



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



SmartCargo

CMP

Configuration Management Plan

SmartCargo

Riferimento	C07_SMP_SmartCargo_V2.0.pdf C07_RMP_SmartCargo_V2.0.pdf
Versione	2.0
Data	20/01/2024
Destinatario	Prof.ssa Filomena Ferrucci (FF)
Presentato da	Vincenzo Esposito (VE) - 0522501385 Nicola Tortora (NT) - 0522501445
Approvato da	



Sommario

REVISION HISTORY	3
CONFIGURATION MANAGEMENT PLAN (CMP).....	4
DEL PROGETTO SMARTCARGO	4
1. INTRODUZIONE	4
1.1 AMBITO	4
1.2 SCOPO DEL DOCUMENTO.....	4
1.3 RIFERIMENTI	4
2. MANAGEMENT.....	5
2.1 FASI DEL PROGETTO.....	5
2.2 ORGANIZZAZIONE	6
2.3 RUOLI E RESPONSABILITÀ.....	6
3. ATTIVITÀ.....	7
3.1 CONFIGURATION IDENTIFICATION	7
3.2 CONFIGURATION ITEMS	8
3.3 CONFIGURATION CONTROL.....	9
3.4 CONFIGURATION VERSION RELEASE (CVR).....	9
3.5 CONFIGURATION STATUS ACCOUNTING	9
3.6 CONFIGURATION AUDIT	10



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

Revision History

Data	Versione	Descrizione	Autori
16/12/2023	1.0	Stesura sezioni 1, 2, 3	V.E. N.T.
20/01/2024	2.0	Revisione per consegna finale	V.E.



Configuration Management Plan (CMP)

del Progetto SmartCargo

1. Introduzione

1.1 Ambito

Il progetto si propone come obiettivo principale l'implementazione un sistema integrato finalizzato al potenziamento delle infrastrutture portuali, nonché al miglioramento della sicurezza ed efficienza delle operazioni di carico e scarico delle merci nel Porto di Valencia. L'obiettivo specifico è consolidare il controllo sulle attività portuali, prevenire il trasporto illegale di merci e ottimizzare i percorsi di transito per ridurre le emissioni di CO2 e mitigare l'impatto ambientale complessivo. Per conseguire tale scopo, è essenziale sviluppare una web application che integri tutte le funzionalità necessarie per agevolare il monitoraggio dei veicoli pesanti in transito nel porto, l'analisi dei percorsi con il minor impatto ambientale possibile e una comunicazione efficiente tra gli operatori della sicurezza e i trasportatori. L'attuazione di questo sistema comporterà la creazione di un ambiente portuale all'avanguardia, rendendolo attraente per le aziende e generando un aumento delle attività commerciali, consolidando così la posizione del porto nell'ambito dell'industria marittima.

1.2 Scopo del documento

Lo scopo del documento è fornire una guida dettagliata su come gestire e controllare le modifiche apportate alla configurazione di un sistema, prodotto o progetto durante l'intero ciclo di vita. Questo piano delinea le procedure, le responsabilità e gli strumenti necessari per garantire che tutte le modifiche vengano gestite in modo coerente, documentato e approvato.

1.3 Riferimenti

Di seguito sono elencati i deliverables utilizzati per la stesura del Configuration Management Plan:

- [C07_SMP_SmartCargo_V2.0.pdf](#)
- [C07_RMP_SmartCargo_V2.0.pdf](#)



2. Management

2.1 Fasi del progetto

Il processo di sviluppo di un progetto è delineato attraverso diverse fasi, ognuna delle quali svolge un ruolo fondamentale nell'assicurare il successo e l'efficienza dell'intero ciclo di vita del progetto. Queste fasi sono le seguenti:

1. **Avvio del progetto:** In questa fase iniziale, gli obiettivi del progetto vengono definiti e delineati, fornendo una base chiara per l'intero processo.
2. **Formazione e revisione delle tecnologie da utilizzare:** Questa fase è di particolare importanza per garantire che il team coinvolto sia adeguatamente formato e comprensivo delle tecnologie coinvolte nel progetto.
3. **Requirement Elicitation and Requirement Analysis:** Qui si focalizza sull'identificazione e l'analisi dei requisiti del progetto, fornendo una base solida per le fasi successive.
4. **System Design:** Questa fase si concentra sulla progettazione del sistema, determinando la sua architettura e struttura complessiva.
5. **System Test Design:** Qui viene sviluppato il design dei test di sistema, assicurando una copertura completa e accurata delle funzionalità del sistema.
6. **Object Design:** Questa fase affronta la progettazione degli oggetti, contribuendo alla realizzazione di un sistema coeso e ben strutturato.
7. **Implementazione:** Si traduce la progettazione in codice eseguibile, creando il prodotto software effettivo.
8. **Testing:** Questa fase è dedicata alla verifica e validazione del software per garantire la sua conformità ai requisiti.
9. **Rilascio del Progetto:** In questa fase, il prodotto finito viene distribuito al cliente o all'utenza prevista.
10. **Manutenzione del Progetto:** L'ultima fase del ciclo di vita del progetto si occupa delle necessità a lungo termine, inclusi eventuali miglioramenti post-implementazione. Ogni fase contribuisce in modo essenziale alla gestione sistematica e al successo duraturo del progetto.

