



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



Software Project Management Plan Environmental Intelligence for Agriculture

Riferimento	
Versione	1.0
Data	09/01/2022
Destinatario	Prof.ssa Filomena Ferrucci
Presentato da	Carminc Laudato, Pierluigi Lambiase
Approvato da	



Revision History

Data	Versione	Descrizione	Autori
15/12/2022	0.1	Capitolo 1,2	Carmine Laudato Pierluigi Lambiase
17/12/2022	0.2	Capitolo 3	Carmine Laudato Pierluigi Lambiase
18/12/2022	0.4	Capitolo 4,5	Carmine Laudato Pierluigi Lambiase
28/12/2022	0.6	Capitolo 6	Carmine Laudato Pierluigi Lambiase
30/12/2022	0.7	Capitolo 7	Carmine Laudato Pierluigi Lambiase
03/01/2023	0.8	Capitolo 8	Carmine Laudato Pierluigi Lambiase
09/01/2023	1.0	Revisione di tutto il documento	Carmine Laudato Pierluigi Lambiase



Sommario

Revision History	2
1. Panoramica del progetto	5
1.1. Riepilogo del progetto	5
1.1.1. Scopo, ambito e obiettivi.....	5
1.1.2. Assunzioni e vincoli.....	5
1.1.3. Deliverable del progetto.....	5
1.1.4. Riepilogo delle schedule e del budget.....	6
2. Riferimenti ad altri documenti.....	7
3. Contesto del progetto.....	7
3.1. Modello di processo	7
3.2. Process Improvement Plan	7
3.3. Infrastrutture Plan	8
3.4. Metodi, tool e tecniche.....	8
3.5. Product Acceptance Plan	9
3.6. Organizzazione del progetto.....	9
4. Pianificazione del progetto	10
4.1. Avvio del progetto	10
4.1.1. Estimation Plan	10
4.1.2. Staffing Plan.....	10
4.1.3. Resource Acquisition Plan	10
4.1.4. Training Plan	11
4.2. Project Works Plan.....	11
4.2.1. Work activities	11
4.2.2. Schedule Allocation	11
4.2.3. Resource Allocation	12
4.2.4. Budget Allocation	12
5. Valutazione e controllo del progetto.....	12
5.1. Requirements Management Plan	12
5.2. Scope Change Control Plan.....	13
5.3. Schedule Control Plan.....	13
5.4. Budget Control Plan	13
5.5. Quality Assurance Plan	13
5.6. Project Closeout Plan.....	14



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

6. Consegna del prodotto	14
7. Supporting Process Plan	15
7.1. Supervisione e ambiente di lavoro del progetto	15
7.2. Decision Management	15
7.3. Risk Management	16
7.4. Configuration Management	16
7.5. Quality Assurance	16
7.6. Misure	16
8. Piani aggiuntivi	17



1. Panoramica del progetto

1.1. Riepilogo del progetto

L'obiettivo del prodotto è fornire strumenti di supporto alle attività agroindustriali atte a garantire un'efficiente gestione delle risorse idriche e un tracciamento del livello di inquinamento ed esposizione ambientale della coltivazione. Deve supportare:

- la localizzazione e gestione dei vari terreni;
- le previsioni microclimatiche e delle precipitazioni per i luoghi di interesse provenienti da sorgenti open;
- MODULO FIA: sviluppo di un modulo di intelligenza artificiale per supportare l'agricoltore nella fase decisionale tramite gli insight della piattaforma;
- informazioni inerenti ai livelli di inquinamento dei luoghi di interesse;
- una fase di analisi delle previsioni meteo con lo scopo di gestire gli impianti di irrigazione;
- una fase di analisi dell'esposizione ambientale delle varie coltivazioni.

1.1.1. Scopo, ambito e obiettivi

Lo staff di IAgricoltura punta a fornire ai propri clienti prodotti di supporto alla gestione delle attività agroindustriali, con lo scopo di ottimizzare le varie attività dal punto di vista economico e ridurre l'impatto ambientale. Per questo motivo nasce EnIA (Environmental Intelligence for Agriculture): prodotto software atto a migliorare la gestione dei terreni, a rilevare l'inquinamento assorbito dalle varie colture durante il loro ciclo di vita e automatizzare gli impianti di irrigazione, utilizzando i dati relativi ai cambiamenti microclimatici.

