



# Software Project Management Plan

## *BeeHave*

|                      |                                      |
|----------------------|--------------------------------------|
| <b>Riferimento</b>   | C14_SPMP_ver1.0                      |
| <b>Versione</b>      | 1.0                                  |
| <b>Data</b>          | 20/01/2023                           |
| <b>Destinatario</b>  | Prof.ssa Filomena Ferrucci           |
| <b>Presentato da</b> | Gianmario Voria, Antonio Della Porta |



## Revision History

| Data       | Versione | Descrizione             | Autori  |
|------------|----------|-------------------------|---------|
| 09/12/2022 | 0.1      | Prima stesura           | GV, ADP |
| 15/12/2022 | 0.2      | Scrittura capitoli 1, 2 | GV      |
| 27/12/2022 | 0.4      | Scrittura capitolo 3, 4 | GV      |
| 28/12/2022 | 0.3      | Scrittura capitolo 5, 6 | ADP     |
| 16/01/2023 | 0.5      | Revisione e modifiche   | GV, ADP |
| 20/01/2023 | 1.0      | Revisione per consegna  | GV, ADP |

## Project Managers

| Nome                | Ruolo           | Acronimo | Contatto   |
|---------------------|-----------------|----------|--|
| Gianmario Voria     | Project Manager | GV       | <a href="mailto:g.voria6@studenti.unisa.it">g.voria6@studenti.unisa.it</a>             |
| Antonio Della Porta | Project Manager | ADP      | <a href="mailto:a.dellaporta26@studenti.unisa.it">a.dellaporta26@studenti.unisa.it</a> |



# Sommario

|   |    |
|---|----|
| Revision History.....                   | 2  |
| Project Managers .....                  | 2  |
| 1. Introduzione .....                   | 4  |
| 1.1. Scopo del progetto .....           | 4  |
| 1.2. Vincoli .....                      | 5  |
| 1.3. Deliverables .....                 | 5  |
| 1.4. Schedule e Costi .....             | 5  |
| 2. Riferimenti ad altri documenti ..... | 6  |
| 3. Contesto del progetto .....          | 6  |
| 3.1. Modello di processo .....          | 6  |
| 3.2. Process Improvement Plan.....      | 6  |
| 3.3. Infrastructure Plan.....           | 7  |
| 3.4. Metodi, tools, tecniche.....       | 7  |
| 3.5. Acceptance Plan .....              | 8  |
| 3.6. Organizzazione .....               | 8  |
| 4. Pianificazione.....                  | 9  |
| 4.1. Initiation .....                   | 9  |
| 4.1.1. Estimation Plan.....             | 9  |
| 4.1.2. Staffing Plan.....               | 9  |
| 4.1.3. Resource Acquisition Plan .....  | 10 |
| 4.1.4. Training Plan .....              | 10 |
| 4.2. Project Work Plans .....           | 10 |
| 4.2.1. Work Activities .....            | 10 |
| 4.2.2. Schedule Allocation .....        | 11 |
| 4.2.3. Resource Allocation .....        | 11 |
| 4.2.4. Budget Allocation .....          | 11 |
| 5. Valutazione e Controllo .....        | 11 |
| 5.1. Requirements Management Plan.....  | 11 |
| 5.2. Scope Change Control Plan .....    | 12 |
| 5.3. Schedule Control Plan .....        | 12 |
| 5.4. Quality Assurance Plan .....       | 13 |
| 5.5. Closeout Plan.....                 | 13 |
| 6. Consegna del prodotto .....          | 13 |



|  |    |
|--|----|
| 7. Processi di supporto .....                | 14 |
| 7.1. Supervisione e ambiente di lavoro ..... | 14 |
| 7.2. Decision Management.....                | 16 |
| 7.3. Risk Management.....                    | 17 |
| 7.4. Configuration Management .....          | 17 |
| 7.5. Quality Assurance.....                  | 17 |
| 7.6. Measurements .....                      | 17 |

# 1. Introduzione

## 1.1. Scopo del progetto

Gli effetti delle azioni dell'uomo hanno portato ad un progressivo peggioramento delle condizioni di sopravvivenza di molte specie animali presenti sul nostro pianeta. Tra le varie specie in via d'estinzione, particolare preoccupazione da parte della comunità è rivolta ad una delle più importanti per il nostro ecosistema: le api. Solo in Europa, in trent'anni, il numero di api si è ridotto notevolmente e la durata media della loro vita è diminuita. La scomparsa degli impollinatori, in particolare delle api, non è da sottovalutare: gran parte delle principali colture agrarie e delle piante selvatiche da fiore si riproducono grazie a questi insetti.

