

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Configuration Management Plan  GreenLeaf   |  |  | | --- | --- | | Riferimento |  | | Versione | 1.0 | | Data | 24/11/2022 | | Destinatario |  | | Presentato da | Angelo Afeltra, Antonio Giametta, Raffaele Squillante | | Approvato da |  | |

RevisionHistory

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versione** | **Descrizione** | **Autori** |
|  | 1.0 | Prima stesura | A. Afeltra,  A. Giametta,  R. Squillante |

Sommario

[1. Introduzione 4](#_Toc126856127)

[2. Management 4](#_Toc126856128)

[3. Attività 4](#_Toc126856129)

1. Introduzione
2. Ambito

L’obiettivo di GreanLeaf è fornire uno strumento di sensibilizzazione assicurando che tutti gli stakeholder possano prendere parte alla causa in modo agevole ed efficiente. Nello specifico la piattaforma dovrà supportare il monitoraggio e la previsione dell’inquinamento dell’area. Inoltre dovrà permettere l’adozione di uno o più albero per cui sarà poi disponibile la geolocalizzazione, il calcolo della CO2 emessa e l’aggiornamento in tempo reale sullo stato dell’albero.

Come ultima funzionalità il sistema dovrà permettere il calcolo della CO2 causata dalla persona.

1. Scopo del documento

Per consentire il corretto progresso del progetto e la sua riuscita risulta necessario produrre un Configuration Management Plan (CMP). In tale documento saranno trattati ruoli nell’ambito della configuration mangament, configuration items e modifiche durante lo sviluppo del progetto.

1. Riferimenti ad altri documenti

Di seguito viene elencata una lista agli altri documenti di management:

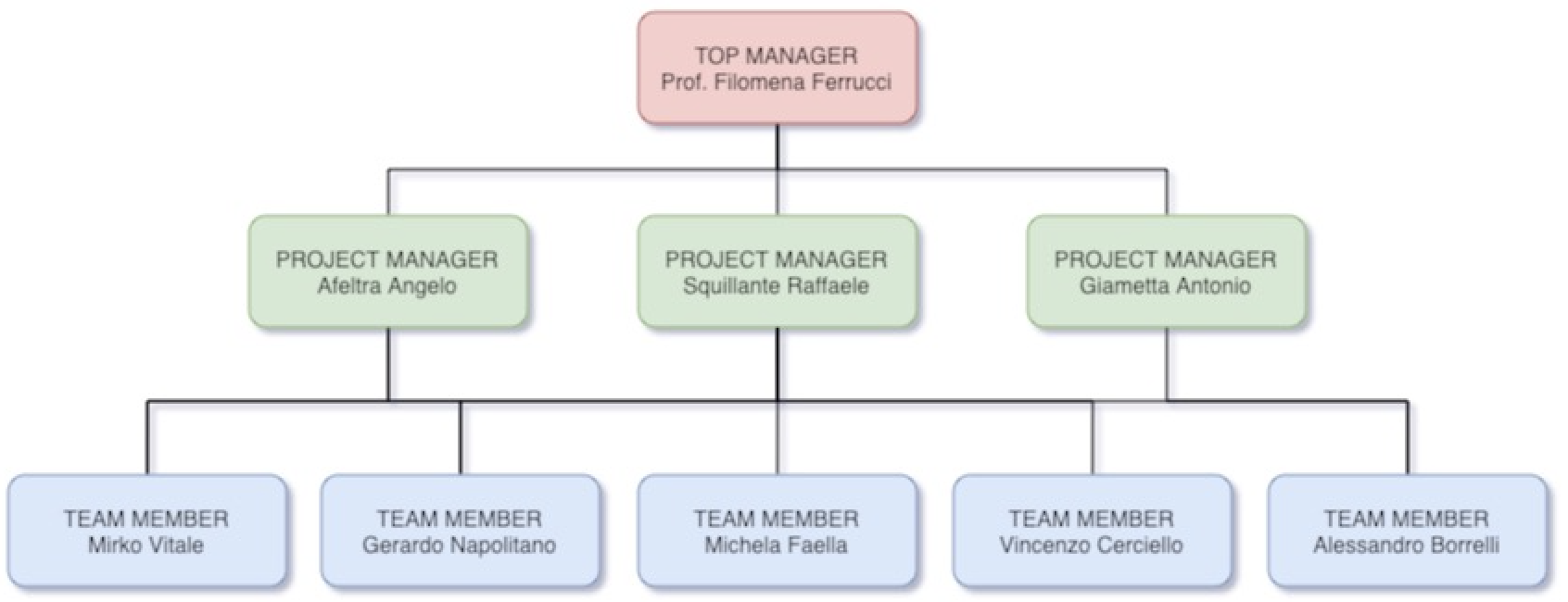
* Risk Management Plan
* Schedule Management Plan
* Configuration Management Plan
* Quality Management Plan
* Software Project Management Plan

1. Management
2. Fasi del progetto

Sono state identificate le seguenti principali fasi di progetto:

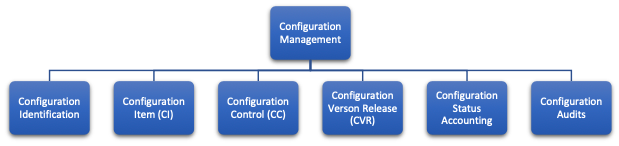
1. Avvio del progetto
2. Requirements Elicitation and Analysis
3. System Design
4. System Test Design
5. Object Design
6. Implementazione
7. Testing
8. Rilascio
9. Organizzazione

Di seguito è mostrato un diagramma dell’organizzazione.