IAgricoltura, con Enia, punta a raggiungere i seguenti obiettivi:

- ridurre al minimo i consumi delle risorse idriche, diminuendo al contempo l'impatto ambientale;
- fornire un tracciamento del livello di inquinamento ed esposizione ambientale della coltivazione;
- localizzazione e gestione dei vari terreni.

1.1.2. Assunzioni e vincoli

Il sistema proposto deve facilitare la gestione dei terreni e dei processi agroindustriali al fine di ridurre l'impatto ambientale e le spese relative all'irrigazione. Deve inoltre semplificare il processo di monitoraggio del livello di assorbimento, da parte delle colture, dei vari inquinanti. Il Project Manager dovrà gestire e monitorare l'intero processo di sviluppo, e il team di progetto dovrà includere studenti del corso triennale di Ingegneria del Software dell'Università di Salerno.

1.1.3. Deliverable del progetto

Viene riportata di seguito la tabella con i deliverables previsti dal progetto.



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

Deliverables	WBS ID	Responsabile	Completamento pianificato
RAD	1.2.3	Team	25/11/2022
SDD	1.3.4	Team	07/12/2022
TP e TCS	1.4.3	Team	09/12/2022
Consegna intermedia	1.6	Team	13/12/2022
ODD	1.7.3	Team	29/12/2022
Test Incident report e Test Summary report	1.9.4	Team	08/02/2023
Consegna finale	1.10	Team	14/02/2023

1.1.4. Riepilogo delle schedule e del budget

Task Name	Durata	Inizio	Fine	Predecessori	WBS	Num. Risorse	Numero di Ore per ogni risorsa	Num. Ore Totali	Costo Totale
EnIA	82 giorni	ven 28/10/22	mar 14/02/23		1	5	50	250	7,500.00 €
Training	2 giorni	ven 28/10/22	lun 31/10/22		1.1	5	2	10	300.00 €
Requirements Elicitation and Analysis	19 giorni	mar 01/11/22	ven 25/11/22		1.2	5	9.4	47	1,410.00 €
System Design	9 giorni	ven 25/11/22	mer 07/12/22	3	1.3	5	4.5	27.5	825.00 €
System Test Design	4 giorni	mer 07/12/22	sab 10/12/22	18	1.4	5	2.2	11	330.00 €
Revisione Finale RAD, SDD, STD	3 giorni	ven 09/12/22	lun 12/12/22		1.5	3	2	6	180.00 €
Consegna intermedia	1 giorno	mar 13/12/22	mar 13/12/22	3,18,35	1.6	0	0	0	0.00 €
Object Design	4.4 giorni	ven 23/12/22	gio 29/12/22	40	1.7	5	3.7	18.5	555.00 €
System Implementation	27 giorni	lun 02/01/23	mar 07/02/23	41	1.8	5	22.2	111	3,330.00 €
Testing	6 giorni	mar 07/02/23	mar 14/02/23	55	1.9	5	3.8	19	570.00 €
Consegna finale	0 giorni	mar 14/02/23	mar 14/02/23	41,52,58	1.10	0	0	0	0.00 €

Il progetto EnIA prevede lo sviluppo di un software di gestione terreni e irrigazione supportato da intelligenza artificiale. La data di consegna è prevista entro la fine di gennaio 2023 e il costo stimato per lo sviluppo è di 15000€, che comprende anche 2500€ di spese generali. Inoltre, si stima un costo annuale di 2000€ per la manutenzione, per un totale di 21000€ in tre anni. Il prodotto sarà disponibile in due versioni: Standard e Premium. La versione Standard include tutte le funzionalità di base di EnIA, mentre la versione Premium include anche le funzionalità di monitoraggio ambientale delle coltivazioni. Le versioni saranno vendute con licenza annuale al prezzo di 1500€ per la versione Standard e 2000€ per la versione Premium. Secondo un'analisi di mercato, si prevede di vendere tre unità della versione Standard e due unità della versione Premium durante il primo anno, con un aumento di due unità per la versione Premium durante il secondo anno e un aumento di un'unità per la versione Standard e di due unità per la versione Premium durante il terzo anno. I guadagni previsti per i primi tre anni saranno di 8500€, 12550€ e 18000€ rispettivamente. L'analisi finanziaria si basa su tre fattori chiave: il Net Present Value, il Return On Investment e il Payback.

Per maggiori dettagli, visualizzare:

- Business Case;



- Schedule Management Plan.