Per provare a contrastare il declino di queste specie, si è pensato di adottare le seguenti strategie:



- Fornire al popolo italiano una piattaforma digitale che coinvolga attivamente nel loro allevamento e sostentamento;
- Aiutare gli apicoltori ad espandersi in modo tale da incrementare il numero di api presenti sul territorio;
- Creare un processo di autofinanziamento del lavoro degli apicoltori favorendo un rapporto di duplice interesse con i fruitori della piattaforma.

## 1.2. Vincoli

Per ciò che concerne lo scope del prodotto, è necessario che possa essere inquadrato nei punti stabiliti dal recente PNRR per la transizione ecologica. È inoltre richiesto che il progetto presenti una innovazione tecnologica tangibile.

Tra i vincoli di processo stabiliti dal top management, risultano particolarmente importanti quello sul budget orario di ogni membro del team, che ammonta a 50 ore, ed i vincoli sulle scadenze.

I vincoli sui tool di management prevedono l'utilizzo di Trello e Slack per la gestione del lavoro con i team member e l'uso di Microsoft Project per i task di management. Le tecnologie sono stabilite dai PM, ed in particolare prevedono l'utilizzo di Python per la realizzazione dell'applicativo.

Per i vincoli tecnici ed altri vincoli comunicativi si rimanda allo Statement of Work

## 1.3. Deliverables

I deliverables del progetto possono essere raggruppati in due gruppi, e contengono i seguenti artefatti:

- Project Management: business case, project charter, team contract, statement of work, WBS, project management plan, cost baseline, status reports, final project presentation, final project report, lessons-learned report, e ogni altro documento richiesto per gestire il progetto.
- Di Prodotto: RAD, SDD, ODD, Matrice di Tracciabilità, Test Plan, Test Case Specification, Test incident Report, Test Summary Report, Manuale D'Uso, Manuale Installazione e ogni altro documento richiesto per lo sviluppo del sistema

## 1.4. Schedule e Costi

Di seguito è mostrato un riepilogo dello schedule e del budget del progetto ottenuto tramite il software Microsoft Project.



|     |   |       |       |              |              |            |             |      |             |
|-----|---|-------|-------|--------------|--------------|------------|-------------|------|-------------|
| 1   | 1 BeeHave                                 | 76 g? | 244 h | lun 31/10/22 | mer 25/01/23 |            | € 17.080,00 | 100% | € 17.080,00 |
| 1.1 | 1.1 Requirements Elicitation and Analysis | 23 g  | 66 h  | lun 31/10/22 | ven 25/11/22 |            | € 4.620,00  | 100% | € 4.620,00  |
| 1.2 | 1.2 System Design                         | 8 g   | 26 h  | sab 26/11/22 | sab 03/12/22 | 2          | € 1.820,00  | 100% | € 1.820,00  |
| 1.3 | 1.3 System Test Design                    | 3 g   | 12 h  | lun 05/12/22 | mer 07/12/22 | 17         | € 840,00    | 100% | € 840,00    |
| 1.4 | 1.4 Consegna intermedia                   | 0 g   | 0 h   | lun 23/01/23 | lun 23/01/23 | 17;2;30;48 | € 0,00      | 0%   | € 0,00      |
| 1.5 | 1.5 Object Design                         | 9 g   | 14 h  | mer 07/12/22 | gio 15/12/22 | 33         | € 980,00    | 100% | € 980,00    |
| 1.6 | 1.6 System Implementation and Testing     | 25 g? | 115 h | mar 27/12/22 | mar 24/01/23 | 34         | € 8.050,00  | 100% | € 8.050,00  |
| 1.7 | 1.7 Training                              | 75 g  | 7 h   | lun 31/10/22 | mar 24/01/23 |            | € 490,00    | 100% | € 490,00    |
| 1.8 | 1.8 Final Documents                       | 3 g   | 4 h   | ven 20/01/23 | lun 23/01/23 |            | € 280,00    | 100% | € 280,00    |
| 1.9 | 1.9 Consegna finale                       | 0 g   | 0 h   | mar 24/01/23 | mar 24/01/23 | 34;40      | € 0,00      | 0%   | € 0,00      |

