



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

CHEMO SMART
SCHEDULER FOR CHEMOTERAPY



Configuration Management C02 - ChemoSmart

Versione	1.0
Data	22/02/2023
Destinatario	Prof.ssa Filomena Ferrucci
Presentato da	Alessandro Bergamo, Francesco Pio Ianuzziello
Approvato da	/



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

Revision History

Data	Versione	Descrizione	Autori
19/12/2022	0.1	Prima stesura del documento	A. Bergamo, F. P. Ianuzziello
26/12/2022	0.2	Aggiunta dei protocolli di configurazione riguardanti il codice sorgente	A. Bergamo, F. P. Ianuzziello
22/02/2023	1.0	Revisione per consegna	A. Bergamo, F. P. Ianuzziello

Project Managers

Nome	Acronimo	Informazioni di contatto
Alessandro Bergamo	AB	a.bergamo2@studenti.unisa.it
Francesco Pio Ianuzziello	FPI	f.ianuzziello1@studenti.unisa.it



Sommario

1. Introduzione	4
1.1. Ambito	4
1.2. Scopo del documento	4
1.3. Riferimenti ad altri documenti	4
2. Management.....	4
2.1. Fasi del progetto	4
2.2. Organizzazione.....	5
2.3. Ruoli e responsabilità	5
3. Attività	6
3.1. Configuration Identification	7
3.2. Configuration Item.....	7
3.3. Configuration Control.....	8
3.4. Configuration Version Release (CVR)	8
3.5. Configuration Status Accounting	9
3.6. Configuration Audit	9



1. Introduzione

1.1. Ambito

Il progetto Chemosmart vuole fornire alle istituzioni sanitarie uno strumento per semplificare le operazioni di scheduling delle terapie chemioterapiche per i pazienti, con lo scopo preciso di minimizzare gli sprechi di farmaci e di organizzare al meglio le poltrone e il personale medico. Obiettivo, quindi, di questo progetto sarà quello di fornire a tali istituzioni una Web App che consenta quanto detto, attraverso l'ausilio di una AI.

1.2. Scopo del documento

Per consentire che il progetto prosegua correttamente è necessario produrre un Configuration Management Plan (CMP) dove saranno trattati i ruoli nell'ambito della configuration management, configuration item e modifiche durante lo sviluppo del progetto.

1.3. Riferimenti ad altri documenti

A seguire un elenco degli altri documenti di management:

- [Risk Management Plan](#);
- [Schedule Management Plan](#);
- Configuration Management Plan;
- [Quality Management Plan](#);
- [Software Project Management Plan](#).

2. Management

2.1. Fasi del progetto

Sono state identificate le seguenti fasi principali di progetto:

1. Avvio del progetto

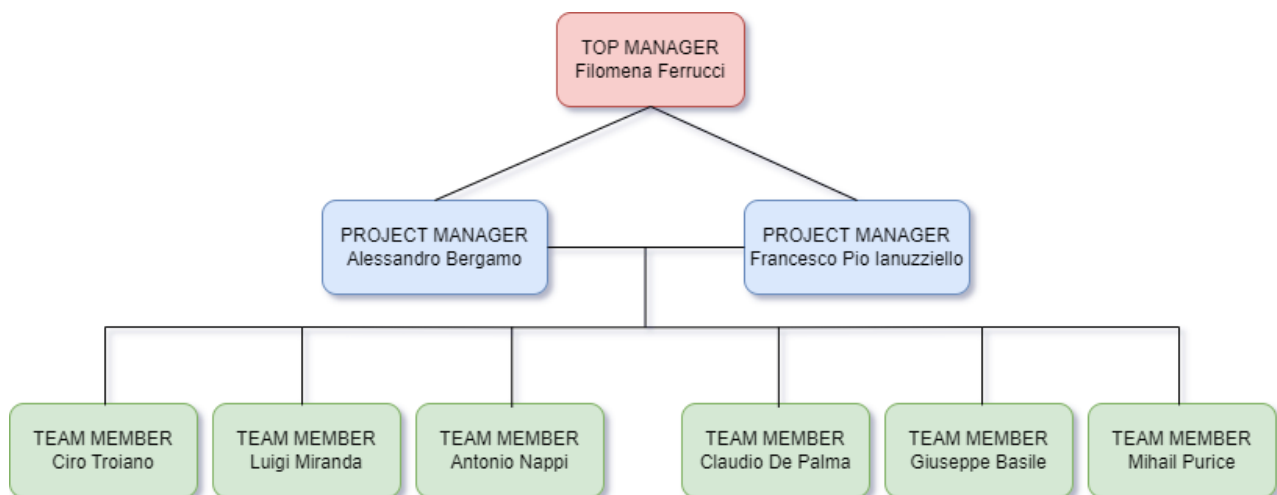


2. Requirements Elicitation and Analysis
3. System Design
4. System and Integration Test Design
5. Object Design
6. Implementazione
7. Testing
8. Rilascio

Inoltre, durante tutta la durata del progetto verrà effettuato una continua attività di Training per aiutare i Team Members nei loro lavori.