2. Riferimenti ad altri documenti

Documenti di management a cui si farà riferimento:

- Business Case.
- Risk Management Plan.
- Schedule Management Plan.
- Configuration Management Plan.
- Quality Management Plan.
- Software Project Management Plan.

3. Contesto del progetto

In questa sezione viene descritto il modello di processo utilizzato per lo sviluppo del progetto software, inclusa la pianificazione del progresso del processo, la pianificazione dell'infrastruttura necessaria per lo sviluppo, i metodi, gli strumenti e le tecniche utilizzati, la pianificazione dell'accettazione del prodotto e l'organizzazione del progetto.

3.1. Modello di processo

Nel progetto in questione è stato utilizzato il modello di sviluppo "waterfall con retroazione" durante la fase di design e l'approccio Agile "Scrum" con elementi di Kanban e Extreme Programming (XP) durante la fase di implementazione. Sono stati utilizzati gli elementi del metodo Scrum come Sprint, pianificazione degli Sprint, Daily Scrum Meeting, Sprint Review e sessioni di retrospettiva di sprint, e una Kanban Board per aiutare il team a organizzare il lavoro. Inoltre, è stata utilizzata la pratica di pair programming, che fa parte del modello Extreme Programming. Si è scelto di utilizzare questi modelli Agile per gestire meglio lo sviluppo del progetto software e ottenere dei feedback tempestivi dal cliente.

3.2. Process Improvement Plan

Per migliorare la qualità dei processi del progetto, sono state pianificate alcune attività, come il training per istruire i team members su ciò che deve essere fatto e come farlo, e le retrospective di sprint, che sono analisi a posteriori di ogni fase di sviluppo basate su questionari e riunioni formali per raccogliere informazioni che possono essere utilizzate per migliorare il processo. Queste informazioni saranno conservate in documenti separati dal resto del progetto.



3.3. Infrastrutture Plan

Per configurare l'ambiente di sviluppo del prodotto, i team members utilizzeranno principalmente le loro macchine personali su cui verranno installate una Integrated Development Environment (IDE) per lo sviluppo, come Visual Studio Code, e uno strumento di versioning locale come Git; inoltre, il database MongoDB verrà hostato su una macchina gratuita fornita dal servizio MongoDB Atlas per garantire l'univocità delle varie versioni del database e dei dati presenti all'interno. Queste scelte sono state fatte per motivi didattici e per ovviare al problema della distanza dei vari team member e dei project manager. Prima dell'inizio della fase di implementazione, verrà organizzato un meeting ad hoc per impostare queste macchine.

3.4. Metodi, tool e tecniche

➤ Metodi

- Durante la fase di implementazione e sviluppo del progetto, verranno utilizzate le seguenti metodologie: Scrum come modello di ciclo di vita per lo sviluppo, pair programming per lo sviluppo di alcune parti dell'applicazione, pull based development per gestire l'avanzamento del lavoro e revisioni per garantire la qualità del codice.

➤ Tool

- In questa sezione vengono elencati i principali strumenti utilizzati durante tutto il ciclo di vita del progetto EnIA e le tecnologie di sviluppo utilizzate.
- Per quanto riguarda l'organizzazione del lavoro del team, verranno utilizzati i seguenti strumenti:
 - Slack, per la comunicazione formale tra tutto il team;
 - Telegram, per la comunicazione informale;
 - Discord, per le sessioni di lavoro dei Team Member;
 - Trello, per la gestione e il coordinamento delle varie task e della Kanban Board per la fase di implementazione.
- Per quanto riguarda lo sviluppo degli artefatti, verranno utilizzati i seguenti tool:
 - One Drive, per lo sviluppo simultaneo e lo storage dei vari artefatti;
 - Microsoft Word, per la stesura dei documenti;
 - Microsoft Excel, per la gestione dei fogli di lavoro;
 - Microsoft Project, per la gestione di tutto il progetto;
 - Draw.io, per la creazione dei diagrammi UML;
 - Figma, per la creazione dei mockups delle interfacce grafiche;
 - Visual Studio Code, per lo sviluppo del codice sorgente.
- Per sviluppare correttamente il software, verranno utilizzate le seguenti tecnologie:
 - Python, come linguaggio di programmazione back-end;