## 2. Riferimenti ad altri documenti

## 3. Contesto del progetto

### 3.1. Modello di processo

Il modello di ciclo di vita seguito durante la realizzazione del progetto è un modello a V con retroazione. Nonostante l'assenza del cliente, i PM si sono occupati di effettuare le operazioni di Verifica e Validazione dopo ogni fase del progetto.

Durante la fase implementativa, si è effettuato uno shift verso un modello Agile, in particolare il modello Scrum. I motivi di questa scelta sono principalmente due:

- Didattici, in quanto Scrum rientra tra gli argomenti del corso di Ingegneria del Software;
- Accrescimento personale, in quanto l'esperienza di Scrum sarà sicuramente utile sia per i team member che per i project manager.

### 3.2. Process Improvement Plan

Per ciò che riguarda il miglioramento della qualità dei processi attuati, sono state effettuate le seguenti attività:

- **Monitoring:** prima di iniziare ogni attività, i project manager effettueranno un incontro con i team member per istruirli su cosa deve essere fatto e come;
- **Scrum Retrospective:** durante la fase implementativa, alla fine di ogni sprint verrà fatta una retrospective in collaborazione con tutti i team member per migliorare il lavoro già a partire dallo sprint successivo;
- **Discussione 1 ad 1 con i team member:** durante la comunicazione delle valutazioni, i PM chiederanno ad i team member di fare una analisi del lavoro svolto, col fine di migliorare i processi.



### 3.3. Infrastructure Plan

Le macchine utilizzate dai team member per lo sviluppo della piattaforma sono principalmente macchine personali sulle quali verrà installato un qualsiasi IDE per lo sviluppo in Python e Git.

### 3.4. Metodi, tools, tecniche

#### *Metodi*

Nella presente sezione verranno elencate le metodologie usate durante lo sviluppo del progetto:

- **Scrum** come modello di ciclo di vita durante la fase di implementazione;
- **Pull based development** per il lavoro collaborativo tramite sistema di version control;
- **Pair programming** per lo sviluppo dell'applicativo;
- **Code review** per garantire una migliore qualità del codice.

#### *Tools*

Nella presente sezione verranno elencati i principali tool usati durante la realizzazione del progetto.

Per ciò che riguarda la gestione del team, sono stati usati i seguenti tool:

- **Trello** per la coordinazione del lavoro e dei task;
- **Telegram** per la comunicazione informale;
- **Slack** per la comunicazione formale;
- **Discord** per i meeting da remoto.

Per lo sviluppo degli artefatti, sono stati utilizzati i seguenti tool:

- **One Drive** per la gestione delle cartelle contenenti i documenti;
- **Microsoft Word** per la scrittura dei documenti;
- **Microsoft Excel** per i fogli di lavoro;
- **Draw.io** per la creazione dei diagrammi;
- **Pencil** per la creazione dei mock-up;
- **PyCharm IDE** per lo sviluppo del software.

#### *Tecnologie*

Nella presente sezione verranno illustrate le tecnologie implementative utilizzate.

Per lo sviluppo vero e proprio, verranno usati:

- **Python** come linguaggio di programmazione della logica di business;
- **Flask** come framework per la gestione dello sviluppo web e del web server;
- **SQL-alchemy** come libreria per le interrogazioni al database;
- **PyTest** per il test di unità;



- **HTML, CSS, Bootstrap e JavaScript** per la realizzazione delle view;
- **Jinja** come engine per la creazione di pagine dinamiche con Flask;

I tool di supporto utilizzati sono i seguenti:

- **Github** per il version control;
- **Github Actions** per la Continuous Integration;
- **PyLint e SonarQube** per la verifica del codice;

### 3.5. Acceptance Plan

La prima versione del software verrà rilasciata al Top Manager durante la prima settimana successiva al 17/01/2023. Verrà preparata una presentazione che illustrerà l'approccio al management utilizzato. Sarà anche preparata una presentazione relativa a tutto il progetto, con annessa una demo del software.

