

**Licenciatura em Engenharia Informática**

**Engenharia de Software II**

**SCM PLAN**

Docentes: Cristóvão Sousa

Fábio Silva

Alunos: João Marques 8170200

Pedro Pinto 8170262

Ricardo Ferreira 8170279

**Índice**

[**1 INTRODUCTION** 3](#_Toc535194733)

[1.1 PURPOSE 3](#_Toc535194734)

[1.2 SCOPE 3](#_Toc535194735)

[1.3 REFERENCES 3](#_Toc535194736)

[**2 SCM** 3](#_Toc535194737)

[2.1 INTRODUCTION 3](#_Toc535194738)

[2.2 ORGANIZATIONS & NON-CM MEMBERS 3](#_Toc535194739)

[2.2.1 SOFTWARE DEVELOPMENT TEAM MEMBERS CHANGE REQUESTS 3](#_Toc535194740)

[2.2.1 SOFTWARE DEVELOPMENT TEAM NON-MEMBERS CHANGE REQUESTS 4](#_Toc535194741)

[2.2.2 CUSTOMERS 4](#_Toc535194742)

[2.3 SCM RESPONSIBILITIES 4](#_Toc535194743)

[2.3.1 CM TEAM RESPONSIBILITIES 4](#_Toc535194744)

[2.3.1b CM LEADER 4](#_Toc535194745)

[2.3.2 CONFIGURATION CONTROL BOARD (CCB) 4](#_Toc535194746)

[2.3.2.1 CCB LEADER 4](#_Toc535194747)

[2.3.2.2 CCB MEMBERS 4](#_Toc535194748)

[2.3.2.3 CCB LIBRARIAN 5](#_Toc535194749)

[**3 CM ACTIVITIES** 5](#_Toc535194750)

[3.1 CONFIGURATION CONTROL 5](#_Toc535194751)

[3.1.1 REQUESTING CHANGES 5](#_Toc535194752)

[3.1.2 EVALUATING CHANGES 5](#_Toc535194753)

[3.1.3 APPROVING OR DISAPPROVING CHANGES 5](#_Toc535194754)

[3.1.4 IMPLEMENTING CHANGES 6](#_Toc535194755)

[3.1.4.1 CHANGE RULES 6](#_Toc535194756)

[3.2 CONFIGURATION STATUS ACCOUNTING 6](#_Toc535194757)

[**4 TOOLS AND METHODOLOGIES** 6](#_Toc535194758)

**1 INTRODUCTION**

## 1.1 PURPOSE

O presente trabalho surge no âmbito da unidade curricular de Engenharia de Software II, referente á Licenciatura em Engenharia Informática da Escola Superior de Tecnologia e Gestão.

O desenvolvimento deste trabalho tem como principal objetivo explorar os conceitos demonstrados na unidade curricular sobre metodologias de testes, metodologias de desenvolvimento ágeis, Continuous Integration (CI) e planos de Software Configuration Management (SCM).

## 1.2 SCOPE

Mesmo que um projeto de software esteja bem desenvolvido e bem gerenciado, é provável que diante da necessidade de fazer alterações possa haver problemas, isso acontece por causa da falta de gestão na configuração do software.

Para isso, utilizamos o SCM Plan que define todas as estratégias a serem percorridas no processo de controlo e de alteração.

## 1.3 REFERENCES

* MOODLE

**2 SCM**

## 2.1 INTRODUCTION

O objetivo desta seção é descrever as organizações que devem interagir com a equipe de SCM e descrever a natureza dessas interações e quais são as responsabilidades dessas interações.

## 2.2 ORGANIZATIONS & NON-CM MEMBERS

### 2.2.1 SOFTWARE DEVELOPMENT TEAM MEMBERS CHANGE REQUESTS

Os Software Development Team Members são João Pedro Ferreira Marques, Ricardo Miguel Barros Ferreira e Pedro Luís Rangel Pinto.

### 2.2.1 SOFTWARE DEVELOPMENT TEAM NON-MEMBERS CHANGE REQUESTS

Os **SOFTWARE DEVELOPMENT TEAM NON-MEMBERS,** para fazer alguma alteração ou algum problema detetado, devem por meio do **YOUTRACK** relatar á equipa de **SOFTWARE DEVELOPMENT**. Para algum membro da equipa iniciar a solução necessita da aprovação do CCB. Qualquer pessoa externa pode enviar algum pedido de alteração .

### 2.2.2 CUSTOMERS

Os clientes são os utilizadores deste Software.

