

**Grupo1**

**WriteAndShare**

**Configuration Management Plan**

**Versão <1.02>**

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <10/02/2020> | <1.0> | Introdução de algumas definições, acrónimos, papéis, mecanismos de identificação e mecanismos de controlo de alterações. | Daniel Castro |
| <11/02/2020> | <1.01> | Atualização da estrutura do relatório, descrição dos Sprints, e Milestones. | Pedro Silva |
| <16/02/2020> | <1.02> | Atualização das lables, da descrição das baselines do sprint 3 | Daniel Castro |

**Table of Contents**

[Revision History 2](#_Toc32323803)

[Configuration Management Plan 4](#_Toc32323804)

[Introdução 4](#_Toc32323805)

[Propósito 4](#_Toc32323806)

[Definições, acrónimos e abreviações 5](#_Toc32323807)

[Software Configuration Management 7](#_Toc32323808)

[Organização, Responsabilidades e Interfaces 7](#_Toc32323809)

[CM LEADER (Product Owner) 7](#_Toc32323810)

[Scrum Master 7](#_Toc32323811)

[Development team 7](#_Toc32323812)

[Ferramentas, ambiente e infraestrutura 7](#_Toc32323813)

[SCM Responsabilidades 8](#_Toc32323814)

[CM LEADER (Product Owner) 8](#_Toc32323815)

[Scrum master 8](#_Toc32323816)

[Development team 9](#_Toc32323817)

[The Configuration Management Program 10](#_Toc32323818)

[Configuration Identification 10](#_Toc32323819)

[Project Baselines 10](#_Toc32323820)

[Mecanismo de identificação de testes 10](#_Toc32323821)

[***Labels*** 11](#_Toc32323822)

[Configuration and Change Control 12](#_Toc32323823)

[Processamento e aprovação de solicitação de alteração 12](#_Toc32323824)

[Change Control Board (CCB) 14](#_Toc32323825)

[Configuration Status Accounting 15](#_Toc32323826)

[Project Media Storage and Release Process 15](#_Toc32323827)

[Milestones 16](#_Toc32323828)

# Configuration Management Plan

## Introdução

## Propósito

Este artefacto tem como principal foco identificar e controlar mudanças no software, certificar que as mudanças sejam implementadas corretamente e expor as mudanças para qualquer pessoa que tenha interesse.

Com isto, é pretendido reduzir o impacto das mudanças no Sistema, permitindo assim eliminar mudanças desnecessárias, controlar as necessárias o que resulta numa maior qualidade de produto.

## Definições, acrónimos e abreviações

**CCB** Configuration Control Board

**CI** Configuration Item

**CM** Configuration Management

**CSA** Configuration Status Accounting

**PDL** Project Documentation Library (Gitlab Repository)

**PSL** Project Software Library (Gitlab Repository)

**Merge** Combine 2 branches

**Branch** A separate path of development

**Master** The development branch from which release branches emerge

**Repo** The code and document repository (aka PSL/PDL)

**Repo Owner** The CCB Librarian

**Gitlab** Remote repository host

**Push** Move changes from a local repository to the remote repository

**Pull** Move changes from the remote to a local repository

**Merge Request** A Merge Request is a re-submittal of a change request by a developer once the changes that are indicated on the change request are done on a branch.

**CM Leader** The head of the configuration management team.

**Configuration Control Board** The CCB are those responsible for enforcing the following of change control procedures.

**CCB Leader**  The head of the CCB chosen by the CM Leader.

**CCB Librarian** (AKA Repo Owner) A member of the CCB who controls access to the PSL and PDL.

**Change Request** Basically, a Gitlab Issue.

**Change Request Form** A form used by developers to request changes to

the PSL or PDL. This form is accessed via the Gitlab Downloads Page. The form is copied into Gitlab tickets/issues to be filled out and submitted.

**Version Number** Unique identifier of a release.

**Configuration Identification** The process of tracking project components.

**Configuration Item** Any item associated with the project, i.e. files.

**Developer** All project members who develop or design code.

**Project Documentation Library** The official location of all project documentation. (Gitlab Repo)

**Project Software Library** The official location of all project code. (Gitlab Repo)

**Project Teams** The project teams consist of a Project Management Team, Configuration Management Team, and Quality Assurance Team.

