

Instituto Superior de Engenharia do Porto

Departamento de Engenharia Informática

Mestrado em Engenharia Informática



ODSOFT

Organização e Desenvolvimento do Software

Relatório Técnico

Project Plan / Roadmap

*José Adriano Ferreira
1970390@isep.ipp.pt*

*Paulo Henrique Russo
1150285@isep.ipp.pt*

**Docentes:
Alexandre Bragança
Nuno Betencourt**

**Porto
6 de Dezembro de 2015**

1. Introdução

Este relatório técnico descreve o plano de um projecto de desenvolvimento de software, seguindo a metodologia de “continuous delivery”.

O projecto consiste na implementação de um pipeline de “continuous integration” que permita o desenvolvimento de quatro componentes, da aplicação designada por “Memorix”:

- Server application baseado no OFBiz
- Web application
- IOS application
- Android Application

A aplicação Memorix é integrante do OFBiz e é uma “mobile application” que permite que os utilizadores registem notas sob a forma de texto, áudio, fotos, vídeos, etc. e as possam partilhar através de redes sociais. Os dados das notas devem ser registados nos servidores da Memorix.com, mas também devem ser acessíveis via interface web.

Resumidamente o pipeline deverá contemplar:

- A compilação do OFBiz.
- Fase de testes unitários do OFBiz, incluindo a geração de um relatório de “Cobertura”.
- Fase de testes de aceitação semi-automatizada, que implica a realização de testes de UI (user interface) e aceitação manual do utilizador de testes.
- Fase de testes UAT (User Acceptance Tests), que implica a aceitação do utilizador final. Estes devem ser realizados no ambiente do utilizador final (para o efeito será utilizada uma máquina virtual).

2. Roadmap

Pipeline – Server Application no OFBiz

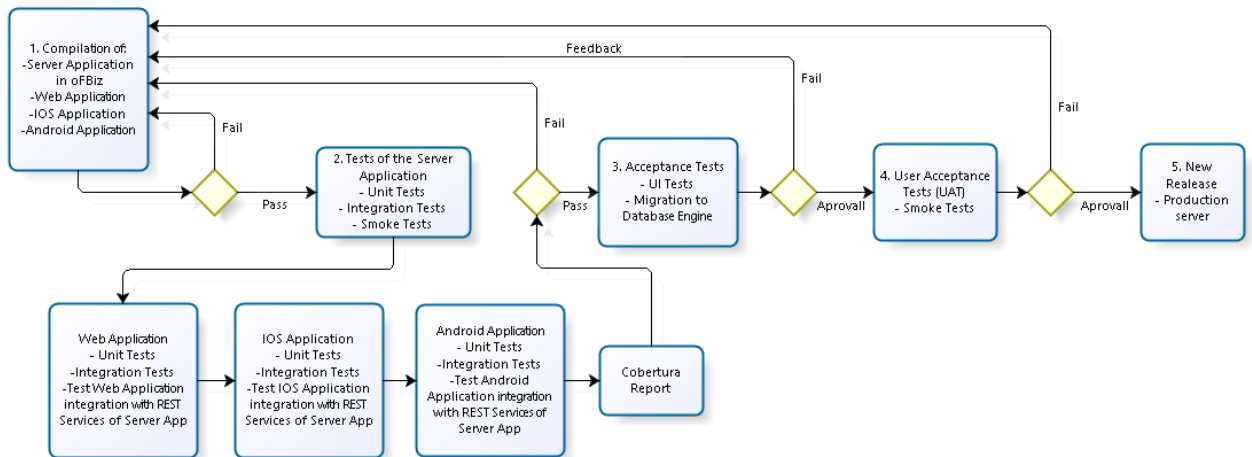


Figura 1- Diagrama do pipeline da Server Application

Repositório: **Bitbucket** - mei-isep/mei-rep-2015-g047

Linguagem: **Java** (jdk1.8.0_60)

Build tool: **Ant**

Database Engine: **MySQL**

Será necessário o recurso a:

- Continuous Integration Server (ex. Jenkins)
- IDE (ex. Eclipse)
- Control version system (ex. GIT with Sourcetree)
- VirtualizationSoftware (ex. Virtualbox)
- Framework: OFBiz

Componentes do OFBiz:

- A aplicação será desenvolvida no componente Hot-deploy do OFBiz.
- Serviços REST que suportem o acesso das restantes componentes, web e mobile, ao backend no servidor: **noteresource.java** (para suportar os casos de

uso criar e ver nota, que devem contemplar pelo menos os métodos GET, PUT e POST), ***pingresource.java***.

