile" Application, which must be complete at 4 January 10:00am for the presentation to the investment group at the event.

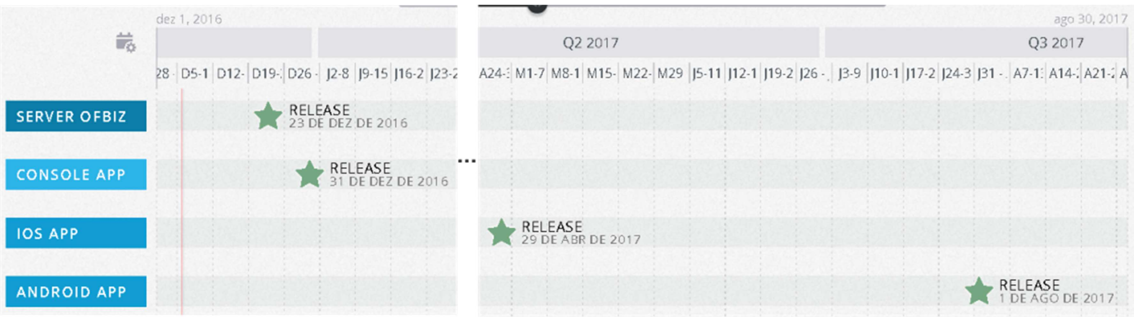


Figure 1 - The milestones for the four products

The diagram is cut at the middle, and the part that was cut means empty space. In the next image we include the complete diagram, where we lost legibility because of the small letters, but gain in understanding as it shows as the complete view.

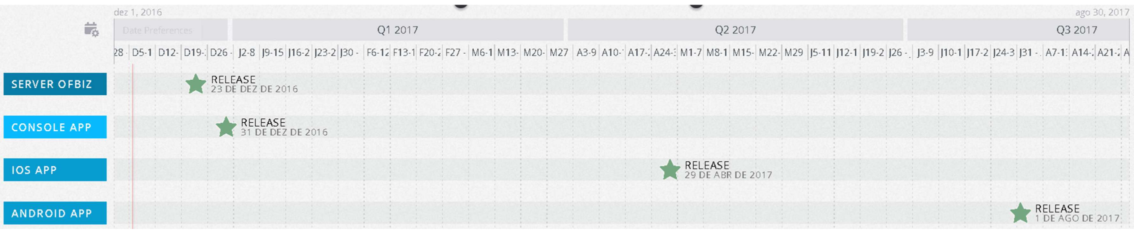


Figure 2 - Complete view of the time lapse

We defined that the iOS application has its development starting at April, the first. This date appear because of the eventual need of the MyRecipe company to complete analyze our proposal with time after the event.

For the Server OFBiz milestone to be accomplished, three parts are to be developed: the part referent to the entities, the part referent to the services and the REST component.

For the Console "Mobile" Application to be ready, two parts are to be developed, respecting to the use cases "Register a recipe" and "View a recipe".

They need to be ready for the event, so it must start immediately after this document is sent to the company and acceptance is received and must be ready at 3 January.

DESIGN OF THE PIPELINE(S)

For the design of the pipeline(s) there are some options, but two stand out from the others for the advantages they offer.

One of the solutions could be to have a pipeline with the four products, where the first one to be compiled was the OFBiz server, then the "Mobile" Application Console, then the iOS Application, and finally the Android Application, all in sequence. However, we consider that, although this solution is possible and it works, it is not the most adequate solution to our problem.

The two options that we consider the most appropriate are:

- Compile the OFBiz server and then all three other applications concurrently/in parallel;
- The Console "Mobile" Application depends on the OFBiz server and both iOS Application and Android Application depends on the Console “Mobile” Application and run in parallel – this was our choice, as we assume that it is the more adequate solution to our problem, because as the Console “Mobile” Application is like a replica of the other two mobile applications, we ensure that all the communication needed between the server and the client is working before the compilation of the real mobile applications.

