**Software Conﬁguration Management Plan**

**Sommario**

[1 Introduction to the plan 2](#_Toc498082817)

[1.1 Purpose of the plan 2](#_Toc498082818)

[1.2 Scope of the plan 2](#_Toc498082819)

[1.3 Key terms 3](#_Toc498082820)

[1.4 References 3](#_Toc498082821)

[2. CM responsabilities and authorities 3](#_Toc498082822)

[2.1 SCM Organizational Role 3](#_Toc498082823)

[3. CM Activities 3](#_Toc498082824)

[3.1 Configuration Identification 3](#_Toc498082825)

[3.2 Configuration Control 3](#_Toc498082826)

[3.2.1 Branch Management 4](#_Toc498082827)

[3.2.2 Promotion Management 4](#_Toc498082828)

[3.3 CM Configuration Audits and Reviews 4](#_Toc498082829)

[3.4 Applicable policies, directives, and procedures 4](#_Toc498082830)

[4. Planned activities, schedule and resources 5](#_Toc498082831)

[4.1 CM Schedule 5](#_Toc498082832)

[4.2 CM Resources 5](#_Toc498082833)

1 Introduction to the plan

1.1 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

pubblicazione di una release del prodotto verso l’esterno (i clienti)

1. Definizione di regole per il Change Management: regolamentazione della gestione di

richieste di cambiamento che possono riguardare bug report o aggiunta di funzionalità

1. 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 fine dicembre 2017
2. Costante aggiornamento della documentazione in base allo sviluppo
3. Rispetto delle milestone fissate
4. Rispetto degli sprint fissati
5. Costanti scrum settimanali a sostegno dello sviluppo agile

Limitazioni:

1. Differenti disponibilità temporali dei componenti del gruppo

1.3 Key terms

* Ciclista: utente iscritto all’applicazione
* Sfida: competizione sportiva tra ciclisti

1.4 References

* Standard IEEE 828-2012: standard per la stesura del SCM plan
* Documento di analisi contenente: problem statement, analisi dei requisiti, use case,

design uml e testing effettuati

2. CM responsabilities and authorities

2.1 SCM Organizational Role

Il SCM team è costituito da tre sviluppatori, i quali coopereranno allo sviluppo applicativo dell’applicazione mobile. Inoltre, al fine di soddisfare i criteri di qualità del codice e del software i tre 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 del sistema Android)

- “.php” (lato server)

2) Le librerie con estensione

- “.jar”

3) I file di configurazione con estensione

- “. gradle” (file di build)

- “. properties”, “.ec” (appartenenti al plugin *Jacoco*)

4) I file relativi alla documentazione, con estensione

- ".docx”

- “.md” (per tenere traccia dei cambiamenti all’interno di file *.docx*)

3.2 Configuration Control

Nella seguente sezione si riporta il processo di sottomissione di una c*hange 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.
* 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
* **QUESTION:** specifica che la natura della richiesta è volta ad aprire un dibattito su eventuali scelte progettuali intraprendibili, declinabili o attuabili.
* **INVALID:** 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 Git. 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.

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 del sistema, lo sviluppatore dovrà effettuare un *push* in remoto solo dopo aver 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 agli use case, 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.

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 degli *use case*:

**Sprint 1:** Come sistema, si vuole realizzare un’applicazione che consenta di registrarsi, loggarsi e visualizzare la propria posizione sulla mappa. Tutto questo con l’ausilio di script in linguaggio PHP che permettono la creazione e successiva gestione del database, in cui si renderanno persistenti informazioni relative agli utenti.

Data di completamento: 03/11/2017

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

**Milestone 0.2**

**Sprint 2.** Come sistema, si vuole introdurre una nuova funzionalità: iniziare una nuova sessione d’allenamento. Si implementeranno opportuni script PHP che consentiranno la gestione della persistenza delle posizioni sulla mappa degli utenti loggati.

Data di completamento: 06/11/2017

**Sprint 3.** Introduzione dell’opzione per visualizzare i risultati degli allenamenti portati a termine da un utente.

Data di completamento: 07/11/2017

**Milestone 0.3**

**Sprint 4.** Come sistema, si vuole introdurre una nuova funzionalità: il ciclista può lanciare una sfida, di durata limitata a 60 o 90 minuti, ad un altro ciclista loggato; il ciclista sfidato può accettare o meno la sfida.

Data di completamento: 14/11/2017

**Milestone 0.4**

**Sprint 5.** Come sistema, si vuole introdurre una nuova funzionalità: il ciclista può portare a termine una sfida e riceverne il risultato. Inoltre si vogliono apportare migliorie grafiche.

Data di completamento: 17/11/2017

4.2 CM Resources

Personnel:

- Configuration Manager: Sara Caruso, Michele Fredella, Marianna Fucci

- Change Control Board Member: Sara Caruso, Michele Fredella, Marianna Fucci

- Developers: Sara Caruso, Michele Fredella, Marianna Fucci

Software:

* **Git:** tool di versioning distribuito, in cui ciascuno sviluppatore ha un proprio repository
* locale, in cui effettua il cloning del repository centrale. Possibilità di gestire brach, issue, change management e statistiche di utilizzo
* **Android Studio:** ambiente di sviluppo integrato (IDE) per la piattaforma Android, basato

sul software della JetBrains IntelliJ IDEA.

* **JUnit:** framework 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.

* **ActivityInstrumentationTestCase2:** Classe android che consente di effettuare il test di tutte le attività di sistema implementate
* **Espresso:** testing framework utilizzato per realizzare test per la UI di applicazioni Android
* **UIAnimator:** testing framework utilizzato per realizzare test per la UI di applicazioni Android
* **Altervista**: servizio di hosting gratuito utilizzato per creare un database MySQL, gestito tramite script PHP
* **Travis CI:** servizio di continuous integration utilizzato per effettuare build e testing del progetto Android ospitato in GitHub.