2.2. Organizzazione

A seguire un diagramma dell'organizzazione:



2.3. Ruoli e responsabilità

I seguenti ruoli e responsabilità riguardano il Configuration Management Plan per il progetto ChemoSmart.

Project Manager

Il Project Manager è colui che è responsabile della gestione di tutti gli aspetti manageriali del progetto.

Il PM è responsabile per:

- Identificatione dei Configuration Items



- Accettazione di change requests
- Assegnazione delle responsabilità delle change requests.

Revisione Leader

Il Lead Engineers è un membro del team di sviluppo che ha maggiori responsabilità nell'ambito della configuration management rispetto ai suoi colleghi.

Il Lead Engineers è responsabile per:

- Eseguire revisioni periodiche degli artefatti prodotti
- Comunica al PM risultati riguardanti le revisioni effettuate
- Comunica al PM eventuali necessità di cambiamenti negli artefatti prodotti

Developer

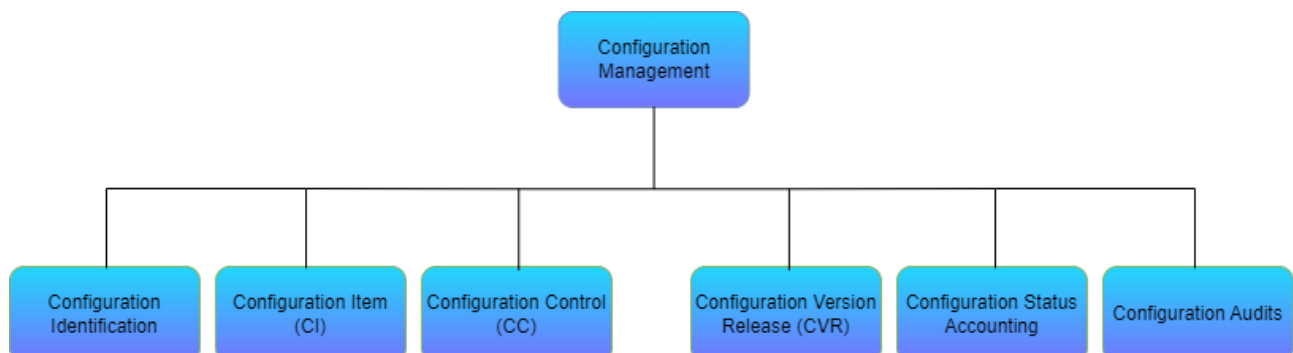
Ogni membro del team riveste il ruolo di engineer all'interno del progetto.

L'Engineer è responsabile per:

- Eseguire revisioni programmate assegnategli dal PM
- Compilare i documenti di revisione
- Implementare le change requests per gli artefatti prodotti

3. Attività

Il Configuration Management comprende le attività seguenti:





3.1. Configuration Identification

Una volta identificati i Configuration Item, il Project Manager, seguendo le direttive del Top Manager, assegnerà un nome all'item e lo inserirà 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, spesso situato nell'artefatto stesso.

3.2. Configuration Item

Nella categoria di Configuration Item ricadono:

- 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 PM.

Ogni Configuration Item è caratterizzato da:

- Un identificativo univoco solitamente della forma “C02_*[Acronimo Documento]*_V[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 meno rilevanti.

Una volta identificato e inserito nel CMDB, all'item viene assegnata la versione 0.1 indicante lo stato di iniziazione. Ulteriori 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 questo momento in poi ogni modifica all'item deve essere approvata dai 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.

Configuration Management Database (CMDB)

Nell'ambito del progetto Chemosmart 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. Inoltre, 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.

3.3. Configuration Control

Una volta che un Configuration Item rientra nella baseline, tutte le modifiche ad esso fatte devono seguire un preciso protocollo. Chiunque faccia parte del team può sottoporre 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, il quale, 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. Sarà poi compito dei PM accettare la modifica e 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 team members dovranno lavorare usando il sistema di branch offerto dalla piattaforma e sottoporre le proprie modifiche mediante pull request sui branch master (per le release) e develop (per lo sviluppo).

3.4. Configuration Version Release (CVR)

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

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

3.6. Configuration Audit

Al raggiungimento di una milestone, o poco prima di una consegna, verranno effettuati dei lavori di revisione da parte di tutto il team di sviluppo su ogni CI facente parte della baseline. Dopo questi lavori, i PM 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 CMDDBs

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