# Software Configuration Management

## Organização, Responsabilidades e Interfaces

## CM LEADER (Product Owner)

Daniel Castro;

O líder irá supervisionar todas as atividades do SCM. Ele irá receber todas as solicitações de mudança e tomará as decisões finais sobre essas mudanças, incluindo qual engenheiro de software executará as alterações aprovadas. O mesmo também mantém um histórico de todas as solicitações enviadas, mesmo as que foram negadas.

## 

## Scrum Master

Daniel Castro, José Cunha e Pedro Silva;

## Development team

Daniel Castro, José Cunha e Pedro Silva;

## Ferramentas, ambiente e infraestrutura

O projeto será dividido em front-end e back-end e que será utilizado a framework angular e a framework asp dotnet core, respetivamente, e serão usadas as seguintes ferramentas para que possamos gerir, controlar e assegurar que todos os procedimentos estão a ser usados e aplicados corretamente:

* Angular 9 – framework de design de aplicações e plataforma de desenvolvimento para criar aplicações de página única eficientes e sofisticados;
* .Net Core – ;
* Jasmine e Karmen – uma Framework de testing para o Angular 9;
* Gitlab – repositório do código fone;
* GitKraken – ferramenta que torna as operações do Git não apenas desempenho e eficiência, mas mais compreensíveis e visuais.

## SCM Responsabilidades

A equipa do CM é responsável por manter toda a documentação oficial do projeto, código e software do projeto. A equipa do CM é composta por: Daniel Castro, José Cunha, Pedro Silva;

## CM LEADER (Product Owner)

*CM LEADER*: Daniel Castro

O *CM Leader* informará os membros da equipa sobre suas responsabilidades. Ele também executará reuniões de revisão de código para mantê-las possíveis no prazo de entrega anteriormente definido. Por último, terá a responsabilidade de aprovar/rejeitar uma alteração (caso não seja de elevada importância e seja necessário a opinião de todos).

## Scrum master

Ajuda a equipe de desenvolvimento a aprender e aplicar as habilidades do *Scrum*. O *Scrum Master* ajuda a equipa, o *product owner* e a organização com seus conhecimentos, habilidades e por qualquer meio para se tornar cada vez mais bem-sucedido. O *Scrum Master* não é o líder dos membros da equipa, nem é gerente de projeto ou representante da equipa. No entanto, o trabalho do *Scrum Master* é servir a equipa. Eles ajudam a remover barreiras ao projeto, protegem a equipa de qualquer conflito externo e ajudam a equipa a adotar práticas modernas de desenvolvimento.

## Development team

Os membros irão trabalhar no estudo de requisitos dos métodos propostos. Aqui deverão estudar as necessidades do usuário para se encontrar uma definição completa do sistema ou requisito de software. Os requisitos colhidos devem ser quantitativos, detalhados e relevantes para o projeto. Pois eles fornecerão a referência para validar o produto final.

Depois vão trabalhar no processo de código para cada método. Os responsáveis receberão os requisitos que terão de desenvolver e terão de ter cuidado com a estruturação do mesmo de modo que seja compreensível e facilite as seguintes fases de *testing* e de *code review*.

A próxima função será a execução testes para o código feito na fase anterior.

Será importante pois validará se o processo de *coding* cobriu todas as hipóteses da funcionalidade.

No *Code Review* será revisto o código recebido por parte das fases de *Coding* e de *Testing* de modo a otimizar o mesmo. Como por exemplo limpar variáveis não utilizadas minimizar linhas de código, entre outros.

# The Configuration Management Program

## Configuration Identification

Esta seção do plano identifica os itens de configuração.

Os membros da equipa analisaram todas as especificações do projeto e procederá com a divisão do software em subsistemas. Todos os subsistemas consistirão nas principais funcionalidades do programa ou interfaces.