1. Ruoli e responsabilità

I seguenti ruoli e responsabilità riguardano il Configuration Management Plan per il progetto GreenLeaf:

1. Attività
2. Configuration Identification

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, lo item sarà assegnato al team di sviluppo che dovrà lavorarci e tenere traccia delle modifiche attraverso un log con identificativo incrementale, spesso situato nell’artefatto stesso.

1. Configuration Item

Ricadono nella categoria di Configuration Item:

* 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 del PM

Ogni Configuration Item è caratterizzato da:

* Un identificativo univoco solitamente del formato “[Anno]\_[Acronimo Documento]\_C09\_Ver[x.y]”
* Un numero di versione, della forma “x.y” dove x è usato per indicare la consegna o il raggiungimento di una milestones, mentre y per indicare modifiche di minore entità.

Al momento della sua individuazione e inserimento nel CMDB, all item viene assegnata la versione 0.1 indicante lo stato di iniziazione. Successive modifiche vanno a incrementare solo la y della versione. Alla prima release del documento o al raggiungimento di una milestones ad esso dedicata, la versione diventa 1.0. Da tale momento in poi ogni modifica allo item deve essere approvata da PM e sottoposta secondo protocollo e porterà all’incremento della sola y. La x verrà nuovamente incrementata solo in caso di nuove consegne, nuove milestones, o modifiche di natura particolarmente importanti, a discrezione del PM.

**Configurations Management Database (CMDB)**

Nell’ambito del progetto GreenLeaf sono stati individuati due diversi CMDB:

* **One Drive:** per la gestione dei documenti riguardanti il design dell’applicativo (RAD, SDD, ODD e altri)
* **GitHub:** per la gestione del codice sorgente di BiblioNet.

Entrambi i sistemi offrono nativamente funzioni per il versioning degli artefatti, alle quali sono state aggiunte ulteriori protocolli per irrobustire il tutto. Nello specifico, nel caso dei documenti su One Drive, ogni artefatto ha internamente una tabella “Revision History” le quali entry associano ad una specifica versione del documento una descrizione della modifica implementata. Ancora, ogni rilascio di una versione con modifiche importanti (che quindi vanno a incrementare il primo numero della versione) prevedono la creazione di una nuova copia del documento separata dalla precedente versione. Nel caso di GitHub, invece, ci si affida completamente al sistema basato su commit del sito stesso, andando però a evidenziare in maniera chiara un commit di rilascio corrispondente alle versioni con l’incremento del primo numero.

1. Configuration Control

Una volta che un Cofiguration 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 al Revision Leader che ha il compito di fare una valutazione preliminare. Se il Revision Leader approva la change request, essa passa al Project Manager che la valuta e decide se approvarla o rifiutarla. Se approvata, l’implementazione della modifica viene assegnata ad uno o più developer, eventualmente lo stesso che l’ha proposta, che, dopo averla implementata, si assicurerà della sua corretta propagazione e consistenza.

Nel caso di modifiche al codice, l’implementatore dovrà assicurare il corretto build e pass dei test dopo l’implementazione della change request, eventualmente avvalendosi dei sistemi di continuous integration offerti da GitHub Action.

1. 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.

1. Configuration Status Accounting

Durante il tempo che intercorre tra due consegne o milestones, i CI sono conservati nella cartella One Drive o su GitHub, nelle quali avvengono le principali attività di modifica.

Lo accounting dello stato della configurazione avverrà ogni milestones importante o release (consegna) degli artefatti prodotti. In tale situazione, il PM dovrà creare una cartella contenente tutte le versioni più recenti, e approvate, degli artefatti richiesti, eventualmente aggiungendo precedenti versioni per motivi didattici. All’interno di tale cartella, sarà compreso un file “readme” con lo scopo di informare chi visiona di informazioni utili alla comprensione degli artefatti.

Nel caso del codice sorgente, sarà effettuato un commit univocamente identificato con un numero di rilascio indicante il raggiungimento della milestones o deliverables. Sarà inoltre prodotto un eseguibile dell’applicativo contenuto anch'esso nella cartella con gli altri artefatti.

1. Configuration Audit

Al raggiungimento di una milestone, o poco prima una consegna, saranno effettuati dei lavori di revisione da parte di tutto il team di sviluppo su ogni CI facente parte della baseline. Dopo tali lavori, il PM visionerà 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, il PM potrà richiedere il rollback di alcune modifiche. Tale rollback avverrà avvalendosi dei sistemi integrati in One Drive e GitHub.