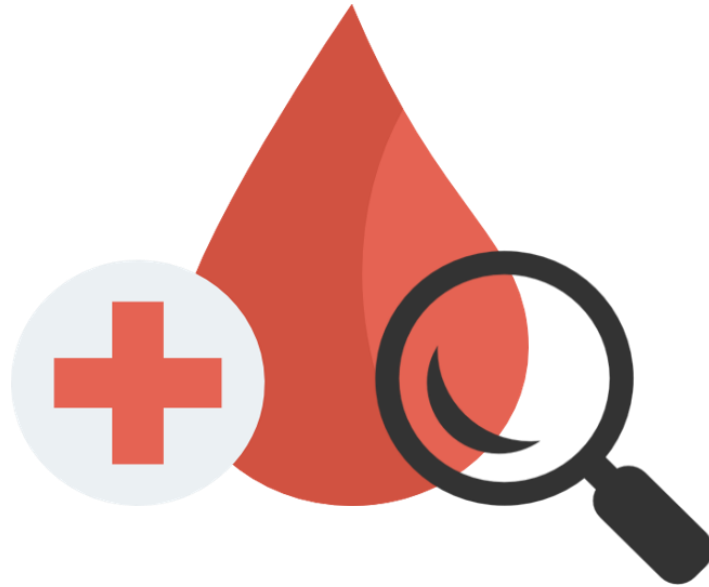




Laurea Magistrale in informatica-Università di Salerno  
Corso di Gestione dei Progetti Software- Prof.ssa F. Ferrucci



# Configuration Management Plan

## Progetto

# Digital Donation

|               |                                  |
|---------------|----------------------------------|
| Riferimento   |                                  |
| Versione      | 1.0                              |
| Data          | 16/01/2022                       |
| Destinatario  | Prof.ssa F. Ferrucci             |
| Presentato da | Francesco Abate, Carmine Ferrara |
| Approvato da  |                                  |



## Sommario

---

|  |    |
|--|----|
| Revision History .....                                     | 3  |
| 1. Introduzione .....                                      | 4  |
| 1.1 Documenti di riferimento, definizioni e acronimi ..... | 4  |
| 1.2 Scopo .....  | 4  |
| 1.3 Ambito .....   | 5  |
| 2. Management.....   | 5  |
| 2.1 Fasi.....  | 5  |
| 2.2 Organizzazione .....                                   | 6  |
| 2.3 Ruoli e responsabilità .....                           | 6  |
| 3. Attività .....  | 7  |
| 3.1 Identificazione dei Configuration Item.....            | 7  |
| 3.2 Controllo della configurazione .....                   | 8  |
| 3.2.1 Presentazione della Change Request.....              | 10 |
| 3.2.2 Valutazione della Change Request .....               | 11 |
| 3.2.3 Approvazione o rifiuto della Change Request .....    | 11 |
| 3.3 Gestione delle release .....                           | 11 |
| 3.4 Configuration Status Accounting .....                  | 12 |
| 3.5 Configuration Audits .....                             | 12 |



## Revision History

| Data       | Versione | Descrizione              | Autori                              |
|------------|----------|--------------------------|-------------------------------------|
| 08/12/2021 | 0.1      | Prima stesura            | Francesco Abate,<br>Carmine Ferrara |
| 16/01/2022 | 1.00     | Revisione di consistenza | Francesco Abate,<br>Carmine Ferrara |



# Configuration Management Plan del Progetto

## Digital Donation

---

## 1. Introduzione

---

### 1.1 Documenti di riferimento, definizioni e acronimi

Segue una lista di acronimi e abbreviazioni utilizzati nella stesura del CMP:

- CMP: Configuration Management Plan;
- RAD: Requirement Analysis Document;
- SDD: System Design Document;
- ODD: Object Design Document;
- CI: Configuration Item;
- SMC: Software Configuration Management;
- MC: Configuration Management;
- CR: Change Request;
- CCB: Configuration Control Board;
- PM: Project Manager;
- SOW: Statement Of Work;
- BC: Business Case.

### 1.2 Scopo

L'obiettivo del CMP è quello di fornire un modello che riguardi l'identificazione, il controllo e la manutenzione di tutti i CI, cioè di tutti gli artefatti soggetti a controllo di configurazione. Insomma, si sente la necessità di:

- Mantenere integrità e coerenza tra gli artefatti del prodotto software;
- Supportare le attività di sviluppo e in particolare di manutenzione tenendo traccia di ogni modifica svolta, con conseguente massimizzazione della produttività.