In other words, the server is the first to be build, tested, and deployed. Only when everything is ok with this product, we advance to the Console “Mobile” Application and only when everything is ok too with this product, we compile the iOS and the Android app in concurrency, using the parallel feature of Jenkins.

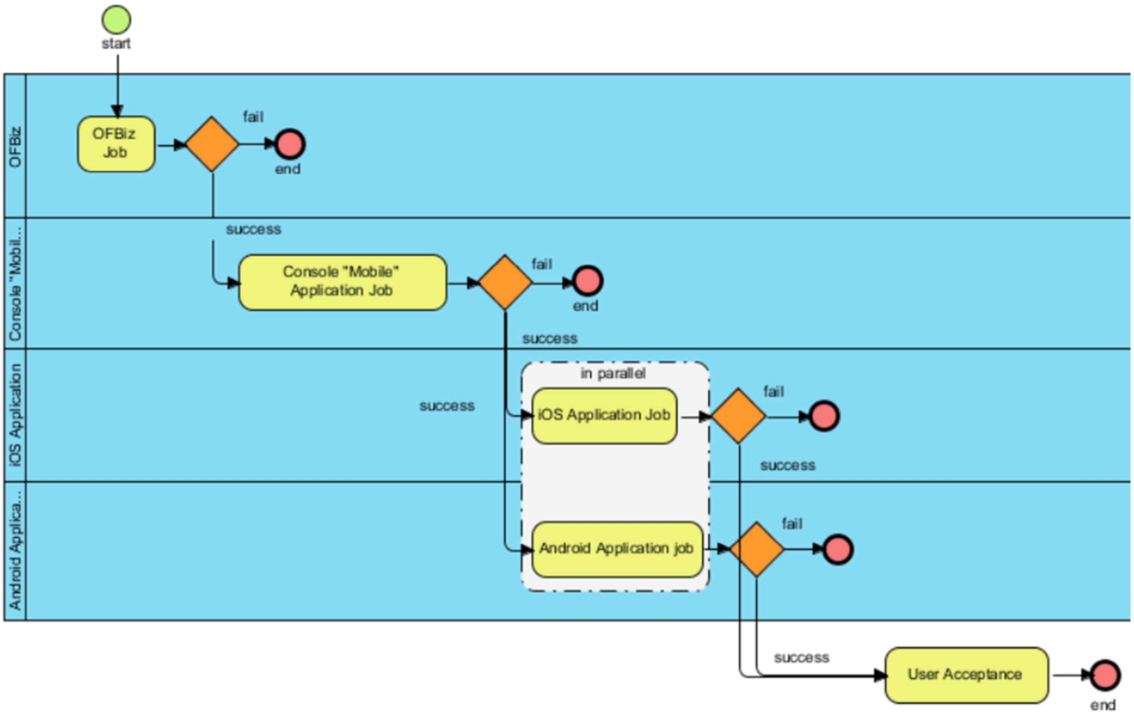


Figure 3 - Job Sequence where it is possible to see the iOS and the Android pipelines running in parallel