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

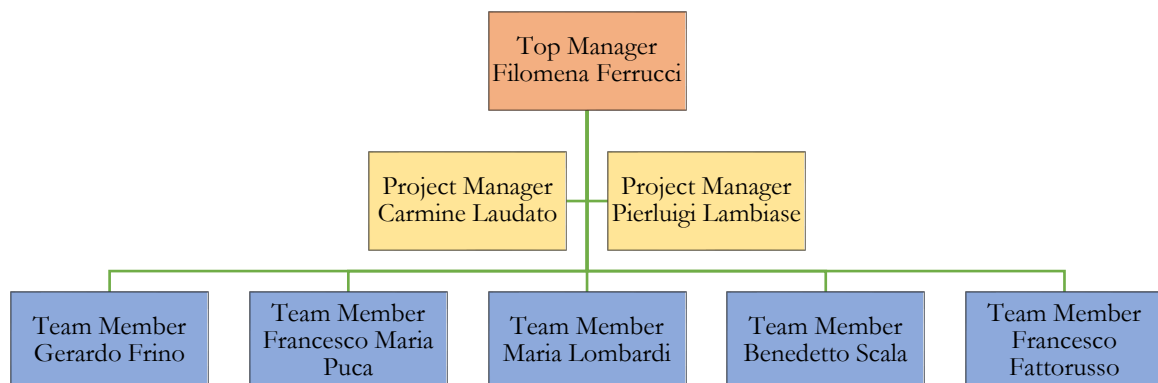
- PyUnit, come framework per il test di unità e integrazione;
 - HTML5, come linguaggio di programmazione front-end;
 - CSS, come linguaggio di styling;
 - Javascript, come linguaggio di scripting per il front-end;
 - Bootstrap, come framework per lo sviluppo front-end.
- Per il deployment del software, le verifiche e le build saranno utilizzati:
 - Flask, per il building e il testing del software;
 - Github, per il versioning;
 - PyCodeStyle, per il rispetto della convenzione Pep8 di Python.

3.5. Product Acceptance Plan

La prima versione rilasciata del prodotto verrà consegnata e mostrata al Top Manager la settimana successiva al 3 febbraio 2023. La consegna includerà una presentazione sull'approccio di gestione e sviluppo del progetto e sui principali artefatti prodotti, oltre a una breve demo per validare alcune funzionalità dell'applicazione. Il prodotto sarà accettato in base a criteri quali la consegna dei deliverable di progetto entro le date stabilite dal Top Manager, la consegna dell'applicazione con le funzionalità a priorità elevata entro la data di scadenza del progetto, un'elevata qualità di tutti gli artefatti prodotti, dimostrata attraverso l'utilizzo di check-list di qualità e strumenti per il calcolo delle metriche del codice, una copertura del ramo dei test dell'applicazione superiore al 75% calcolata tramite l'utilizzo dello strumento coverage.py, e un numero di errori emessi da Pep8, calcolati in base al rispetto della convenzione Pep8 di Python, inferiore a 20.

3.6. Organizzazione del progetto

La struttura dell'organizzazione del progetto è gerarchica, con il Top Manager in cima alla gerarchia che supervisiona il progetto e comunica con il Project Manager. Il Project Manager ha il compito di portare a termine il progetto con successo, gestendo i team members che rispondono direttamente a lui. L'organigramma dell'organizzazione è mostrato di seguito.





4. Pianificazione del progetto

Il piano di progetto prevede la stima delle risorse necessarie, il reclutamento del team, l'acquisto delle risorse e il training del team. In particolare, verranno esaminati i costi e i tempi previsti per completare il progetto, verrà selezionato il team giusto per l'esecuzione del lavoro, verranno acquistate le risorse necessarie (ad esempio, attrezzature, software, materiali) e verrà organizzato un programma di formazione per garantire che il team sia pronto a lavorare in modo efficiente.

4.1. Avvio del progetto

4.1.1. Estimation Plan

Il budget a disposizione per il progetto è di 250 ore per il team e 50 ore per ogni Project Manager, per un totale di 350 ore. Queste ore sono state suddivise tra le diverse macro-attività del progetto EnIA, tenendo conto del tempo necessario per completarle. Il budget delle ore sarà utilizzato per garantire che il progetto venga portato a termine in modo efficiente e che il team abbia il tempo sufficiente per svolgere il lavoro assegnato.

Di seguito le macro-attività del progetto:

- Training: 10h;
- Requirements Elicitation and Analysis: 47h.
- System Design: 27,5h.
- System Test Design: 11h.
- Consegna intermedia: 6h.
- Object Design: 18.5h.
- System Implementation: 111h.
- Testing: 19h.