2.2 Organizzazione

Di seguito è mostrata l'Organization Breakdown Structure del progetto SmartCargo.



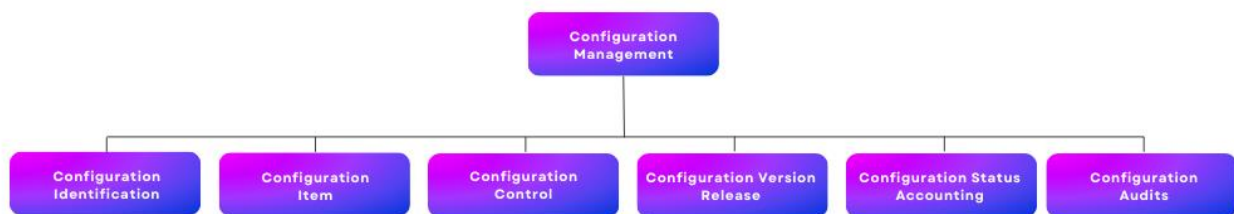
2.3 Ruoli e responsabilità

Nel contesto del progetto SmartCargo, le principali responsabilità associate al Configuration Management Plan ricadono sui seguenti ruoli: Top Manager, Stakeholder, Project Manager, Team Member e Deliverable Reviewer.

- Il **Top Manager**, in qualità di figura di alto livello, assume la responsabilità dell'accettazione e dell'approvazione dei deliverables generati nel corso del progetto.
- Gli **stakeholder**, in veste di committenti, assumono la responsabilità per l'accettazione e l'approvazione del prodotto software che sarà rilasciato.

- Il **Project Manager**, sovrintendente al contesto gestionale dell'intero progetto, si incarica della corretta gestione degli aspetti manageriali e, per quanto concerne il Configuration Management, è responsabile dell'identificazione dei Configuration Items, dell'accettazione delle richieste di modifica e dell'assegnazione delle relative responsabilità.
- Il **Team Member**, compreso ogni membro del team coinvolto nello sviluppo, assume l'incarico di eseguire i compiti assegnatigli dal Project Manager, garantendo la corretta stesura della documentazione e l'implementazione del sistema.
- Il **Deliverable Reviewer**, a sua volta identificato come Team Member, assume il ruolo di revisore per le consegne. Oltre alle attività di sviluppo e redazione della documentazione di prodotto, il Reviewer è responsabile della revisione dei paragrafi redatti da altri membri del team del progetto.

3. Attività



3.1 Configuration Identification

Il "Configuration Identification" riguarda il processo di assegnare un identificativo univoco a ciascun Configuration Item (CI) all'interno di un progetto. Un Configuration Item è un elemento identificabile e gestibile del sistema o del prodotto che deve essere soggetto a controllo di configurazione. I Project Manager, sotto la guida del Top Manager, assegneranno un nome univoco a ciascun Configuration Item e lo inseriranno nel Configuration Management Database (CMDB).

Ogni Configuration Item è caratterizzato da:

- Un identificativo univoco della forma "Year_C07_[Nome Documento]_SmartCargo_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.



3.2 Configuration Items

Sono Configuration Items:

- Documentazione relativa alla gestione del progetto
- Applicativo software e relativa documentazione
- Documentazione generale del prodotto software
- Altri documenti

Il processo di gestione dei Configuration Item (CI) prevede un sistema strutturato di assegnazione dei numeri di versione all'interno del Configuration Management Database (CMDB) al fine di tracciare le evoluzioni e le modifiche apportate a ciascun elemento di configurazione. Inizialmente, ogni Configuration Item è inserito nel CMDB con uno stato iniziale e assegnato al team di sviluppo, ricevendo un numero di versione come "1.0". Questo numero di versione rappresenta la condizione iniziale dell'elemento prima di qualsiasi modifica.