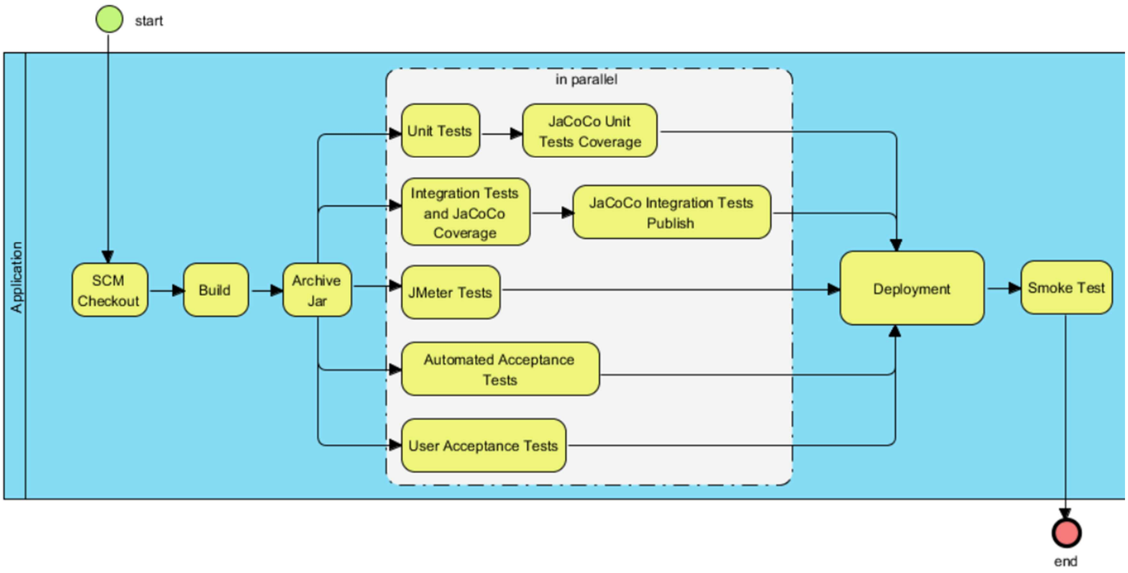

start
Application
in parallel
SCM Checkout
Build
Archive Jar
Unit Tests
JaCoCo Unit Tests Coverage
Integration Tests and JaCoCo Coverage
JaCoCo Integration Tests Publish
JMeter Tests
Automated Acceptance Tests
User Acceptance Tests
Deployment
Smoke Test
end

Figure 4 - The flow of each pipeline and product

ARCHITECTURE OF THE SOLUTION

The entire process starts with the developer, who uses git to commit each of his changes to one or more projects. With this tool, you can control the various versions that are being introduced by each of the developers. It is also possible for each developer to work on different features, and in the end it is possible to merge all additions as long as they pass the tests.

Bitbucket is the repository, where you can find not only the most current source code, but also the one of each of the versions (commits) that the developers have been introducing. In order for each of the developers changes to be viewed and integrated by other developers, it is necessary to start from git push, that is, to be sent to the repository. The developer who wants to get the most current version should then pull. Other options are available and are important to the process, but will not be addressed as this is not the scope of this document.

Por fim, é utilizado o Jenkins como servidor de Continuous Integration. No caso deste projeto, este encontra-se nos servidores do Departamento de Engenharia Informática (DEI) do Instituto Superior de Engenharia do Porto.

Although the flow is made from left to right, in the following diagram, we assume that each component to the right depends on the leftmost components.

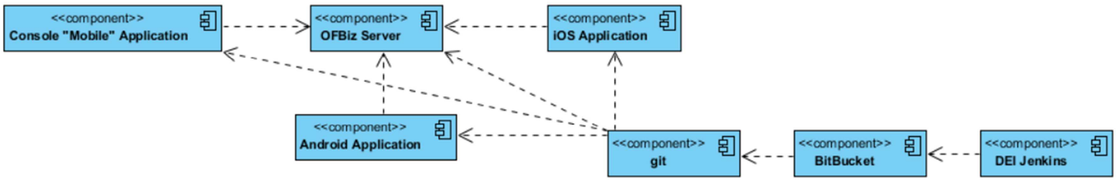


Figure 5 - Components diagram of the solution

The components of this project are the different applications, from the OFBiz server, through the "Mobile" Application Console and terminating in iOS and Android Application, all of which depend in some way on the OFBiz server, for obtaining and for persistence of data.