4.1.2. Staffing Plan

Il piano per la selezione del team di progetto prevede l'identificazione di soggetti con diverse abilità, sia di tipo tecnico che comportamentale. Si cercherà di ottenere figure che possano compensarsi a vicenda, ma che allo stesso tempo possiedano alcune abilità comuni, come la predisposizione all'apprendimento di nuove tecnologie e la flessibilità al cambiamento. In particolare, si cercano figure con conoscenze in programmazione orientata agli oggetti, costruzione di basi di dati e schermate web, e competenze ingegneristiche nella modellazione, formalizzazione e risoluzione di problemi. Si prevede che tali abilità possano essere migliorate durante il progetto.

4.1.3. Resource Acquisition Plan

Il processo di selezione del team per il progetto prevede alcuni passaggi:



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

1. Gli studenti triennali compilano un questionario per valutare le loro capacità tecniche e le loro esperienze precedenti;
2. Il Project Manager fa una presentazione delle proprie idee e della propria persona a tutti gli studenti triennali;
3. Gli studenti formano dei gruppi autonomi ed esprimono le loro preferenze per un certo numero di Project Manager;
4. I Project Manager esaminano le preferenze e scelgono un gruppo, risolvendo eventuali conflitti o selezionando singoli membri da gruppi divisi. In questo modo, il team viene selezionato in base alle capacità tecniche e alle esperienze precedenti degli studenti e alle preferenze espresse nei confronti dei Project Manager.

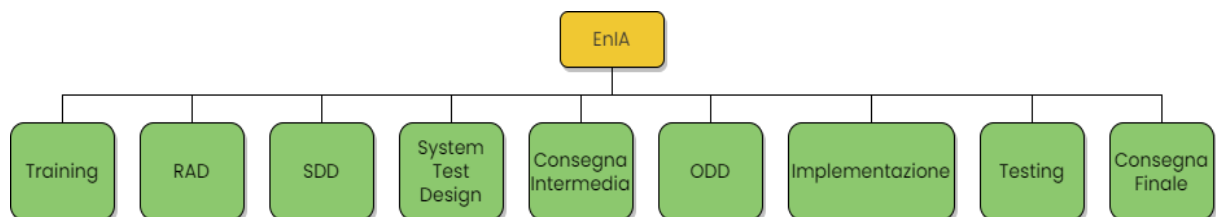
4.1.4. Training Plan

Il piano di training per il progetto prevede sessioni di formazione organizzate prima di ogni inizio di macro-attività per facilitare il lavoro del team. Il Project Manager può anche decidere di organizzare sessioni di training aggiuntive se necessario. Inoltre, i team members possono concordare ulteriori sessioni di training con il Project Manager a loro discrezione. Le principali attività di training previste per il progetto includono la formazione su Python, Flask, Bootstrap, PyUnit e MongoDB.

4.2. Project Works Plan

4.2.1. Work activities

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



4.2.2. Schedule Allocation

Il piano di gestione delle attività (Schedule Management Plan) e i file ottenuti tramite Microsoft Project 2021 descrivono lo schedule delle attività del progetto. Questi documenti e file forniscono informazioni dettagliate sulla sequenza e il timing delle attività, nonché sui rischi e le opportunità associate ad ognuna di esse. Lo schedule delle attività è un elemento importante del piano di progetto e aiuta a gestire il lavoro in modo efficiente e a rispettare i tempi previsti.



4.2.3. Resource Allocation

Per la maggior parte del progetto, il team sarà suddiviso in base a compiti specifici, con ogni membro che lavora parallelamente su diverse attività all'interno della stessa macro-attività. Questo approccio "verticale" al lavoro consente a tutti i membri del team di lavorare su diverse parti del progetto e di acquisire conoscenze e competenze specifiche. Questo tipo di suddivisione dei compiti è stato scelto anche per ragioni didattiche, in modo che ogni membro della squadra possa imparare cose nuove e sviluppare le proprie capacità.

4.2.4. Budget Allocation

Il budget del progetto consiste principalmente delle ore di lavoro del gruppo. L'assegnazione di queste ore è basata sull'esperienza precedente dei membri del gruppo e sul numero di artefatti da realizzare per ogni macro-attività. Si prevede che le attività che richiederanno più risorse saranno la raccolta dei requisiti e la fase di implementazione. Nel documento di schedule è possibile vedere come i costi sono stati assegnati alle diverse attività. Si noti che non tutte le ore di progetto (250) sono state assegnate: il resto fa parte dei costi di riserva, compresi il contingency plan (per la fase di training, che è difficile da prevedere) e le riserve di gestione (per situazioni straordinarie e non previste).

