Laurea Magistrale in informatica-Università di Salerno
Corso di *Gestione dei Progetti Software*- Prof.ssa F.Ferrucci

FITDIARY

DECIDE, COMMIT, SUCCEED

Configuration Management Plan

Riferimento	2021_CMP_C07_FitDiary_Fasano-Spinelli_V1.0
Versione	1.0
Data	20/01/2022
Destinatario	Prof.ssa Filomena Ferrucci
Presentato da	Salvatore Fasano, Gianluca Spinelli
Approvato da	

Revision History

Data	Versione	Descrizione	Autori
18/12/2021	0.1	Prima stesura	Salvatore Fasano, Gianluca Spinelli
20/01/2022	1.0	Revisione e completamento	Salvatore Fasano, Gianluca Spinelli

Team Members

Ruolo	Nome e Cognome	Acronimo	Email
PM	Salvatore Fasano	SF	s.fasano10@studenti.unisa.it
PM	Gianluca Spinelli	GS	g.spinelli18@studenti.unisa.it
TM	Daniele De Marco	DM	d.demarco11@studenti.unisa.it
TM	Ilaria De Sio	IS	i.desio7@studenti.unisa.it
TM	Rebecca Di Matteo	RM	r.dimatteo10@studenti.unisa.it
TM	Daniele Giaquinto	DG	d.giaquinto2@studenti.unisa.it
TM	Davide La Gamba	DL	d.lagamba@studenti.unisa.it
TM	Leonardo Monaco	LM	l.monaco11@studenti.unisa.it
TM	Simone Spera	SS	s.spera7@studenti.unisa.it
TM	Antonio Trapanese	AT	a.trapanese8@studenti.unisa.it

Sommario

1. Introduzione

1.1 Ambito

L'azienda specializzata nel settore del fitness intende consentire ai professionisti del campo, di supportare al meglio i propri clienti con lo scopo di raggiungere in modo agevole i risultati prefissati. L'azienda, quindi, vuole introdurre un cambiamento all'interno del settore, in modo da eliminare l'utilizzo di tutti i canali inadeguati attualmente utilizzati, quali WhatsApp, Instagram, Telegram ecc. . Data l'esistenza di tale necessità e riscontrata una mancanza di sistemi capaci di offrire soluzioni al problema, l'azienda ritiene interessante la possibilità di offrire una piattaforma in grado di supportare il lavoro di tali professionisti. L'obiettivo del progetto è fornire uno strumento di supporto alle attività di comunicazione e gestione dei clienti da parte di personal trainers e nutrizionisti garantendo una esperienza migliore ed una maggior probabilità di raggiungimento degli obiettivi prefissati. Il sistema proposto dovrà prevedere:

- tutte attività di creazione e gestione di un nuovo cliente;
- il processo di creazione dei protocolli creati dai professionisti;
- l'attività di consultazione dei contenuti da parte del cliente;
- la possibilità, per ogni cliente, di comunicare informazioni relative ai propri progressi;
- un meccanismo di previsione dei possibili obiettivi raggiungibili dal cliente.

1.2 Scopo del documento

L'obiettivo di questo documento è fornire un modello standard per identificare, controllare, manutenere e verificare le versioni di tutti i Configuration Items. In particolare, gli scopi del CMP sono:

- mantenere l'integrità del prodotto software durante tutta la sua vita;
- supportare le attività di sviluppo e manutenzione;
- massimizzare la produttività riducendo gli errori derivanti dalla modifica degli artefatti prodotti.

In questo documento sono indicati:

- l'identificazione dei Configuration Items;
- il controllo del processo di cambiamento di un documento;
- la procedura da seguire nella richiesta dei cambiamento;
- i responsabili del SCM.