DEPLOYMENT DIAGRAM OF THE PIPELINE

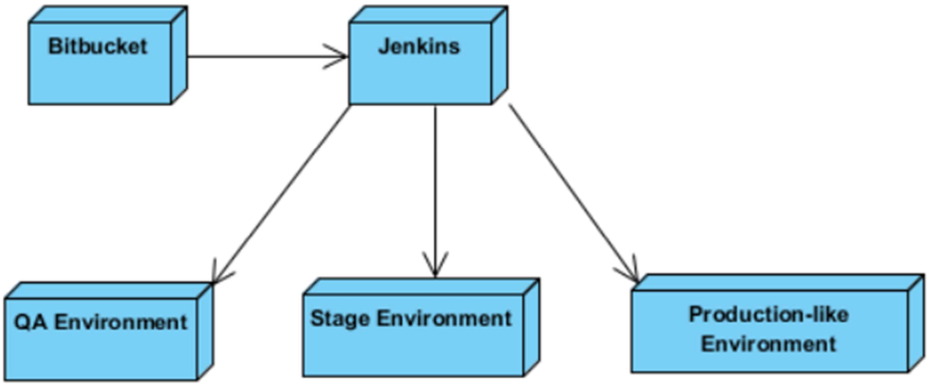
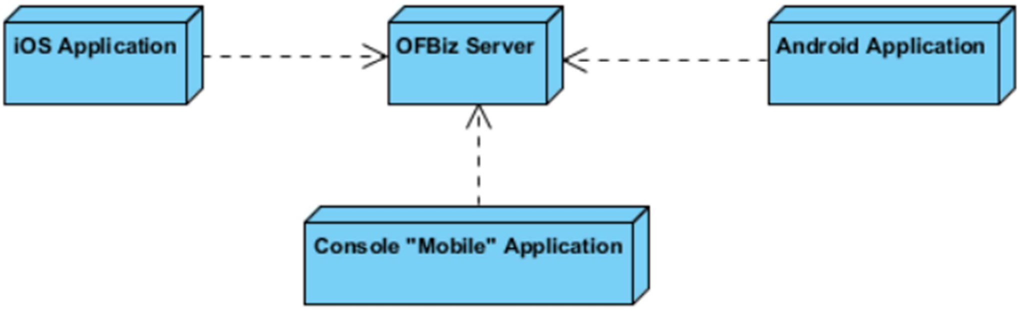
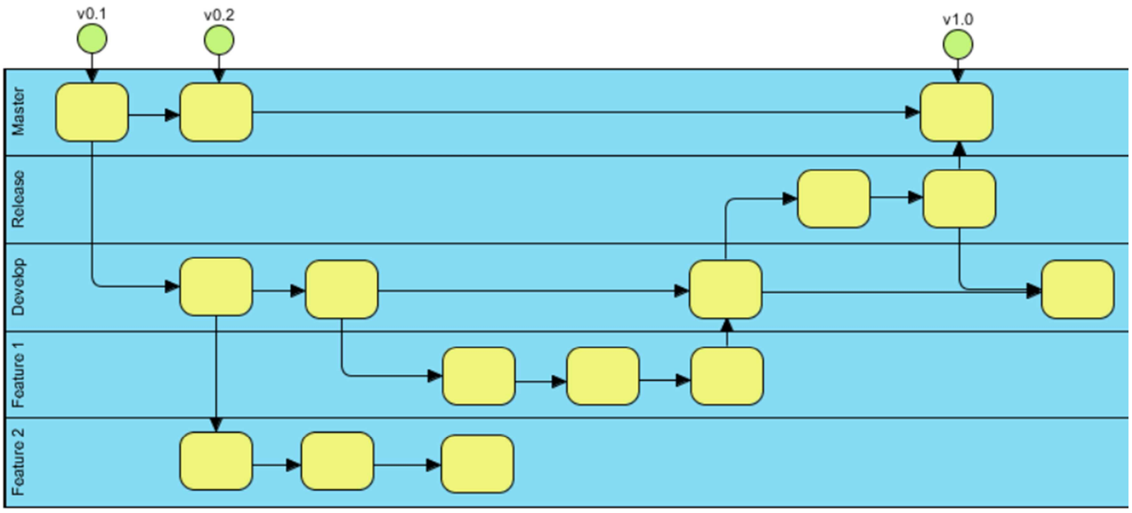


Figure 6 - Deployment diagram of the pipeline

DEPLOYMENT DIAGRAM OF THE SOLUTION



STRUCTURE OF THE REPOSITORY


v0.1
v0.2
v1.0
Master
Release
Develop
Feature 1
Feature 2