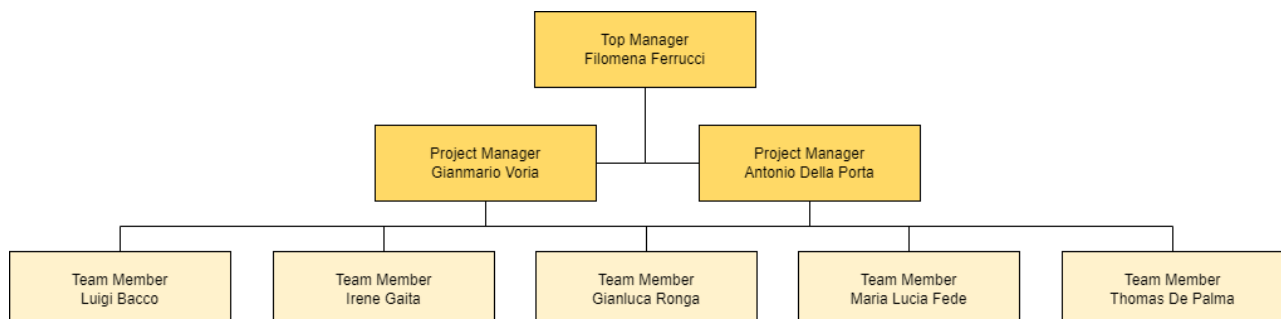
Il prodotto sarà accettato sulla base dei seguenti criteri:

- Consegna dei deliverables entro le scadenze;
- Elevata qualità di tutti gli artefatti prodotti;
- Rispetto di tutti i vincoli stabiliti dal top management.

### 3.6. Organizzazione

Nella presente sezione si andrà a descrivere la struttura dell'organizzazione del progetto. La suddetta struttura è di tipo gerarchico e prevede al suo apice il Top Manager, che si occuperà di supervisionare il progetto e comunicare con i Project Manager. I Project Manager hanno la responsabilità di assicurarsi che il progetto venga concluso con successo, gestendo i Team Member.

Di seguito è mostrato il diagramma dell'organizzazione del progetto:





## 4. Pianificazione

Di seguito si andrà a descrivere nel dettaglio come sono state condotte le fasi di avvio del progetto, di stima delle risorse e del budget, e i piani relativi alla gestione delle risorse.

### 4.1. Initiation

#### 4.1.1. Estimation Plan

Il budget a disposizione è dato dalla somma delle 50 ore previste per ogni membro del team, per un totale di 250 ore. Le ore sono state suddivise tra le varie attività necessarie allo sviluppo del sistema come segue:

| Attività                              | WBS ID | Ore Stimate |
|---------------------------------------|--------|-------------|
| Requirements Elicitation and Analysis | 1.1    | 66 h        |
| System Design                         | 1.2    | 26 h        |
| System Test Design                    | 1.3    | 12 h        |
| Object Design                         | 1.5    | 14 h        |
| Implementazione e Testing             | 1.6    | 115 h       |
| Training                              | 1.7    | 7 h         |
| Finalizzazione                        | 1.8    | 4 h         |

Non tutte le ore sono state previste nella stima, le rimanenti saranno usate come riserva.

#### 4.1.2. Staffing Plan

Il piano per la selezione del team prevede l'identificazione di persone dotate di diverse skill, sia dal punto di vista tecnico che relazionale. Si darà precedenza a soggetti con lo spiccato senso del dovere e della responsabilità, prediligendo il lato umano al lato puramente tecnico.

Le abilità di tipo tecnico riguardano prettamente conoscenze in ambito di programmazione, basi di dati e sviluppo web, che non rappresentano però un vincolo stretto. Il motivo di tale decisione è che si prevede che le skills dal punto di vista implementativo andranno a migliorare nel corso del progetto. Per questo motivo, i team members dovranno essere predisposti all'apprendimento di nuove tecnologie e flessibili al cambiamento.

