



## Configuration Management Plan

Riferimento	2023_C10_CMP_beehAlve_V2.0
Versione	1.0
Data	02/01/2024
Destinatario	Azienda "HiveGuard Analytics"
Presentato da	Delogu Nicolò, Mazza Dario
Approvato da	



## Revision History

---

Data	Versione	Descrizione	Autori
02/01/2024	1.0	Prima stesura	Delogu Nicolò, Mazza Dario



## Team Members

Ruolo	Nome e Cognome	Acronimo	Email
PM	Delogu Nicolò	DN	<a href="mailto:n.delogu@studenti.unisa.it">n.delogu@studenti.unisa.it</a>
PM	Mazza Dario	MD	<a href="mailto:d.mazza6@studenti.unisa.it">d.mazza6@studenti.unisa.it</a>
TM	Festa Francesco	FF	<a href="mailto:f.festa19@studenti.unisa.it">f.festa19@studenti.unisa.it</a>
TM	Gallotta Nicolò	GN	<a href="mailto:n.gallotta@studenti.unisa.it">n.gallotta@studenti.unisa.it</a>
TM	Valente Sara	VS	<a href="mailto:s.valente8@studenti.unisa.it">s.valente8@studenti.unisa.it</a>
TM	De Pasquale Andrea	DA	<a href="mailto:a.depasquale10@studenti.unisa.it">a.depasquale10@studenti.unisa.it</a>
TM	Milione Lorenzo	ML	<a href="mailto:l.milione4@studenti.unisa.it">l.milione4@studenti.unisa.it</a>
TM	Boninfante Carmine	BC	<a href="mailto:c.boninfante2@studenti.unisa.it">c.boninfante2@studenti.unisa.it</a>



## Sommario

---

Revision History .....	2
Team Members.....	3
Sommario .....	4
1. Introduzione .....	5
1.1. Ambito .....	5
1.2. Scopo del Documento .....	5
1.3. Riferimenti ad altri artefatti.....	5
2. Configuration Management Approach.....	6
2.1 Fasi del Progetto .....	6
2.2 Organizzazione.....	6
2.3 Ruoli e Responsabilità .....	7
3. Attività .....	7
3.1 Configuration Identification .....	7
3.2 Configuration Item .....	8
3.3 Configuration Management Databases (CMDB) .....	8
3.4 Configuration Control.....	9
3.5 Configuration Version Release (CVR) .....	9
3.6 Configuration Status Accounting .....	10
3.7 Configuration Audits .....	10

## 1. Introduzione

---

### 1.1. Ambito

La HiveGuard Analytics si impegna a modernizzare il settore apicolo attraverso l'implementazione di tecnologie avanzate. L'azienda riconosce l'urgenza di affrontare il "Colony Collapse Disorder" (CCD), una sfida crescente che minaccia gli apicoltori globalmente. Per trattare questa problematica, è stato deciso di sviluppare una piattaforma web innovativa, dotata di una dashboard user-friendly, per consentire agli apicoltori di monitorare e gestire i loro alveari in modo efficiente.

L'iniziativa non solo mira a introdurre innovazioni tecnologiche nell'apicoltura, ma si propone anche come risposta diretta alle gravi perdite economiche e ambientali causate dal CCD. Il progetto integrerà un modulo di intelligenza artificiale che analizzerà in tempo reale i dati provenienti da dispositivi IoT, individuando anomalie e segnali precoci del CCD, in modo da consentire agli apicoltori di intervenire tempestivamente per salvaguardare la salute delle loro colonie d'api.

### 1.2. Scopo del Documento

Il presente documento si propone di fornire un modello standard per gestire in modo efficace l'identificazione, il controllo, la manutenzione e la verifica delle versioni di tutti gli elementi di configurazione (Configuration Items). Il Configuration Management Plan (CMP) è progettato, inoltre, per raggiungere diversi obiettivi chiave, tra cui:

- Mantenere l'integrità del prodotto software;
- Supportare le attività di sviluppo e manutenzione;
- Massimizzare la produttività riducendo gli errori derivanti dalla modifica degli artefatti prodotti;

### 1.3. Riferimenti ad altri artefatti

Di seguito inseriamo anche collegamenti ai due CMDB:

- [GitHub](#)
- [Google Drive](#)

## 2. Configuration Management Approach

Questa sezione del documento delinea la metodologia, i ruoli e le responsabilità, nonché l'approccio finanziario per identificare, valutare e affrontare i rischi che potrebbero influenzare il conseguimento dei nostri obiettivi.