Man mano che il team di sviluppo lavora sull'elemento, le modifiche successivamente apportate vanno ad incrementare solo la parte decimale del numero di versione (la "y"). Ad esempio, se vengono apportate delle modifiche, la versione potrebbe passare da "1.1" a "1.2". Questo processo consente di mantenere una tracciabilità dettagliata delle modifiche senza dover incrementare la parte intera del numero di versione.

Quando si raggiunge una milestone o si effettua una release del documento, la parte intera del numero di versione (la "x") viene incrementata. Ad esempio, se si raggiunge la prima milestone di consegna intermedia, i CI potrebbero ricevere una versione "2.0". Le successive modifiche avranno numeri di versione come "2.1", "2.2", e così via. Questo sistema di numerazione delle versioni fornisce un meccanismo chiaro e ordinato per gestire le evoluzioni degli elementi di configurazione nel corso del progetto, consentendo una tracciabilità accurata e facilitando la gestione delle release e delle milestones.

Configuration Management Database (CMDB)

Per la gestione della configurazione, vengono utilizzati due Configuration Management Database (CMDB) distinti:

- **OneDrive:** Questo CMDB è dedicato alla gestione della documentazione associata al progetto e all'applicativo software. All'interno di OneDrive, ogni documento contiene una tabella denominata "Revision History," dove vengono annotate le diverse versioni relative alle modifiche apportate.



- **GitHub:** Questo CMDB è specificamente utilizzato per la gestione del codice sorgente del progetto. In GitHub, il controllo delle versioni è affidato al sistema basato sui commit del sistema di versioning. Ogni commit rappresenta chiaramente una modifica nel codice, e vengono evidenziati in modo esplicito i commit di rilascio associati a versioni specifiche, indicando l'incremento numerico corrispondente.

3.3 Configuration Control

Tutte le modifiche apportate a un Configuration Item devono seguire una procedura ben definita. Una volta approvata una richiesta di modifica, questa viene assegnata a uno o più sviluppatori. Per quanto concerne la documentazione, dopo ogni modifica effettuata da un membro del team, la stessa dovrà essere sottoposta a una revisione iniziale da parte dell'autore prima di essere ulteriormente esaminata e approvata dal revisore designato per il documento.

Nel caso di modifiche al codice sorgente, lo sviluppatore è tenuto a garantire l'esecuzione corretta del processo di build e il superamento dei test dopo l'implementazione della change request, potendo eventualmente avvalersi dei sistemi di continuous integration forniti da GitHub Action. Successivamente, spetta al Project Manager accettare la modifica e integrarla nel progetto effettivo.

3.4 Configuration Version Release (CVR)

Le condizioni che determinano il rilascio di una nuova versione (con incremento della “y”) dei documenti o del codice sorgente sono:

- Aggiunta di sezioni all'interno dei documenti
- Aggiunta di nuovi componenti nel codice
- La risoluzione o la modifica di piccole parti della documentazione dovute ad inconsistenze
- Il fixing di uno o più bug nel codice

Viene incrementata la “x” della versione quando avviene la consegna del deliverable relativa ad una milestone.

3.5 Configuration Status Accounting

Durante il periodo intercorrente tra le milestone di consegna, i Configuration Items saranno archiviati nella cartella OneDrive (Deliverables), mentre il codice sorgente sarà conservato su GitHub. L'accounting



dello stato della configurazione avverrà ad ogni milestone. Il Project Manager sarà responsabile di creare un file compresso (zip) contenente tutti i deliverables con le versioni più recenti. Per quanto riguarda il codice sorgente, ogni modifica verrà effettuata su branch separati; una volta approvata, la modifica sarà unita (merged) con il branch principale di sviluppo. In vista di una milestone di consegna, l'intero codice sarà spostato nel branch dedicato ai rilasci.

3.6 Configuration Audit

Prima di ogni consegna o al raggiungimento di una milestone, è necessario condurre revisioni approfondite di tutte le attività svolte dal Team relative a ciascun Configuration Item. Successivamente, sarà eseguita una verifica supplementare da parte dei Project Manager.

Le principali azioni da compiere includono:

- Verificare il corretto versioning dei deliverables;
- Valutare la coerenza delle modifiche apportate;
- Analizzare la qualità degli Items;
- Assicurarsi dell'adeguata organizzazione delle Configuration Management Databases.

Qualora si ritenga opportuno, i Project Manager potranno richiedere il ripristino di determinate modifiche tramite OneDrive/GitHub.