5. Valutazione e controllo del progetto

La sezione descrive le procedure per la pianificazione del controllo dei requisiti, della schedulazione, del budget, della qualità, del reporting e dell'insieme di metriche. Viene inoltre presentata una lista dei possibili rischi del progetto e i piani per evitarli o minimizzarli, nonché i piani di contingenza da attuare in caso di emergenza. Queste procedure sono essenziali per garantire il successo del progetto e per gestire eventuali imprevisti che possono verificarsi durante lo svolgimento delle attività.

5.1. Requirements Management Plan

La raccolta dei requisiti è una fase cruciale nello sviluppo di un prodotto software di qualità. I requisiti vengono raccolti all'inizio della fase di raccolta dei requisiti utilizzando diagrammi degli attori, specifiche del cliente e diagrammi di attività. I requisiti ottenuti vengono poi perfezionati e analizzati utilizzando casi d'uso e scenari. Una volta compresi sufficientemente, vengono elencati seguendo lo standard IEEE e vengono assegnate loro una priorità. I requisiti ad alta priorità verranno completamente documentati e implementati nella prima versione. I requisiti a priorità media e bassa non verranno documentati, a meno che non siano considerati importanti per altre ragioni. I requisiti possono essere modificati solo previa accettazione da parte di tutto il team e prima della fase di implementazione, ad eccezione di casi particolari.



5.2. Scope Change Control Plan

Durante lo sviluppo del progetto, può essere necessario modificare, eliminare o aggiungere requisiti funzionali. Tuttavia, per procedere con queste modifiche, è necessario seguire un preciso processo che prevede diverse fasi:

1. proposta di cambiamento (qualsiasi stakeholder può proporre una richiesta di modifica al Project Manager, specificando le motivazioni e i benefici della modifica);
2. valutazione (il PM valuta l'impatto della modifica sul progetto e sul budget);
3. condivisione di opinioni (la modifica viene discussa con il team di sviluppo e ogni membro del team può esprimere la propria opinione);
4. accettazione o rifiuto (il PM decide se accettare o rifiutare la richiesta sulla base delle valutazioni effettuate);
5. annotazione sulla board (la modifica viene segnata su una board condivisa e il team individua gli artefatti che devono essere modificati per implementare la richiesta di modifica. Una volta che gli artefatti sono stati modificati, la card viene chiusa e la modifica viene considerata implementata).

5.3. Schedule Control Plan

La pianificazione dello schedule consiste nell'organizzare le attività del progetto in modo da ottenere il risultato desiderato nel modo più efficiente possibile. Per fare ciò, viene solitamente utilizzato un software come Microsoft Project, che offre diverse funzionalità utili per la pianificazione, come il calcolo dello earned value (che permette di calcolare metriche come l'indice di performance dello schedule (SPI) e l'indice di performance dei costi (CPI)). Durante il progetto, il Project Manager (PM) aggiorna manualmente i progressi del progetto utilizzando il software e prende decisioni sulla base delle metriche ottenute. Inoltre, il PM tiene traccia dello stato di avanzamento dei tasks utilizzando una board su Trello e fissando scadenze per ognuno di essi. Infine, vengono organizzate riunioni formali con cadenza settimanale per fare il punto sui tasks in corso e sui progressi verso le milestones previste.

5.4. Budget Control Plan

Per il controllo dei costi e del budget del progetto, si utilizzeranno gli strumenti offerti da Microsoft Project. Questo software permette di calcolare l'Earned Value, che a sua volta può essere utilizzato per ottenere metriche come l'indice di performance dei costi (CPI). Il PM prenderà decisioni sulla base di queste metriche, al fine di mantenere il budget del progetto sotto controllo.

5.5. Quality Assurance Plan

I processi di quality assurance (QA) sono le attività pianificate e sistematiche attraverso cui si verifica che il prodotto software soddisfi i requisiti stabiliti. Questi processi sono descritti in dettaglio nel documento di Quality Management Plan (QMP), che specifica come saranno effettuate le verifiche della qualità durante lo sviluppo del



progetto e come verranno gestiti i problemi di qualità riscontrati. Il QMP include anche le responsabilità del team di progetto per il controllo della qualità e le procedure da seguire per garantire che il prodotto finale soddisfi gli standard di qualità stabiliti.