1.3 Riferimenti ad altri documenti

Di seguito la lista agli altri documenti di management:

- Risk Management Plan
- Schedule Management Plan
- Configuration Management Plan
- Quality Management Plan
- Software Project Management Plan

Di seguito inseriamo anche collegamenti ai due CMDB:

- GitHub
- OneDrive (solamente account UNISA)

2. Management

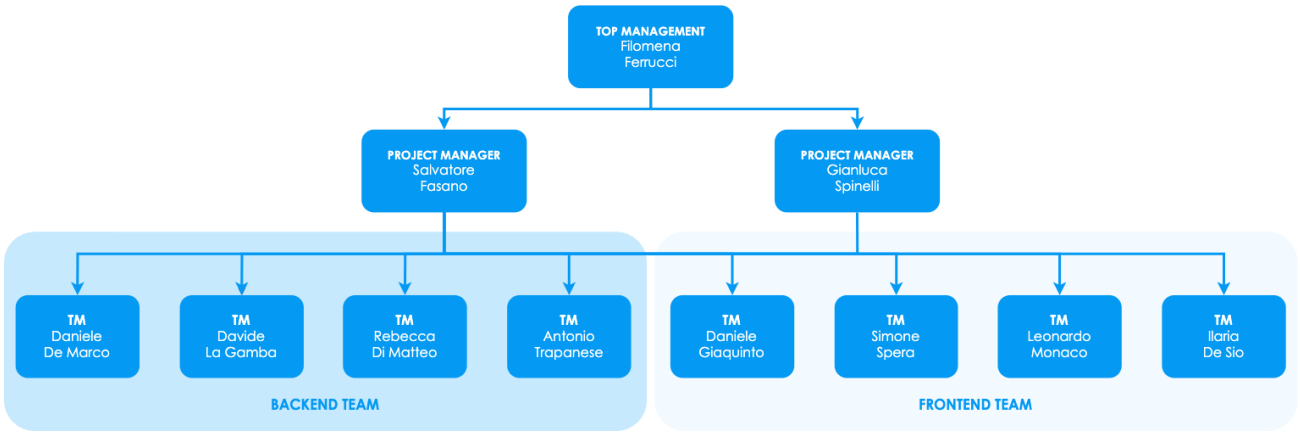
2.1 Fasi del progetto

Sono state identificate le seguenti principali fasi del progetto:

1. Avvio del progetto
2. Requirements Elicitation
3. Requirements Analysis
4. System Design
5. System Test Plan and Specification Design
6. Object Design
7. Source Code Implementation
8. Testing
9. Rilascio (creazione manuali d'installazione e d'uso)

2.2 Organizzazione

Di seguito è riportato un diagramma rappresentante l'organizzazione del progetto.



2.3 Ruoli e responsabilità

Di seguito verranno definiti i ruoli e le relative responsabilità nella gestione delle configurazioni.
I **Project Managers** sono responsabili di tutta la gestione degli aspetti manageriali. In particolare, si occuperanno di:

- identificare i Configuration Items
- accettare o rifiutare Change Request
- assegnare delle responsabilità circa eventuali Change Requests

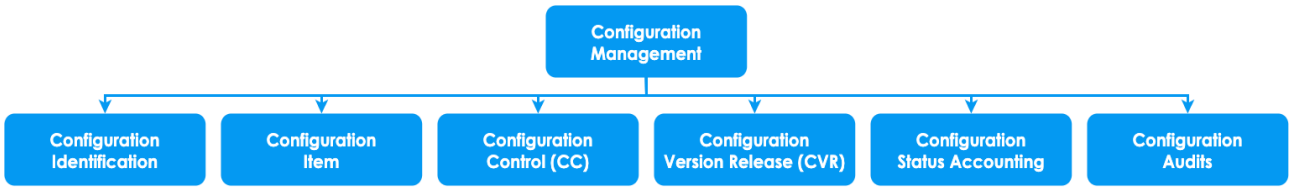
Ogni Team Members riveste il ruolo di **Developer**, che è responsabile di:

- compilare i documenti di revisione
- implementare le change request per gli artefatti prodotti

Inoltre, per ogni task assegnato, viene evidenziato un Developer che assume il ruolo di Reviewer: ciò significa che prima di sottomettere il task in revisione ai Project Managers, viene effettuata una prima revisione.