## 2.3 SCM RESPONSIBILITIES

### 2.3.1 CM TEAM RESPONSIBILITIES

A CM Team é responsável por manter toda a documentação oficial do projeto, código e software subjacente durante a duração do projeto.

A CM Team é composta por todos os membros do grupo.

### 2.3.1b CM LEADER

CM Leader: João Pedro Ferreira Marques

O CM Leader desenvolve o CM Plan e dirige o CCB.

Tem como responsabilidade notificar cada membro da equipa, das suas tarefas e o prazo estabelecido para as tais.

### 2.3.2 CONFIGURATION CONTROL BOARD (CCB)

O CCB é responsável por aprovar alterações em todos os softwares e documentação descritos anteriormente durante o projeto. É da responsabilidade do CCB orientar os desenvolvedores sobre como eles estão envolvidos na aprovação das alterações propostas.

### 2.3.2.1 CCB LEADER

CCB Leader: Pedro Luis Rangel Pinto

O CCB Leader tem como função a realização e marcação das reuniões de equipa, notificando todos os elementos.

Pode sem qualquer aviso, aprovar alterações ou convocar alguma reunião para aprovar essas tais alterações.

### 2.3.2.2 CCB MEMBERS

Todos os elementos do grupo fazem parte do CCB, tendo uma função a desempenhar.

### 2.3.2.3 CCB LIBRARIAN

CCB Librarian: Ricardo Miguel Barros Ferreira

O CCB Librarian é o único com permissões para alterar o Código e os documentos do GIT, no entanto o repositório pode ser acedido por qualquer membro do grupo

**3 CM ACTIVITIES**

## 3.1 CONFIGURATION CONTROL

Para evitar um excesso de alterações, foi criado um processo, que se rege pelas seguintes etapas:

* A criação de um problema no YOUTRACK.
* lider do CCB vê e analisa o problema criado, podendo ele mesmo aprovar a inicializaçao da resolução do problema, ou marcar um reunião para a discussão do problema.
* Após a aprovação será formado um ISSUE e disponibilizado no YOUTRACK. Caso não seja aprovado poderá ser feita a alteração ao problema prosposto e voltar a realizar etapas necessarias.
* Qualquer desenvolvedor pode iniciar a resolução do ISSUE(com o consentimento do CCB Librarian), ou CCB Librarian pode atribuir a alguém.
* Verifica se ninguém está a trabalhar num problema que envolva o mesmo Código.
* Realiza o PULL do repositório.
* Desenvolve o Código.
* Testa o Código.
* Realiza o pedido de aprovação ao CCB Librarian, (em caso de não aprovação volta ao ponto 7.)
* Em caso de aprovação realiza o PUSH para o repositório.

### 3.1.1 REQUESTING CHANGES

Para solicitar alguma alteração, será necessário, através do YOUTRACK, associando o pedido ao ISSUE escolhido para a alteração.

### 3.1.2 EVALUATING CHANGES

O lider do CCB Leader se achar necessário uma reunião para a discussão da mudança pedida, terá que notificar todos os membros do CCB.

Na reunião irá ser abordado a natureza da mudança, as vantagens e desvantagens, as possiveis alterações nos processos entre outras questões.

### 3.1.3 APPROVING OR DISAPPROVING CHANGES

O estado da solicitação de alteração, será registado através do YOUTRACK, e assim quem a solicitou será notificado automaticamente. Em caso de aprovado ficará no estado de aberto, e será lançado uma ISSUE.

### 3.1.4 IMPLEMENTING CHANGES

### 3.1.4.1 CHANGE RULES

Regras para alterações:

* Cada desenvolvedor trabalhará individualmente.
* Deve ser utilizado a ferramenta GIT.
* Para alguma alteração no GIT, será preciso uma aprovação o CCB Librarian.
* Caso o problema seja simples não será necessária aprovoção no GIT, para outro grau do problema será necessaria a revisão aprovação do CCB Librarian.

## 3.2 CONFIGURATION STATUS ACCOUNTING

Todo o projeto, será guardado nos ambientes de cada membro, como também no git, e na máquina virtual(172.20.10.107) onde tem acesso todos os membros da equipa.

Na máquina virtual foi instalado o Jenkins e o git, Jenkins esse que cada membro tem um username e password, e para os professors foi criada um conta na qual para acessar terão como username:ES2 e password:passes2.

**4 TOOLS AND METHODOLOGIES**

* Github
* JDK 1.8
* Jenkins
* JetBrains IntelliJ IDEA 2018 8.4
* Microsoft Office 365
* Messenger