#### 4.1.3. Resource Acquisition Plan

La fase di acquisizione delle risorse verrà effettuata attraverso i seguenti step:

- Gli studenti compileranno un questionario in cui evidenzieranno le loro capacità tecniche;
- I PM faranno una presentazione in cui elencheranno la lista delle skills richieste per il successo del progetto;
- Gli studenti esprimeranno la loro preferenza;
- I PM valuteranno i team member sulle base delle preferenze ricevute e sceglieranno un team.

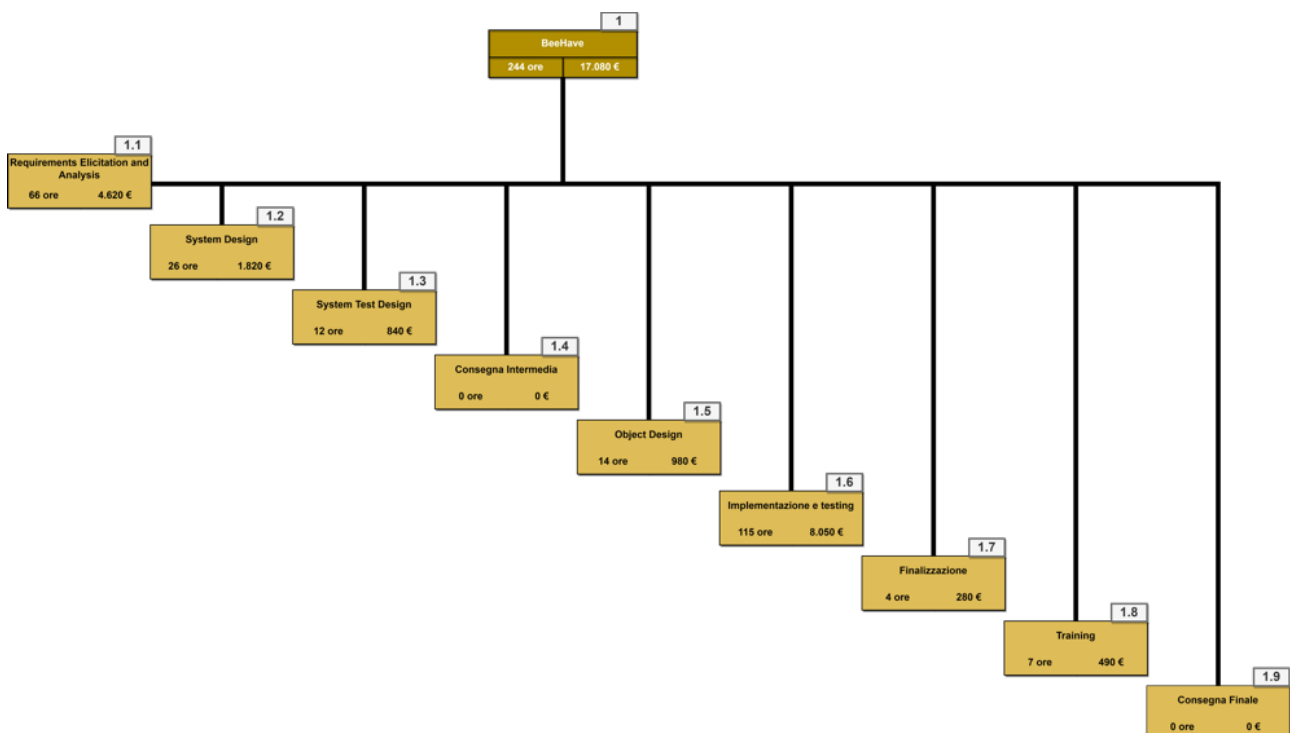
#### 4.1.4. Training Plan

Nel corso della realizzazione del progetto saranno tenute diverse sessioni di training, alcune delle quali sono state istanziate all'interno del budget e riguardano principalmente le fasi implementative e di testing. Nonostante ciò, i PM sfrutteranno i meeting formali prima dell'inizio di ciascun nuovo artefatto per spiegare con precisione il lavoro da fare.

## 4.2. Project Work Plans

#### 4.2.1. Work Activities

Di seguito è riportata una vista ad alto livello della WBS:





#### 4.2.2. Schedule Allocation

Per quanto riguarda lo schedule dei vari Work Packages, fare riferimento al file ottenuto tramite Microsoft Project allegato alla consegna.