3. Attività

Il Configuration Management consta delle seguenti attività:



3.1 Configuration Identification

Il Configuration Management si applica sia alla documentazione che ai sorgenti del progetto.
Una volta identificati i Configuration Item, i Project Managers, seguendo le direttive del Top Manager, assegneranno un nome all'item e lo inseriranno nel rispettivo CMDB in uno stato iniziale. Subito dopo, l'item sarà assegnato al team di sviluppo che dovrà lavorarci e tenere traccia delle modifiche attraverso un log con identificativo incrementale (Revision History), spesso situato nell'artefatto stesso.

3.2 Configuration Item

I Configuration Item rappresentano tutti i tipi di oggetti che sono coinvolti nel Configuration Control. Gli elementi che entreranno a fare parte dei Configuration Item sono elencati di seguito:

- Documenti per la gestione e l'esecuzione del progetto.
- Documenti riguardanti lo sviluppo del sistema.
- Documenti di carattere tecnico del sistema.
- Applicativo software con relativa documentazione.
- Altri documenti a discrezione dei Project Managers.

Ogni Configuration Item è caratterizzato da:

- Un identificativo univoco solitamente della forma: "2021_[Acronimo_Documento]_C07_FitDiary_Fasano-Spinelli_V[x.y]".
- Un numero di versione, della forma "x.y" dove x è usato per indicare la consegna o il raggiungimento di una milestone, mentre y per indicare modifiche di minore entità.

Al momento della sua individuazione e inserimento nel CMDB, allo item viene assegnata la versione 0.1 indicante lo stato iniziale. Successive modifiche vanno ad incrementare solamente la y della versione. Alla prima release del documento o al raggiungimento di una milestones ad esso dedicata, la versione diventa 1.0. La x verrà nuovamente incrementata solo in caso di nuove consegne, nuove milestones, o modifiche particolarmente importanti.

3.2.1 Configuration Management Database (CMDB)

Nell'ambito del progetto FitDiary verranno utilizzati due diversi CMDB:

- **One Drive**: per la gestione dei documenti riguardanti il design dell'applicativo (RAD, SDD, ODD, Documenti di Testing, Manuali, ecc.).
- **GitHub**: per la gestione del codice sorgente del sistema.

Entrambi i sistemi offrono nativamente funzioni per il versioning degli artefatti. In particolare, per quanto concerne il CMDB One Drive, ogni documento ha un riferimento alla versione nel nome e, all'interno di esso, una tabella di Revision History, la quale consente di associare ad ogni versione le relative modifiche ed i rispettivi partecipanti. Invece, per quanto concerne la gestione del codice sorgente, ci si affiderà alle funzionalità offerte dal CMDB GitHub.

Inoltre, per garantire una migliore comprensione delle modifiche al codice sorgente, si favoriscono commit di piccola taglia (poche modifiche) e si utilizzerà, per la stesura del messaggio di commit, il seguente standard ideato da Google https://www.conventionalcommits.org/en/v1.0.0/.

3.3 Configuration Control (CC)

Una volta che un Configuration Item rientra nella baseline, tutte le modifiche ad esso fatte devono seguire un preciso protocollo. Chiunque facente parte del team può sottomettere una Change Request. I Project Managers valuteranno se approvare o rifiutare la Change Request sulla base di diversi fattori quali: l'impatto dei cambiamenti sul progetto, se la modifica rientra o meno nell'ambito delle versioni correnti, priorità, risorse, livello di sforzo, rischio e qualsiasi altro criterio ritenuto rilevante dai Project Managers. Se approvata, l'implementazione della modifica verrà assegnata ad uno o più Developer, eventualmente lo stesso che l'ha proposta. A termine dell'implementazione il Developer si assicurerà della sua corretta propagazione e consistenza. Nel caso di modifiche al codice, il Developer dovrà assicurare il corretto build e pass dei test dopo l'implementazione della Change Request (alla sottomissione della pull request), avvalendosi del sistema di Continuous Integration GitHub Action. Sarà poi compito dei Project Managers accettare la modifica ed integrarla nel progetto effettivo. Per quanto riguarda gli accessi, l'intero team di sviluppo avrà privilegi completi sulla cartella One Drive, intesi come scrittura e lettura degli artefatti. Nel