Saranno, quindi, identificati tutti i CI necessari; sarà indicato il processo di controllo dei cambiamenti dei CI; sarà indicata la procedura da seguire nel caso sia necessario apportare cambiamenti; saranno indicati i responsabili del SCM.

### 1.3 Ambito

L'Ente Nazionale della Ricerca del Sangue intende digitalizzare le attuali politiche interne di ricerca donatori volontari in modo da semplificare e velocizzare le metodologie adottate dai propri operatori interni per l'organizzazione delle sedute di donazione nei centri locali o con soluzioni di trasporto mobile. A tal scopo, l'ente rende nota l'esigenza di poter offrire meccanismi di semplificazione di tale pratica, al fine di alleggerire l'effort del personale interno, il quale potrà dedicarsi con maggiore impegno a mansioni più consone al proprio ruolo quali l'organizzazione di risorse per le sedute di donazione.

## 2. Management

---

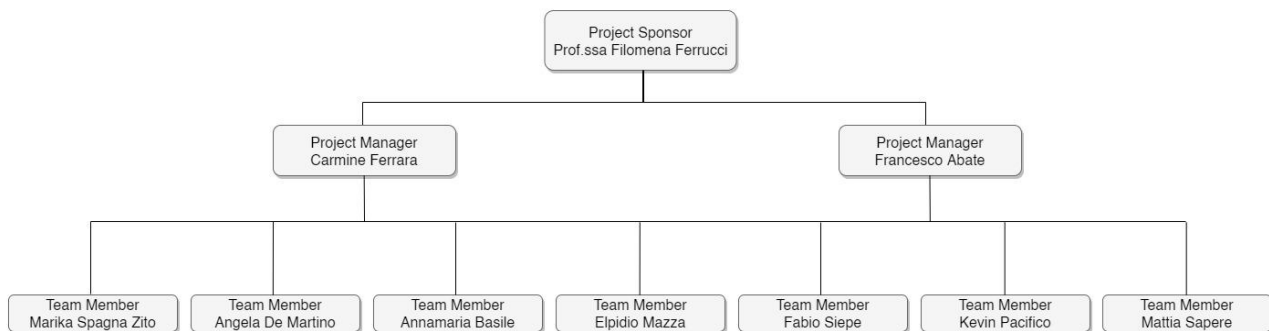
### 2.1 Fasi

Durante un'analisi preliminare sono state individuate le seguenti fasi di progetto:

1. Avvio;
2. Requirements Elicitation;
3. System Design;
4. Object Design;
5. Implementazione;
6. Testing;
7. Rilascio.



## 2.2 Organizzazione



## 2.3 Ruoli e responsabilità

Si è deciso di assegnare un certo ruolo ad ogni team member, con conseguente assunzione delle relative responsabilità. I ruoli sono stati assegnati rispettando passioni, preferenze personali ed esperienza pregressa, ascoltando anche i pareri dei team member stessi dato che, lasciando scegliere a loro, i project manager si aspetteranno che quest'ultimi lavorino con maggior responsabilità e fiducia verso il proprio operato.

I ruoli individuati sono i seguenti:

- Frontend developer;
- Backend developer;
- Fullstack developer;
- Documentation writer;
- Database architect.

Segue la matrice di associazione dei ruoli:

| Nome              | Ruoli  | Contatti   |
|-------------------|--|--|
| Angela De Martino | Documentation writer<br>Database architect<br>Frontend developer | a.demartino68@studenti.unisa.it  |
| Annamaria Basile  | Documentation writer<br>Database architect<br>Frontend developer | <a href="mailto:a.basile40@studenti.unisa.it">a.basile40@studenti.unisa.it</a> |



|                           |   |  |
|---------------------------|---|--|
| <b>Marika Spagna Zito</b> | Documentation writer<br>Fullstack developer | <a href="mailto:m.spagnazito1@studenti.unisa.it">m.spagnazito1@studenti.unisa.it</a> |
| <b>Fabio Siepe</b>        | Documentation writer<br>Fullstack developer | <a href="mailto:f.siepe1@studenti.unisa.it">f.siepe1@studenti.unisa.it</a>           |
| <b>Mattia Sapere</b>      | Documentation writer<br>Fullstack developer | <a href="mailto:m.sapere16@studenti.unisa.it">m.sapere16@studenti.unisa.it</a>       |
| <b>Elpidio Mazza</b>      | Documentation writer<br>Backend developer   | <a href="mailto:e.mazza7@studenti.unisa.it">e.mazza7@studenti.unisa.it</a>           |
| <b>Kevin Pacifico</b>     | Documentation writer<br>Backend developer   | <a href="mailto:k.pacifico@studenti.unisa.it">k.pacifico@studenti.unisa.it</a>       |