#### 4.2.3. Resource Allocation

Per motivi didattici, per la quasi totalità della attività di progetto, la divisione dei compiti tra gli studenti sarà di tipo verticale. Tutti i team member lavoreranno parallelamente su diversi artefatti facenti parte della stessa macro-attività.

#### 4.2.4. Budget Allocation

Il budget del progetto consiste nelle ore lavorative dei team member, che verranno allocate basandosi sull'esperienza pregressa dei PM. Come già evidenziato, non tutte le ore sono state assegnate. Le restanti fanno infatti parte dei costi di riserva.

## 5. Valutazione e Controllo

Nella presente sezione si andranno a specificare le procedure necessarie per la pianificazione del controllo dei requisiti, della schedulazione, del budget, della qualità, di reporting e dell'insieme di metriche, la presentazione dei possibili rischi che possono accadere e i rispettivi piani per evitarli o minimizzarli e quelli di contingenza da attuare in caso essi si verificano.

### 5.1. Requirements Management Plan

La raccolta dei requisiti risulta essere una parte essenziale nell'ambito dello sviluppo di un buon prodotto software. La raccolta dei requisiti sarà fatta all'inizio della fase di requirements elicitation, sfruttando diagrammi degli attori, specifiche del cliente, e activity diagram. I requisiti così ottenuti saranno poi raffinati e analizzati attraverso l'uso di casi d'uso e scenari.

Una volta raggiunta una sufficiente conoscenza di essi, saranno elencati seguendo lo standard IEEE e sarà associata loro una priorità.

Tutti i requisiti a priorità elevata saranno completamente documentati e implementati nella prima versione. I requisiti a priorità media e bassa non saranno documentati, a eccezione di alcuni considerati importanti per diversi motivi.

I requisiti potranno subire modifiche previa accettazione da parte di tutto il team e non oltre la fase di implementazione, a eccezione di casi particolari.

## 5.2. Scope Change Control Plan

Durante lo sviluppo del progetto è possibile che ci sia l'esigenza di modificare, eliminare o aggiungere requisiti funzionali. Proposte di tale tipo dovranno però seguire un preciso percorso che prevede diverse fasi:

1. **Proposta di cambiamento:** un qualsiasi stakeholder può proporre una richiesta di cambiamento direttamente al Project Manager andando a strutturare tale richiesta in un documento esaustivo comprendente le motivazioni e i benefici dell'introduzione della modifica
2. **Valutazione:** i PM dovranno valutare la modifica in termini di impatto sul progetto e sulle risorse del budget.
3. **Condivisione di opinioni:** la modifica sarà valutata con l'intero team di sviluppo e ogni team member potrà esprimere la propria opinione.
4. **Accettazione o rifiuto:** la richiesta potrà essere accettata o rifiutata dai PM sulla base delle analisi effettuate;
5. **Annotazione su Board:** la modifica viene segnata su una board condivisa e il team individua assieme tutti gli artefatti che si devono modificare per implementare la change request. Una volta che tutti gli artefatti sono stati modificati, la card è sarà chiusa e la modifica sarà considerata implementata.

## 5.3. Schedule Control Plan

La pianificazione dello schedule avviene prevalentemente attraverso l'uso di Microsoft Project 2019 il quale implementa in modo automatico numerose funzioni, tra le quali il calcolo dello earned value usabile per calcolare numerose metriche, tra le quali l'indice di performance dello schedule (SPI) e l'indice di performance dei costi (CPI).

Durante il corso del progetto, i PM aggiorneranno manualmente i progressi tramite il software, ottenendo tali metriche e prendendo decisioni sulla base di esse.

Oltre quanto precedentemente detto, dopo l'assegnazione dei singoli tasks i PM terranno traccia dello stato tramite l'uso di una board Trello, andando a fissare scadenze per ognuno di essi. Con cadenza settimanale, saranno effettuati meeting formali comprendenti di una parte di status sui tasks in corso per la milestone.



## 5.4. Quality Assurance Plan

I processi di quality assurance sono riportati nel dettaglio nel documento di Quality Management Plan

## 5.5. Closeout Plan

L'intero prodotto sarà consegnato al cliente a terminazione del progetto. Sarà consegnata l'intera documentazione attraverso la piattaforma e-learning, le valutazioni fatte dai PM ai team member e sarà fatta una presentazione sul progetto sia dai PM che dai team member per illustrare l'intero progetto e le attività svolte.