caso della repository GitHub invece, i Developers dovranno lavorare usando il sistema di branch offerto dalla piattaforma secondo le seguenti modalità: tutte le modifiche verranno apportate al contenuto del branch "dev", al termine di uno Sprint allora si provvederà ad inviare una pull request per effettuare il merge sul branch main. I Project Managers, una volta sicuri della qualità delle modifiche implementate, accetteranno la richiesta e integreranno i commit di modifica nel progetto effettivo; a tal punto, grazie al sistema GitHub Action, si avvierà un processo di testing automatico (unità, integrazione e generazione report branch coverage e CheckStyle). Inoltre saranno presenti i servizi Heroku e Vercel che, in presenza di push sul branch main, effettueranno automaticamente il deploy rispettivamente del backend e del frontend.

3.3.1 Presentazione Change Request

La proposta di una Change Request può essere sottomessa ai project manager da ogni stakeholder rispettando la struttura dell'esempio fornito di seguito:

Change Request Form	
Project: FitDiary	**Number:** 1
Change Requester: Salvatore Fasano, Gianluca Spinelli	**Date:** 30/12/2021
Requested Change: Il cliente può eliminare un suo protocollo	
Request Type: Aggiunta di un nuovo requisito funionale	
Change Analyzer: FS, GS	**Analysis date:** 31/12/2021
Oggetto: Aggiungere la possibilità al cliente di eliminare un proprio protocollo dallo storico.	
Descrizione: Un cliente dopo aver scaricato un protocollo vecchio, deve poter eliminarlo dal proprio storico per qualsiasi motivo.	
Commenti: L'aggiunta di questo nuovo requisito richiede molte modifiche sia alla logica di business che all'interfaccia grafica.	**Data di Cambiamento:** 05/01/2022

3.3.2 Valutazione Change Request

I Project Managers analizzeranno l'impatto del cambiamento in termini di modifiche da apportare, rischi associati e corrispettivo valore di business. A seconda della natura della CR, i Project Managers potrebbero accettare il cambiamento, rifiutarlo o richiedere un consulto col Top Management ed il Cliente.

3.3.3 Approvazione o rifiuto Change Request

La CR verrà accettata o rifiutata dopo aver effettuato delle analisi circa la stima per la realizzazione (in combinazione con lo stato attuale del progetto) e dall'eventuale accettazione da parte del Cliente.
Non è stato previsto un tempo di decorrenza tra la sottomissione e l'accettazione/rifiuto.

3.4 Configuration Version Release (CVR)

L'attività di management delle release è svolta nel momento in cui si verificano condizioni che determinano il rilascio di una nuova release. Esse sono:

- la risoluzione di uno o più bug porta ad un incremento della y all'interno della versione del codice sorgente;
- la risoluzione o la modifica di piccole parti della documentazione dovute ad inconsistenze rilevate porta all'incremento della y all'interno della versione del documento;
- la risoluzione di uno o più bug di notevole entità porta all'aggiornamento della x all'interno del codice sorgente;
- il completamento di un documento con l'aggiunta e la modifica di un numero ingente di sezioni porta all'aumento della x.

3.5 Configuration Status Accounting

Durante il tempo che intercorre tra due consegne o milestones, i Configuration Items sono conservati nella cartella One Drive o su GitHub, nelle quali avvengono le principali attività di modifica. L'accounting dello stato della configurazione avverrà ogni milestones importante o release (consegna) degli artefatti prodotti. In tale situazione, i Project Managers utilizzeranno la funzionalità di release fornita da GitHub per creare una baseline.

3.6 Configuration Audits

Al raggiungimento di una milestone, o poco prima di una consegna, saranno effettuati dei lavori di revisione da parte di tutto il team di sviluppo su ogni Configuration Items facente parte della baseline. Dopo tali lavori, i Project Managers visioneranno personalmente gli items assicurando:

- la corretta numerazione delle versioni;
- la consistenza delle modifiche tra items collegati;
- la qualità delle descrizioni delle modifiche implementate;
- la presenza di tutti gli items;
- la corretta organizzazione dei CMDBs

In caso di necessità, i Project Managers potranno richiedere il rollback di alcune modifiche. Tale rollback avverrà avvalendosi dei sistemi integrati in One Drive e GitHub.