- Entidades: Notes

Testes de aceitação:

- Introdução de uma nova nota com o texto “teste de aceitação #teste” Durante a introdução o utilizador deverá validar que surge no ecrã a data e hora do sistema e o número de caracteres do texto introduzido que deverá estar correcto.
- Verificar se consegue encontrar a nota introduzida no passo anterior procurando-a por hash tag “#teste”. Depois verificar se o conteúdo, data e hora em que foi criada estão correctos.

Na **fase 1** é compilado o código da aplicação no OFBiz (main code e o código dos respectivos testes) e das restantes componentes (web, ios e android).

Na **fase 2** são realizados os testes unitários à aplicação, incluindo testes aos métodos de GET, PUT, POST e DELETE do serviço REST e testes de integração dos restantes componentes (web application, ios application e android application). Estes testes de integração devem servir para garantir que as restantes aplicações continuam a funcionar, apesar das eventuais alterações da estrutura da aplicação servidor (entidades, serviços, etc.). Também são realizados Smoke Tests (testar o ambiente). Só se todos os testes forem bem sucedidos é que o package produzido pela compilação é copiado para a Virtual Machine de testes (VM1). Será realizado um relatório de cobertura.

As fases 1 e 2 são automáticas, ou seja, são despoletadas por um commit dos developers. Caso falhe, é dado o respectivo feedback aos developers.

Na **fase 3**, são realizados testes de interface por um utilizador de testes (tester). Deve usar a aplicação, testando entre outros, os casos de uso: “register a text note” e “view a note”. Esta fase deve prever a migração para um database engine a definir. O avanço para a fase seguinte requer aceitação do utilizador de testes. Caso aceite, o package é copiado para a Virtual Machine de um End-user (VM2).

A **fase 4** é realizada num ambiente de um utilizador final que fará os testes de aceitação (UAT). Inclui a realização de smoke tests e a verificação de casos de uso, tal como na fase anterior. O avanço para a fase seguinte requer aceitação deste utilizador. Caso aceite, é efectuado ao deploy para a máquina de produção.

3 Pipelines adicionais:

- Web Application
- IOS Application
- Android Application

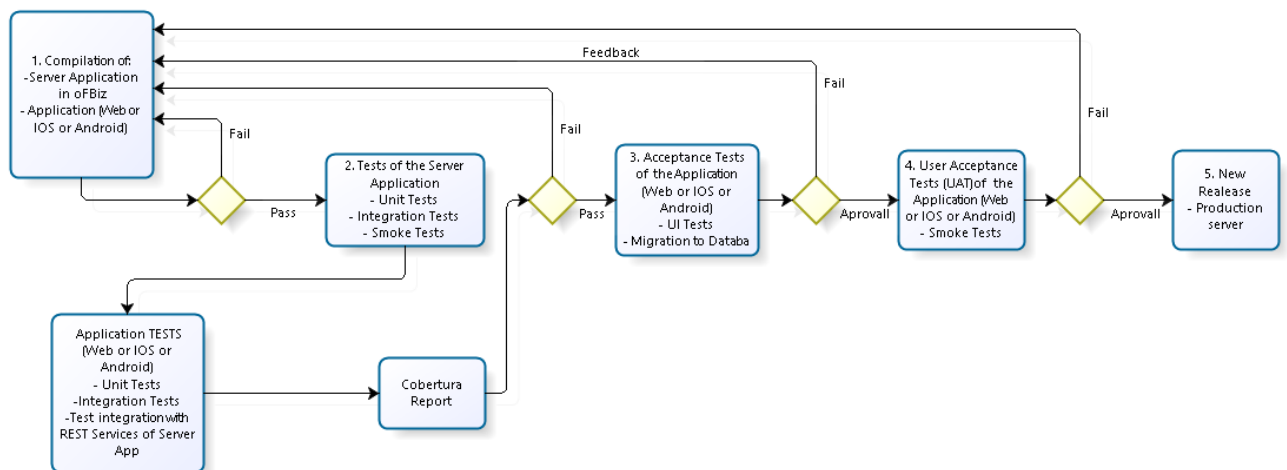


Figura 2- Diagrama conceptual dos 3 pipelines, um para cada uma das aplicações web, ios e android.

Para cada um dos 3 componentes adicionais está previsto um pipeline semelhante, em que a principal diferença reside na fase 1 e 2. Apenas apresentamos um diagrama que serve para representar cada um dos pipelines para cada aplicação (web, ios ou android).

Na **fase 1** é compilado o código da aplicação em causa e da server application.