# 6. Consegna del prodotto

Nella presente sezione si andranno a elencare tutti i deliverables di progetto, distinguendo tra deliverables di management e deliverables di sviluppo.

Di seguito la lista dei deliverables di management:

- [Statement Of Work](#) (SOW)
- [Business Case](#) (BC)
- Software Project Management Plan (SPMP)
- [Quality Management Plan](#) (QMP)
- [Risk Management Plan](#) (RMP)
- [Risk Register](#)
- [WBS Dictionary](#) (WBSD)
- [Configuration Management Plan](#) (CMP)
- [Cost Baseline](#) (CB)
- [Quality Management Plan](#) (QMP)

Di seguito una lista dei deliverables di prodotto

- [Requirements Analysis Document](#) (RAD)
- [System Design Document](#) (SDD)
- [Test Plan](#) (TP)
- [Test Case Specification](#) (TCS)



- [Test Incident Report](#) (TIR)
- [Test Summary Report](#) (TSR)
- [Object Design Document](#) (ODD)
- Codice sorgente
- [Matrice di tracciabilità](#)
- Documenti di revisione
- [Foglio con ore di lavoro](#)
- Agende dei meeting
- Minute dei meeting

## 7. Processi di supporto

Nella presente sezione si andranno a descrivere i piani di supporto per il successo del progetto.

### 7.1. Supervisione e ambiente di lavoro

Durante il corso della settimana, ogni team member lavorerà, eventualmente in gruppo, a un task previsto per quella stessa settimana e appartenente ad una delle attività di progetto. Una volta a settimana, durante il meeting formale, ogni team member dovrà aggiornare i PM sullo status dei task assegnatigli.

Oltre ciò, due volte a settimana, sarà organizzata una giornata di lavoro di gruppo durante la quale tutti i team members lavoreranno assieme su uno dei canali predisposti per il lavoro. In tali occasioni anche i PM parteciperanno al lavoro, supervisionandolo e aiutando se necessario.

Al raggiungimento di una milestone, la revisione degli artefatti sarà assegnata all'intero team e, una volta conclusa, i PM prenderanno in carico il documento e lo visioneranno per individuare eventuali errori da correggere. Ogni consegna, sarà responsabilità del Top Manager eseguire un ulteriore controllo a scopo valutativo.

La valutazione del lavoro svolto per i team members si baserà su una serie di parametri valutativi scelti dai PM e concordati col team di sviluppo durante la firma del Team Contract. Di seguito si riporta una lista di tali parametri valutativi:



| Nome                              | Descrizione Campo   | Significato valutazione minima  | Significato valutazione media   | Significato valutazione massima  |
|-----------------------------------|---|---|---|--|
| <b>Qualità del lavoro</b>         | Qualità del contenuto degli artefatti, correttezza grammaticale, rispetto delle linee guida per la documentazione, rispetto delle regole di buona programmazione. | Lo studente produce artefatti di pessima qualità o non li produce affatto                       | Lo studente produce artefatti di qualità soddisfacente ma spesso non rispetta quanto definito precedentemente | Lo studente produce artefatti di ottima qualità e rispetta tutto quello che viene detto e deciso |
| <b>Comportamento</b>              | Rispetto dei propri compagni di gruppo, rispetto dei ruoli stabiliti all'interno del team.  | Lo studente si comporta in modo pessimo, mancando di rispetto e causando danni ai suoi compagni | Lo studente si comporta in un modo accettabile, senza dare problemi ai colleghi                               | Lo studente è impeccabile e si comporta in modo educato e disponibile verso i suoi compagni.     |
| <b>Rispetto del Team Contract</b> | Rispetto di quanto stabilito nel team contract, in particolare del Code of Conduct.   | Lo studente viola continuamente quanto espresso nel team contract                               | Lo studente viola un numero medio di volte quanto espresso nel team contract                                  | Lo studente rispetta sempre il team contract   |
| <b>Rispetto delle scadenze</b>    | Rispetto delle scadenze assegnate dai PM, sia quelle principali che quelle relative ai task minori.   | Lo studente consegna gli artefatti con largo ritardo continuamente                              | Lo studente consegna gli artefatti con largo ritardo un numero medio di volte                                 | Lo studente consegna gli artefatti rispettando i tempi previsti                                  |