### 3. Attività

Si prevede che il CM sia costituito dalle seguenti attività:

1. Identificazione dei CI;
2. Controllo della configurazione;
  - 2.1 Presentazione della Change Request;
  - 2.2 Valutazione della Change Request;
  - 2.3 Approvazione o rifiuto della Change Request;
3. Gestione della release;
4. Configuration Status Accounting;
5. Configuration Audits.

#### 3.1 Identificazione dei Configuration Item

I CI rappresentano tutti gli artefatti che sono coinvolti nel configuration control, quindi in generale si tratta di:

- Documentazione di sviluppo del sistema;
- Documentazione di gestione del sistema;
- Documentazione descrittiva del sistema;
- Componenti del sistema sviluppato;
- Qualsiasi altro artefatto a discrezione dei project manager.

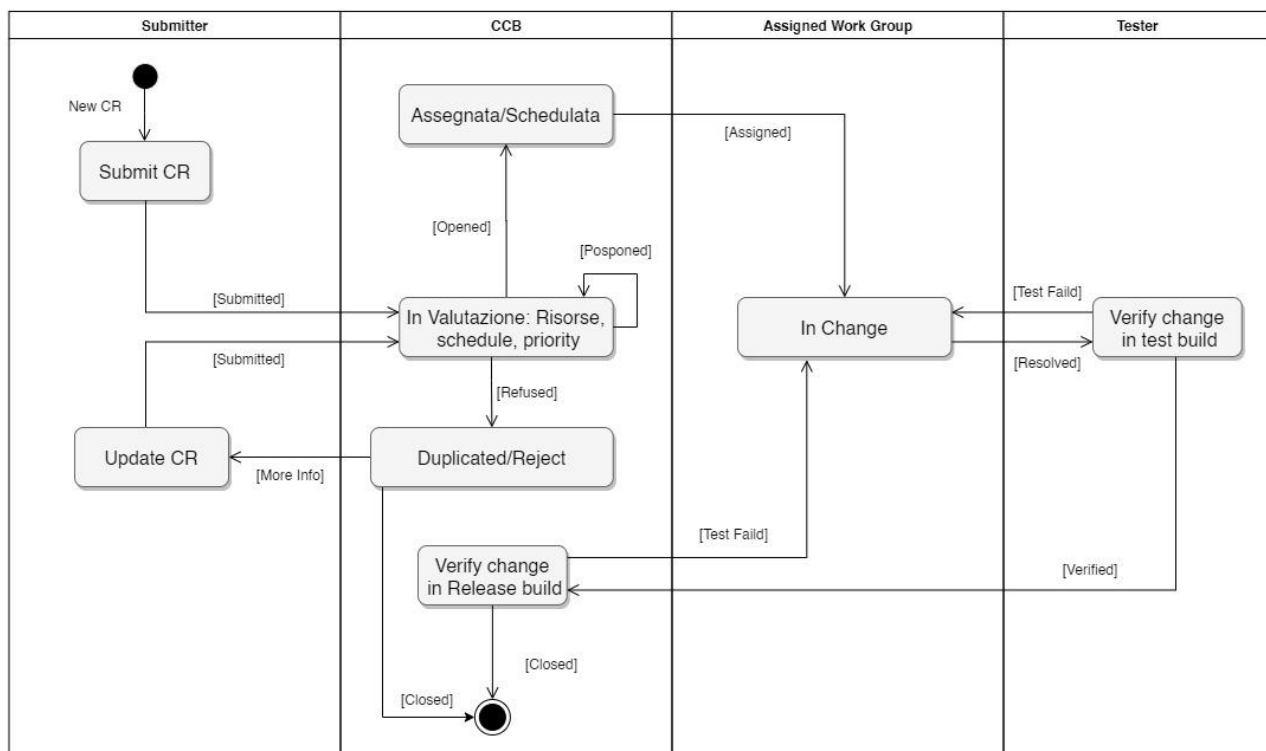
L'attività di identificazione dei CI consiste nell'individuare tutti gli artefatti che saranno gestiti tramite SCM. Si cercherà, quindi, di redigere durante o dopo lo sviluppo l'elenco degli artefatti che compongono la configurazione del prodotto in un momento in particolare.

