**Software Conﬁguration Management Plan**

**Sommario**

[**1 Introduction to the plan** 2](#_Toc461614026)

[**1.1** **Purpose of the plan** 2](#_Toc461614027)

[**1.2 Scope of the plan** 2](#_Toc461614028)

[**1.3 Key terms** 3](#_Toc461614029)

[**1.4 References** 3](#_Toc461614030)

[**2. CM responsabilities and authorities** 3](#_Toc461614031)

[**2.1 SCM Organizational Role** 3](#_Toc461614032)

[**3. CM Activities** 3](#_Toc461614033)

[**3.1 Configuration Identification** 3](#_Toc461614034)

[**3.2 Configuration Control** 3](#_Toc461614035)

[**3.2.1 Branch Management** 4](#_Toc461614036)

[**3.2.2 Promotion Management** 4](#_Toc461614037)

[**3.3 CM Configuration Audits and Reviews** 4](#_Toc461614038)

[**3.4 Applicable policies, directives, and procedures** 4](#_Toc461614039)

[**4. Planned activities, schedule and resources** 5](#_Toc461614040)

[**4.1 CM Schedule** 5](#_Toc461614041)

[**4.2 CM Resources** 5](#_Toc461614042)

**1 Introduction to the plan**

## **Purpose of the plan**

Il presente Software Configuration Management Plan, contiene la pianificazione delle attività di configuration management. Questo documento andrà a descrivere quello che sarà il processo del SCM, il quale nasce per tener traccia dei cambiamenti sui file, per sapere quali sono le ultime versioni funzionanti dei file stessi, e per avere la possibilità di ripristinare stati precedenti del sistema in caso di cambiamenti. L’altro aspetto importante, fornito dai tool utilizzati nel SCM, è la possibilità di lavorare in maniera cooperativa, in modo organizzato ed efficiente. Con questa organizzazione è possibile avere a disposizione sempre lo stato di avanzamento del progetto, dei vari artefatti, e del lavoro dei vari componenti. Le attività riportate in questo documento, riguardano in particolare:

1. Identificazione dei Configuration Item, ovvero degli artefatti da mettere sotto controllo di

configurazione, e per i quali gestire le revisioni, la storia, ecc.

1. Definizione di regole per il Promotion Management, quindi come viene gestita l’attività di
2. Pubblicazione di una parte del prodotto software, verso gli altri sviluppatori
3. Definizione di regole per il Release Management: quindi come viene gestita l’attività di
4. Pubblicazione di una release del prodotto verso l’esterno (i clienti)
5. Definizione di regole per il Change Management: regolamentazione della gestione di
6. Richieste di cambiamento che possono riguardare bug report o aggiunta di funzionalità
7. Individuazione dei tool da utilizzare

Il presente documento va anche ad identificare quelle che sono le responsabilità all’interno del progetto, e in particolare identifica le seguenti figure:

1. Configuration Manager: decide quali sono i Configuration Item
2. Change Control Board Member: gestisce le Change Request
3. Developer: utilizzatori dei vari tool e dei meccanismi di auditing

## **1.2 Scope of the plan**

Tale documento si basa su una serie di assunzioni che si cercherà di rispettare ma che probabilmente subiranno delle variazioni a causa delle molte instabilità organizzative.

Assunzioni:

1. Termine del progetto previsto entro Ottobre 2016
2. Costante aggiornamento della documentazione in base allo sviluppo
3. Sviluppo delle componenti architetturali del sistema (client e server)
4. Rispetto delle milestone fissate
5. Rispetto degli sprint fissati
6. Costanti scrum settimanali a sostegno dello sviluppo agile

Limitazioni:

1. Differenti disponibilità temporali dei componenti del gruppo

## **1.3 Key terms**

* Percorso: Un utente una volta effettuato l’accesso dovrà definire la strada che dovrà percorrere comunicando da che punto parte a che punto vorrà arrivare.
* Utente: Utilizzatore dell’applicazione che dovrà registrarsi per effettuare l’accesso e dovrà inserire un punto di accesso e di arrivo e selezionare un percorso tra quelli possibile
* Stato manto stradale: Tabella che conterrà il percorso effettuato e il numero di fossi della strada.

## **1.4 References**

* Standard IEEE 828-2012 : standard per la stesura del SCM plan
* User stories: raccolta di casi reali di utilizzo dell’applicazione
* Problem statement: descrizione del problema che si va a risolvere
* Documentazione di design in UML
* Presentazione primo milestone: descrizione delle user stories implementate e delle tecnologie utilizzate
* Mockup: prototipo del client da realizzare

# **2. CM responsabilities and authorities**

## **2.1 SCM Organizational Role**

Il SCM team è costituito da due sviluppatori, i quali coopereranno allo sviluppo applicativo del client e del server. Inoltre, al fine di soddisfare i criteri di qualità del codice e del software i due membri collaboreranno anche allo sviluppo e alle attività di gestione del sistema.

# **3. CM Activities**

## **3.1 Configuration Identification**

Il Configuration Manager ha suggerito di prendere in considerazione come elementi da sottoporre a Configuration Management:

1) I file contenenti codice sorgente con estensione

-".java", ".xml", (rappresentativi sia del backend, sia del client android)

2) Le librerie con estensione

- ".jar"

3) I file di configurazione con estensione

- ".gradle" (relativi all'ambiente di building dell'IDE di sviluppo)

- ".properties", ".ec" (appartenenti al tool jacoco)

4) I file relativi alla documentazione, con estensione

- ".docx”

(In fase di aggiornamento)

## **3.2 Configuration Control**

Nella seguente sezione si riporta il processo di sottomissione di una change Request e si illustra le politiche di promozione e di gestione dei branch adottate.