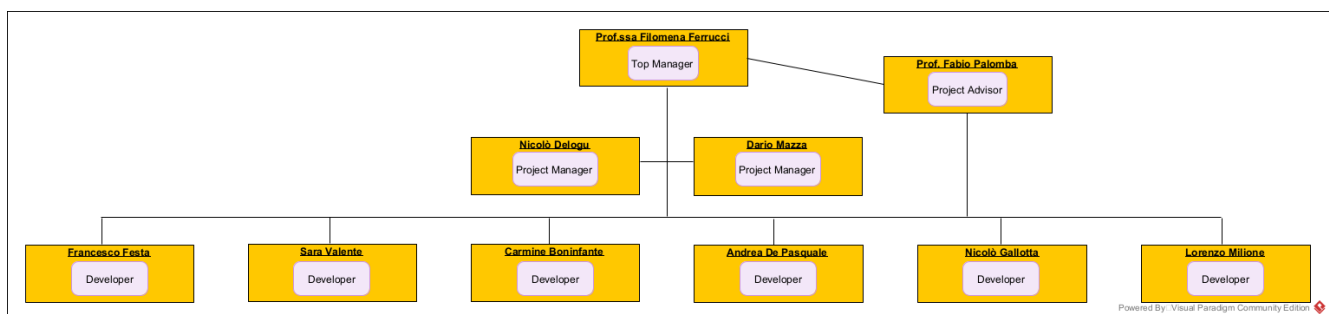
### 2.1 Fasi del Progetto

Il Sono state identificate le seguenti principali fasi del progetto:

- Avvio del progetto
- Requirements Elicitation
- Requirements Analysis
- System Design
- System Test Plan and Specification Design
- Object Design
- Source Code Implementation
- Testing
- Rilascio (creazione manuali d'installazione e d'uso)
- Manutenzione

### 2.2 Organizzazione

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



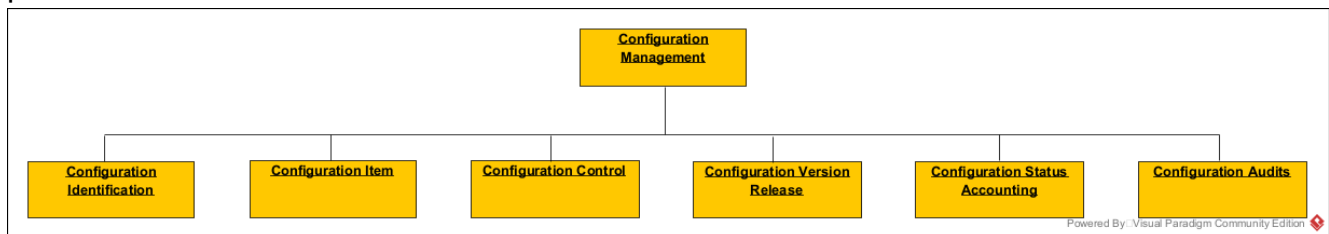
## 2.3 Ruoli e Responsabilità

Nell'ambito del progetto, la struttura organizzativa prevede diversi ruoli:

- Top Manager: responsabile dell'accettazione dei deliverables prodotti e del corretto svolgimento generale del progetto
- Project Advisor: punto di riferimento per i Developers rispetto agli artefatti e agli altri deliverables del progetto.
- Project Manager: responsabile della gestione di tutti gli aspetti manageriali del progetto. In particolare, per quanto riguarda la gestione delle configurazioni, il Project Manager è incaricato di identificare i configuration item.
- Developer: responsabile della documentazione relativa al prodotto e dell'implementazione del software. Svolge i task assegnatigli dai Project Managers, partecipando attivamente allo sviluppo del software e contribuendo alla documentazione necessaria. Il Developer inoltre controlla il proprio lavoro e analizza anche il lavoro svolto dagli altri Developers.

## 3. Attività

Di seguito, vengono riportate e descritte le attività che vengono svolte inerentemente al configuration management, il quale riguarda sia il codice sorgente che i vari altri deliverables prodotti.



### 3.1 Configuration Identification

Il Configuration Management (Gestione della Configurazione) è una pratica che si estende sia alla documentazione che ai sorgenti del progetto. Dopo l'identificazione dei Configuration Item (CI), i Project Managers, in conformità alle direttive del Top Manager, procederanno con l'assegnazione di un nome all'elemento e inseriranno le informazioni corrispondenti nel Configuration Management Database (CMDB) in uno stato iniziale.

Successivamente, l'item sarà affidato al team di sviluppo responsabile, che sarà incaricato di elaborarlo e di monitorare tutte le modifiche apportate. Il processo di tracciamento avverrà attraverso un registro con identificativo incrementale, comunemente noto come Revision History (Storico delle Revisioni), incorporato direttamente nell'artefatto stesso Configuration Item.

## 3.2 Configuration Item

I Configuration Item (CI) costituiscono una categoria diversificata di oggetti coinvolta nel Configuration Control. Gli elementi inclusi come Configuration Item sono i seguenti:

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

Ciascun Configuration Item è caratterizzato da:

- Identificativo univoco: "2023\_C10\_[Acronimo\_Documento]\_beehAlve\_V[x.y]".
- Numero di versione: si presenta come "x.y", dove x indica la consegna o il raggiungimento di una milestone y indica modifiche di minore entità.

Al momento dell'individuazione e dell'inserimento nel Configuration Management Database (CMDB), l'item viene assegnato alla versione 0.1, indicando lo stato iniziale. Le successive modifiche incrementano solo il valore di y della versione. Quando si raggiunge la prima release del documento o una milestone dedicata, la versione diventa 1.0. L'incremento di x avviene solo in caso di nuove consegne, nuove milestones o modifiche particolarmente significative.