Ad ogni CI sarà attribuito un identificatore univoco, in modo da poterlo distinguere da tutti gli altri, indicante anche la versione dell'artefatto; le versioni draft non saranno soggette a questo vincolo. In generale, la versione avrà la forma "x.y", quindi in base all'entità dei cambiamenti da apportare si valuterà se incrementare il valore "x" o il valore "y".

### 3.2 Controllo della configurazione

Nel caso si voglia svolgere una qualsiasi modifica ad un qualsiasi work product presente all'interno di una baseline, sarà necessario avviare l'iter inerente alla presentazione di una CR. L'iter prevede i seguenti step:

1. Presentazione della CR da parte di uno stakeholder;
2. Valutazione della CR;
3. In caso di approvazione della CR, a quest'ultima viene assegnato un gruppo di lavoro;
4. Dopo aver apportato le modifiche al sistema, quest'ultimo viene adeguatamente testato;
5. Viene aggiornato il changelog della nuova release.



Volendo osservare ogni attività che compone il ciclo di vita di una CR nel dettaglio:





| Attività  | Descrizione   | Responsabilità                              |
|---|---|---|
| Submit CR   | Uno stakeholder qualsiasi può inviare una nuova CR, la quale verrà registrata, tracciata e inserita nella coda di revisione CCB, impostando lo status della CR su "inoltrata".        | Stakeholder che effettua il submit della CR |
| In valutazione (Review)                               | Vengono revisionate le CR inviate dagli stakeholder. Viene esaminato il contenuto della CR per determinare se la richiesta effettuata sia valida e fattibile.                         | CCB   |
| Duplicate / Reject (Duplicato confermato o rifiutato) | Casistica in cui una CR sia duplicata o rifiutata, quindi un delegato del CCB la rifiuterà o la sottometterà al cliente per raccogliere maggiori informazioni.                        | Delegato CCB                                |
| Update CR   | Lo stakeholder che ha inviato la CR ha la possibilità di aggiornarla aggiungendo nuovi dettagli e informazioni.   | Stakeholder che ha effettuato il submit     |
| Assegnata / Schedulata                                | Il PM associa alla CR un gruppo di lavoro che si occuperà di soddisfare la richiesta.   | PM  |
| In change   | Il gruppo di lavoro assegnato alla CR implementa ed effettua unit testing in relazione ai cambiamenti richiesti, per poi contrassegnare la CR come "risolta".                         | Gruppo di lavoro o singolo lavoratore       |
| Verify change in Test build                           | Le modifiche apportate vengono inserite in una coda di test per essere assegnate ad un tester e verificate in una build di prova del prodotto.  | Tester                                      |
| Verify change in Release build                        | La CR viene inserita in una coda di rilascio per poter essere verificata, per produrre note di rilascio per una nuova release e così via. Fatto ciò, la CR viene chiusa con successo. | CCB   |



| Change Request Form |  |               |            |
|---------------------|--|---------------|------------|
| Project             | Digital Donation   | Number        | 1          |
| Change Requester    | Mario Rossi - Operatore di sede Centrale ENRS  | Date          | 07/12/2021 |
| Requested Change    | Ogni qualvolta viene emesso un tesserino da una sede locale, esso dovrà essere approvato definitivamente da un'operatore di sede centrale  |               |            |
| Change Analyzer     | CCB  | Analysis Date | 08/12/2021 |
| Oggetto             | Prevedere le nuove funzionalità di revisione e validazione da parte di un'operatore di sede centrale dell'ENRS di ogni tesserino digitale emesso da una sede locale per un volontario donatore   |               |            |
| Descrizione         | Ogni operatore di sede centrale dovrà essere abilitato all'accesso alla piattaforma, ed essere reindirizzato in una dashboard propria. Nella sua dashboard dovrà avere a disposizione un'interfaccia di visualizzazione di tutti i tesserini di sede locale emessi e non ancora approvati (preferibilmente gli stessi dovranno essere filtrabili per Sede). Dall'elenco di visualizzazione l'operatore di sede centrale dovrà avere la facoltà di: Visionare i dettagli di un tesserino, Approvare definitivamente il tesserino (ed abilitare l'accesso alla piattaforma al volontario), rifiutare la richiesta di tesseramento inviando notifica automatica sia al candidato volontario che alla sede locale. |               |            |
| Impatto             | La richiesta di cambiamento può richiedere l'aggiunta di 3 requisiti funzionali: La visualizzazione dell'elenco di tesserini non validati, La validazione e il rifiuto della domanda di tesseramento da parte della sede centrale. Inoltre, è necessario prevedere differenti stati (In attesa di approvazione, approvato o rigettato) dell'artefatto "Tesserino digitale", per supportare il processo di validazione, e consentire l'accesso al volontario solo quando la domanda è stata validata dalla sede centrale.   |               |            |
| Commenti            | La CR richiede molti cambiamenti e ampliamenti del sistema, non è scontato che si rientri nel budget volendola integrare per intero nella release in sviluppo  |               |            |
| Data di Cambiamento | 05/01/2022   |               |            |