Per poter inoltrare una richiesta di cambiamento bisognerà riportare le seguenti informazioni da associare alla issue:

* Titolo: Contiene una breve sintesi dello scopo o della motivazione che ha reso necessario/a la realizzazione della richiesta
* Descrizione: contenente informazioni dettagliate della richiesta
* Milestone: Oltre alla richiesta si specificherà l’obiettivo che si vorrà raggiungere.
* Peso: Da indicare con un valore da 1 a 10 in base all'importanza del contenuto della richiesta.
* Etichetta: un tag specifico che contraddistingue la natura della richiesta

Di seguito si riporta la tipologia di etichette assegnabili per ogni richiesta:

* BUG: usata per segnalare la presenza di un bug
* FEATURE: usata per la promozione e l'introduzione di una nuova funzionalità
* DOCUMENTATION: impiegata per richiamare l'attenzione su questioni riguardanti la documentazione.
* ENHANCEMENT: indica che la richiesta di cambiamento investe gli artefatti relativi alla documentazione del sistema
* SUGGESTION: specifica che la natura della richiesta è volto ad aprire un dibattito su eventuali scelte progettuali intraprendibili, declinabili o attuabili.
* CRITICAL: tale etichetta è impiegata esclusivamente per indicare failure presentatesi durante l'attività di testing che causano il crash del sistema o compromettono l'adempimento delle funzionalità principali.

Qualsiasi tipologia di richiesta di cambiamento dovrà essere sottomessa all'issue tracker. La sottomissione di una richiesta di cambiamento verrà tracciata tramite un'issue tracking offerto dal sistema Gitlab. Il change control board member ha la possibilità di disapprovare la richiesta, a patto di motivare adeguatamente le motivazioni del rifiuto prima della chiusura. L'approvazione di una richiesta richiederà riscontro positivo di tutti i membri.

**(In quest’area sarà inserito un diagramma delle attività allo scopo di far comprendere meglio come avviene la sottomissione di una richiesta)**

### **3.2.1 Branch Management**

L'approvazione di una change request richiederà l'aggiunta di una nuova funzionalità sperimentale o il miglioramento di una determinata componente ciò comporta l'apertura di un nuovo branch di sviluppo, questo ci consentirà di preservare la stabilità del sistema.

### **3.2.2 Promotion Management**

Al fine di evitare rallentamenti nello sviluppo e assicurare un certo livello di qualità e stabilità sistema, lo sviluppatore dovrà effettuare un push in remoto solo dopo essersi appurato che il codice sia compilabile.

## **3.3 CM Configuration Audits and Reviews**

Ispezione della tracciabilità.

Periodicamente, a valle della implementazione e realizzazione di determinate funzionalità associate alle user stories, si verificherà:

* L'allineamento tra la documentazione e gli artefatti software del sistema;
* Il sussistere della tracciabilità tra i requisiti e gli artefatti di testing correlati alle specifiche di progetto.

Gli incontri del SCM Team hanno cadenza settimanale e avvengono con la metodologia SCRUM.

## **3.4 Applicable policies, directives, and procedures**

In seguito alla realizzazione di una determinata componente, sarà realizzata un’attività di test di unità minimale, al fine di convalidare un minimo livello di confidenza raggiunto con l'artefatto da integrare all'interno del sistema. Ogni riunione sarà rigorosamente concentrata in un quarto d'ora, al fine di determinare velocemente il punto della situazione e prefiggere i nuovi obiettivi a breve termine da realizzare. Saranno indette riunioni straordinarie nel caso in cui sia necessaria la revisione di artefatti critici; Per ogni riunione straordinaria sarà riportata una descrizione degli interventi effettuati e delle decisioni intraprese, oltre alla loro catalogazione nell'apposito issue tracker.

# **4. Planned activities, schedule and resources**

## **4.1 CM Schedule**

In tale sezione si riporta una roadmap delle milestone implementate e prefissate:

**Milestone 0.1** Implementazione delle user stories:

**Sprint 1:** Come sistema, voglio realizzare un’applicazione che mi consenta di registrarmi, visualizzare un elenco di percorsi, loggarmi e inserire un percorso da effettuare. Potrò inoltre decidere di salvare il percorso da me effettuato dando un voto al manto stradale.

Data di completamento: 17/09/16

**Milestone 0.2**

**Sprint 2:**

(Tale fase sarà aggiornata man mano che si procede con l’implementazione

## **4.2 CM Resources**

Personnel:

- Configuration Manager: Fabrizio Giorgione

- Change Control Board Member: Alfredo Nazzaro

- Developers: Giorgione Fabrizio, Alfredo Nazzaro

Software:

* **Android Studio:** ambiente di sviluppo integrato (IDE) per la piattaforma Android, basato

sul software della JetBrains IntelliJ IDEA.

* **Git:** tool di versioning distribuito, in cui ciascuno sviluppatore ha un proprio repository

locale, in cui effettua il cloning del repository centrale.

* **Git Lab:** piattaforma per lo sviluppo collaborativo di software. Prevede la possibilità di

scegliere tra una repository privata o pubblica, un sistema di controllo versione, possibilità

di gestire branch, isssue tracker, change management, statistiche di utilizzo.

* **JUnit:** frame work per l’organizzazione e l’esecuzione dei casi di test di unità, e rappresenta

una istanza della famiglia XUnit, che racchiude una serie di tool in vari linguaggi, volti a

seguire una attività di testing guidata dal codice.

* **Telegram:** tool per il tracciamento delle comunicazioni verbali a supporto delle

collaborazioni.

(Successivamente saranno aggiunti ulteriori software man mano che si procederà con l’implementazione)