## 3.3 Configuration Management Databases (CMDB)

Nell'ambito del progetto beehAlve, si prevede l'utilizzo di due differenti Configuration Management Database (CMDB):

- Google Drive: Questo sistema sarà impiegato per la gestione dei documenti inerenti al design dell'applicativo, tra cui RAD (Requirements Analysis Document), SDD (Software Design Document), ODD (Operation and Maintenance Documentation), documenti di testing, manuali, ecc. All'interno del CMDB Google Drive, ogni documento è identificato da un riferimento alla versione nel nome e include una tabella di Revision History. Quest'ultima consente di associare ad ogni versione le relative modifiche e i partecipanti al processo di cambiamento.
- GitHub: Questa piattaforma sarà utilizzata per la gestione del codice sorgente del sistema. GitHub offre nativamente funzionalità avanzate per il versioning del codice, consentendo al team di sviluppo di mantenere il controllo sulle variazioni apportate al software.



### 3.4 Configuration Control

Una volta che un Configuration Item è incluso nella baseline, tutte le modifiche ad esso apportate devono seguire un approccio ben definito.

Ogni Developer lavorerà su un proprio branch, assegnatogli durante la fase iniziale dello sviluppo del codice sorgente e su di esso realizzerà tutte le modifiche/funzionalità richieste dai task individuati.

L'implementazione di una modifica sarà assegnata a uno o più sviluppatori, possibilmente lo stesso individuo che l'ha identificata e, al termine dell'implementazione, il responsabile dello sviluppo si assicurerà della corretta propagazione e consistenza delle modifiche.

Nel caso di modifiche al codice, il Developer dovrà garantire il corretto build e funzionamento del progetto dopo aver incluso la modifica (dal proprio branch) nel branch "main". Inoltre il Developer avrà la responsabilità di assicurarsi che le modifiche apportate non entrino in conflitto con il lavoro svolto dagli altri membri del team.

Per quanto riguarda gli accessi, l'intero team di sviluppo avrà privilegi completi sulla cartella Google Drive, inclusi diritti di scrittura e lettura degli artefatti. Per la repository GitHub, i Developers lavoreranno utilizzando il sistema di branch offerto dalla piattaforma. Tutte le modifiche saranno apportate al contenuto del branch "main", seguendo le modalità di lavoro convenute dal team.

### 3.5 Configuration Version Release (CVR)

L'attività di management delle release del progetto beehAlve viene attuata quando si verificano specifiche condizioni che giustificano il rilascio di una nuova release. Tali condizioni includono:

- **Risoluzione di bug di piccola entità nel codice sorgente:** Quando viene risolto uno o più bug minori, ciò comporta un incremento della variabile "y" all'interno della versione del codice sorgente.
- **Modifiche o risoluzioni minori nella documentazione:** La correzione o la modifica di piccole parti della documentazione, ad esempio dovute a inconsistenze rilevate, determina l'incremento della variabile "y" all'interno della versione del documento.
- **Risoluzione di bug di notevole entità nel codice sorgente:** Nel caso di risoluzione di uno o più bug di rilevanza significativa, si procede all'aggiornamento della variabile "x" all'interno della versione del codice sorgente
- **Completamento di un documento con modifiche ingenti:** Quando viene completato un documento con l'aggiunta o la modifica di numerose sezioni, ciò comporta un aumento della variabile "x" nella versione del documento.

### 3.6 Configuration Status Accounting

Durante il periodo che intercorre tra due consegne o milestones, i Configuration Items vengono conservati nella cartella Google Drive o su GitHub, dove si svolgono le principali attività di modifica e sviluppo.

In questo contesto, la funzionalità di release di GitHub viene sfruttata per marcare e consolidare gli artefatti in un determinato stato, consentendo un'ulteriore riferimento e la possibilità di confrontare facilmente le diverse versioni.

### 3.7 Configuration Audits

Al raggiungimento di una milestone o poco prima di una consegna, il team di sviluppo condurrà un'attenta revisione su ciascun Configuration Item incluso nella baseline. Dopo questa fase di revisione, i Project Managers eseguiranno personalmente una verifica, garantendo:

- **Corretta numerazione delle versioni**, assicurandosi che le versioni degli Configuration Items siano numerate in modo corretto, rispettando gli standard stabiliti.
- **Consistenza delle modifiche tra gli items collegati**, verificando che le modifiche apportate siano coerenti tra gli Configuration Items collegati, mantenendo la congruenza tra le diverse parti del progetto.
- **Presenza di tutti gli items**, accertandosi che tutti gli Configuration Items previsti siano presenti, evitando omissioni involontarie.
- **Corretta organizzazione dei CMDBs**, verificando che i Configuration Management Database siano organizzati in modo appropriato e rispecchino la struttura stabilita per la gestione delle configurazioni.

Nel caso di necessità, i Project Managers avranno la facoltà di richiedere il rollback di alcune modifiche. Tale operazione di rollback sarà eseguita utilizzando i sistemi integrati in Google Drive e GitHub.