## Project Baselines

## Para o primeiro sprint foi acordado a realização de:

* Análise e definição dos requisitos;
* Atas de Reunião Milestone 1;
* Criação dos Relatórios;
* Definição dos Use Cases;
* Definição de User Stories;
* Para o segundo sprint ficou acordado:
* Atas de Reunião Milestone 2;
* Criar Mockups;
* Criar Script Gitlab CI;
* Criar SCM Plan;
* Formação .Net Core;
* Atualização Requisitos;
* Atualizar Relatórios;
* Para o terceiro sprint ficou acordado:
* Atas de Reunião Milestone 3;
* Atualizar Plano SCM;
* Atualizar Relatórios;
* Implementar funcionalidades nos sprints anteriores;

## Mecanismo de identificação de testes

O mecanismo de identificação dos testes deverá ser o seguinte: metodoATestar\_TC\_XX, “XX” deverá ser substituido pelo número do caso de teste e “metodoATestar” deverá ser subsituido pelo método que está a ser testado.

***Labels***

Existem dois tipos de categorias para as labels as que tem prioridade e as outras labels.

As com prioridade, de ordem descrescente, serão as seguintes: Hotfix, Change Request, High, Medium, Low, L, M, S e XS, em que para definir as prioridades devem ser usadas as labels: High, Medium, Low e para estimar as labels: L, M, S, XS.

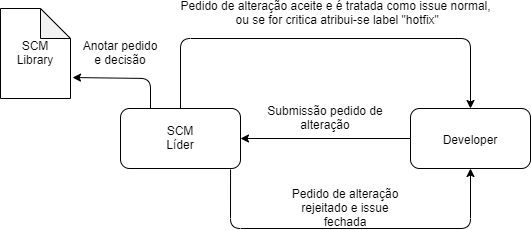
As outras labels serão as seguintes:

* Sprint 1 – Backlog
* Sprint 2 – Backlog
* Sprint 3 – Backlog
* To Do
* Doing
* Task
* User Story
* Epic

## Configuration and Change Control

### Processamento e aprovação de solicitação de alteração

Quando os developers quiserem requisitar uma alteração no código ou na documentação, têm de submeter um pedido para o SCM líder. Depois o SCM líder terá de analisar o pedido e quando tomar a decisão terá de reportar ao developer e anotar o pedido e a decisão na SCM Library. Caso seja necessário o SCM líder poderá convocar uma reunião se achar que a decisão do pedido de mudança for importante e quiser a opinião da equipa toda.



**Workproducts e documentação:**

Submissão pedido de alteração:

1. SCM Library é atualizada com o novo pedido;
2. Issue é criada no gitlab, com label “change request”;
3. Se o pedido de alteração for critico dá-se a label de “hotfix”;
4. Pedido de alteração rejeitado:
5. SCM Library é atualizada com a decisão;
6. Issue é fechada no gitlab;
7. Pedido de alteração aprovado:
8. SCM Library é atualizada com a decisão;
9. A issue fica no backlog até ser atribuída a um sprint, depois será tratada como uma issue normal;

## Change Control Board (CCB)

O Líder do CCB decidirá se a mudança é significativa o suficiente para convocar uma reunião do CCB ou para aprovar a solicitação de mudança imediatamente, dependendo da complexidade da mudança.

Na reunião e abordará essas questões de mudança:

1. Natureza da mudança;
2. Razão da mudança;
3. Impacto da mudança em outros componentes do projeto Alteração de Software;
4. Impacto nos arquivos de suporte (por exemplo,arquivos de configuração);
5. Impacto em outras funções;
6. Impacto da interface;
7. Redundância;
8. Valor adicionado;
9. Impacto em outros processos;

Em caso de reunião a mudança é aprovada caso na primeiro votação haja 3 votos positivos, ou se não houver consenso a pessoa contra a maioria tem oportunidade de defender a tese, depois disto irá ocorrer uma segunda votação ganhando a maioria.

## Configuration Status Accounting

Será controlado quem possui ramificações ativas ou que implementou mudanças no passado pelo sistema Gitlab. A CSA se aplicará apenas ao PSL / Master Branch.

## Project Media Storage and Release Process

O projeto será guardado no gitlab e como segunda via será guardado no computador de cada membro da equipa.

## Milestones

O projeto será dividido em três milestones diferentes, na primeira milestone será realizada toda a base e estruturação do projeto a nível de documentação, na segunda milestone haverá uma revisão e correção da primeira milestone e configuração do continuous integration, e inicialização nas novas tecnologias com a implementação de protótipos.