### 3.2.1 Presentazione della Change Request

La presentazione di una CR può comportare la variazione dei requisiti del sistema, quindi potrebbe aggiungerne dei nuovi, rimuoverli o modificarli.

Ogni stakeholder può presentare una CR ai PM rispettando il seguente template:



### 3.2.2 Valutazione della Change Request

I PM analizzeranno la CR effettuando impact analysis per comprendere quanto le modifiche impattino sul sistema sviluppato fino a quel momento, valutando i rischi associati e il valore di business.

La CR analizzata potrà poi essere categorizzata in base alla richiesta di cambiamento che contiene:

- Nuovo requisito (2/5);
- Eliminazione di un requisito (1/5);
- Modifica di un requisito, bug fixing (4/5);
- Modifica di un requisito, bug fixing critico (5/5);
- Modifica di un requisito, evoluzione (3/5).

Nel caso siano state presentate diverse CR, quest'ultime verranno successivamente risolte in un particolare ordine dettato dalla priorità in cui ricade ognuna di esse.

### 3.2.3 Approvazione o rifiuto della Change Request

L'approvazione o il rifiuto della CR presentata dipende dal risultato dell'impact analysis, tempi e costi stimati per la realizzazione della modifica e dell'accettazione del cliente, eventuale ri-pianificazione delle attività di progetto nel caso in cui le modifiche implicino ritardi su particolari attività e così via. Insomma, si tiene conto di diversi fattori che influenzeranno l'approvazione o il rifiuto.

Non è previsto un tempo di decorrenza tra sottomissione e accettazione o rifiuto.

## 3.3 Gestione delle release

L'attività di gestione delle release è svolta ogni qualvolta in cui viene rilasciata una nuova release.

In particolare, una nuova release richiede che la propria versione sia diversa dalla release precedente.

Sia "x.y" la versione della release, osserviamo come cambia in relazione ai cambiamenti apportati:

- Modifiche al codice, risoluzione di bug: incremento della y;
- Modifica della documentazione dovuta ad inconsistenze: incremento della y;
- Modifiche massicce al codice, risoluzione di bug critici: incremento della x;
- Implementazione di una nuova funzionalità: incremento della x;
- Rimozione di una funzionalità precedentemente implementata: incremento della x;
- Modifica massiccia della documentazione: incremento della x;
- Introduzione di un nuovo documento: incremento della x.



### 3.4 Configuration Status Accounting

I CI documentative sono memorizzati in un apposito repository su OneDrive, cloud storage di Microsoft multiplatforma che permette di hostare file e di sincronizzarli su qualsiasi device in maniera del tutto automatica, sia tramite webapp che tramite applicazione desktop / mobile. Tale applicazione permette di gestire agevolmente il versioning dei CI, permettendo di recuperare anche vecchie versioni di quest'ultimi. La struttura del repository di OneDrive prevede una suddivisione principale con una cartella contenente tutti i work product dei PM (cartella privata ai team members) e una cartella contenente tutti i work product dei team members; per ogni work product fondamentale, come SOW, BC, RAD, SDD e così via, viene posta una cartella dedicata posta nelle principali cartelle accennate.

Riguardo l'implementazione, si avrà un repository pubblico sulla piattaforma GitHub dato che permette di gestire agevolmente il versioning del codice sorgente e semplifica la segnalazione di eventuali bug o errori generici interni al team.

### 3.5 Configuration Audits

Prima delle consegne dei documenti sono fissate verifiche di configurazione formali. Per ogni versione è garantita la possibilità di effettuare roll-back e backup a versioni precedenti.

Per la documentazione di progetto è previsto il servizio di audit offerto da OneDrive, che garantisce la possibilità di rollback per qualsiasi CI.

Per il codice sorgente è previsto il servizio di audit offerto da GitHub che garantisce la possibilità di rollback del codice ad una versione precedente.