|                                 |  |   |  |   |
|---------------------------------|--|---|--|---|
| <b>Capacità di relazionarsi</b> | Capacità di comunicare con gli altri team member e con i PM, capacità di aiutare nella risoluzione di problemi relazionali, capacità di adattarsi alle decisioni del team. | Incapacità di comunicare e interagire con gli altri membri del team e coi PM.                             | Interagisce con il team per la risoluzione di alcune problematiche | Ottime capacità comunicative con tutto il team        |
| <b>Efficienza</b>               | Proattività, produttività.   | Si agisce senza la minima capacità di iniziativa e non si partecipa alla scrittura di tutti gli artefatti | Si agisce cercando di evitare problematiche e ovvie                | Si agisce pensando sempre al futuro e di conseguenza  |
| <b>Partecipazione</b>           | Interesse e partecipazione alle attività del team, proattività nella risoluzione dei problemi.   | Lo studente non partecipa ad alcuna attività di gruppo  | Lo studente partecipa alle attività in modo apatico                | Lo studente partecipa attivamente a tutte le attività |

## 7.2. Decision Management

Per ciò che concerne le decisioni, risulta essere doverosa una distinzione tra decisioni in ambito di management del team di sviluppo e decisioni che riguardano lo sviluppo del prodotto.

Le prime, comprendenti principalmente decisioni sullo schedule, sui rischi da gestire, sull'assegnazione di ruoli di spicco, sul budget, e sulle attività formali, saranno prese dai Project Manager e comunicate ai team member che potranno esporre una loro opinione eventualmente usata per modificare quanto deciso.

Le seconde, comprendenti decisioni sull'assegnazione dei tasks, sulle funzionalità del prodotto, sulle priorità degli artefatti e degli elementi riguardanti il prodotto, e altro, saranno prese da tutto il gruppo durante i meeting formali e le riunioni di gruppo. Solitamente tale processo avviene con la presentazione da parte dei PM di una serie di decisioni da prendere e la discussione di tutto il gruppo a riguardo.





Un tipico processo di decision making prevede:

1. Formalizzazione del problema e sua enunciazione sotto forma di domanda
2. Proposta di una serie di alternative atte a risolvere il problema;
3. Valutazione di pro e contro o di eventuali ostacoli per ogni alternativa;
4. Votazione di una soluzione.

In caso di stallo su una decisione, i Project Manager avranno sempre il potere di prenderne una arbitrariamente, sebbene tale scelta dovrà essere correttamente esposta e supportata onde evitare malcontenti nel gruppo di progetto.

Eventuali proposte potranno essere presentate anche direttamente dai team member e subiranno il medesimo iter valutativo delle altre.

### 7.3. Risk Management

Per ciò che concerne la Risk Management, si rimanda al documento di [Risk Management Plan](#).

### 7.4. Configuration Management

Per ciò che concerne la Configuration Management, si rimanda al documento di [Configuration Management Plan](#).

### 7.5. Quality Assurance

Per ciò che concerne la Quality Assurance, si rimanda al documento di [Configuration Management Plan](#).

### 7.6. Measurements

Per ciò che concerne le metriche da calcolare durante l'esecuzione del progetto, occorre fare una distinzione tra metriche di qualità e di dimensione e complessità.

Le prime sono esaustivamente esposte nel QMP.

Le seconde sono esposte di seguito:



| Nome metrica                                | Descrizione  | Motivazione   |
|---|--|---|
| <b>Average Cyclomatic Complexity Number</b> | Metrica per il calcolo del numero di decisioni del programma.                      | Usata per stabilire la complessità del programma.                       |
| <b>Functions Count</b>                      | Numero di funzioni create.   | Metrica dimensionale per caratterizzare la dimensione dell'applicativo. |
| <b>Warning Count</b>                        | Numero di warning rilevati.  | Metrica utile per avere informazioni sulla qualità del codice.          |
| <b>Average Tokens Count</b>                 | Numero di Token (inteso come parole chiave riconoscibili dall'interprete) rilevate | Metrica utile per avere informazioni sulla qualità del codice.          |