Na **fase 2**, ao nível dos testes de integração, apenas são efectuados testes de integração entre a aplicação servidor e a aplicação em causa (web application por exemplo). Isto porque, se por exemplo não se estiver a conseguir fazer o deploy de

uma nova versão da web application, não significa que uma nova versão IOS não seja produzida e publicada. A única aplicação que tem que estar conforme e compatível com as restantes é a Server Application.

No caso da aplicação Android, uma vez que só se prevê o seu desenvolvimento mais tarde, será feito um desenvolvimento de um código básico (tipo “Hello world”), apenas para que que esteja previsto no pipeline da aplicação servidor desde já.

Diagrama de casos de uso:

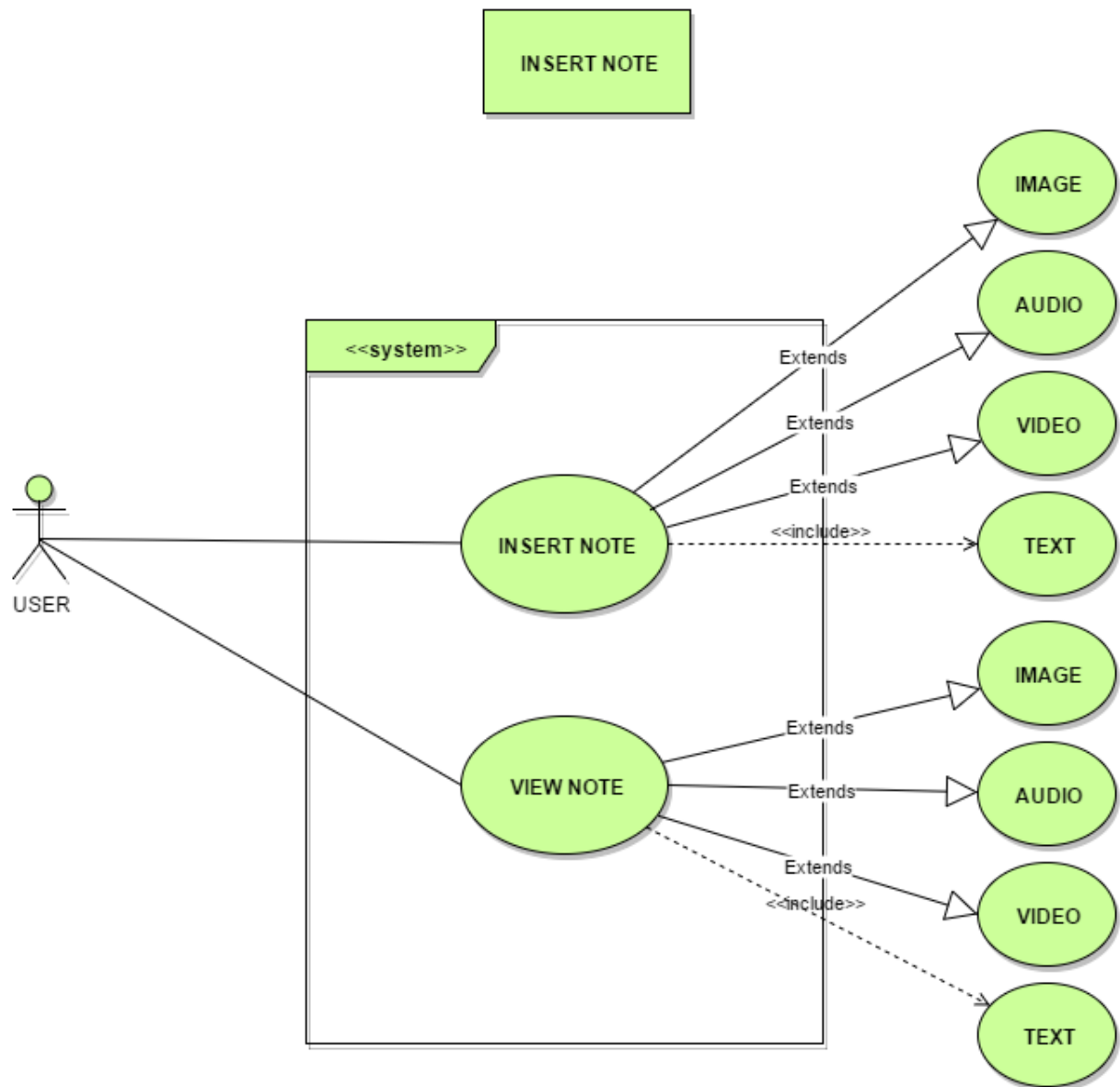


Figura 4- Diagrama de casos de uso "Insert Note"

FIM DO DOCUMENTO