5.6. Project Closeout Plan

Al termine del progetto, il prodotto finale sarà consegnato al cliente insieme alla documentazione completa. Questa documentazione verrà fornita attraverso la piattaforma di e-learning e includerà le valutazioni dei PM sui team member. Inoltre, il progetto verrà presentato sia dai PM che dai team member, al fine di illustrare tutte le attività svolte e il prodotto finale.

6. Consegna del prodotto

I deliverables di progetto sono tutti i risultati e i prodotti che il team di progetto deve consegnare al cliente o al sponsor del progetto. I deliverables di management includono tutti i documenti di gestione del progetto, come il piano di progetto, il piano di gestione delle risorse, il piano di gestione della qualità e il piano di gestione dei rischi. I deliverables di sviluppo, invece, sono i prodotti finali del progetto, come il software o il sistema che viene sviluppato. I deliverables di sviluppo possono includere anche documentazione tecnica, manuali dell'utente e altra documentazione relativa al prodotto finale.

Di seguito la lista dei deliverables di management:

- Statement of Work (SOW);
- Business Case (BC);
- Project Charter;
- Quality Management Plan (QMP);
- Risk Management Plan (RMP);
- Risk Register;
- Schedule Management Plan (SMP);
- Configuration Management Plan (CMP);
- Software Project Management Plan (SPMP);
- WBS Dictionary (WBSD);
- Documentazione di Scrum.

Di seguito la lista dei deliverables di sviluppo:

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



- Matrice di tracciabilità;
- Object Design Document (ODD);
- Codice sorgente;
- Test Incident Report (TIR);
- Test Summary Report (TSR);
- Foglio con ore di lavoro;
- Agende e minute delle riunioni;

7. Supporting Process Plan

7.1. Supervisione e ambiente di lavoro del progetto

durante il corso del progetto ci saranno riunioni settimanali in cui ogni membro del team dovrà aggiornare i PM sullo stato dei propri task. Ci saranno anche giornate di lavoro di gruppo in cui tutti i membri del team lavoreranno insieme sotto la supervisione dei PM. I PM leggeranno gli artefatti prodotti ogni settimana per aumentarne la qualità e aiutare i membri del team a comprendere gli standard di documentazione. Al raggiungimento di una milestone, la revisione degli artefatti sarà assegnata a due membri del team e, una volta conclusa, sarà responsabilità dei PM visionare il documento per individuare eventuali errori da correggere. Il Top Manager eseguirà un ulteriore controllo valutativo di ogni consegna. Il principale ambiente di lavoro sarà online (Microsoft Teams e Discord) e la supervisione del team sarà affidata ai PM, mentre la supervisione dei PM sarà affidata al Top Manager tramite meeting occasionali.

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:

Parametro	Scala di valutazione
Rispetto del team contract	Valutazione su base decimale
Partecipazione attiva ai meeting (idee, proposte, elementi costruttivi)	Valutazione su scala da 1 a 5
Numero di revisioni necessarie	Valutazione basata sul numero di revisioni necessarie, da 1 a 5
Autonomia	Valutazione su scala da 1 a 5

7.2. Decision Management

In sintesi, ci sono due tipi di decisioni che devono essere prese durante il progetto: quelle relative al management del team di sviluppo e quelle relative allo sviluppo del prodotto. Le prime sono prese dai soli PM e possono essere discusse con i membri del team, mentre le seconde sono prese dal gruppo durante le riunioni formali. Il processo



di prendere una decisione comprende la formalizzazione del problema, la proposta di alternative, la valutazione delle pro e contro di ogni alternativa e la votazione di una soluzione. In caso di stallo su una decisione, i PM hanno il potere di prenderne una arbitrariamente, ma devono supportare la loro scelta per evitare malcontento nel gruppo. Le proposte possono essere presentate anche dai membri della squadra e saranno sottoposte allo stesso processo di valutazione delle altre.

7.3. Risk Management

Per tutte le informazioni sulla Risk Management, si rimanda al documento di Risk Management Plan.

7.4. Configuration Management

Per tutte le informazioni sulla Configuration Management, si rimanda al documento Configuration Management Plan.

7.5. Quality Assurance

Per tutte le informazioni sulla Quality Management, si rimanda al documento di Quality Management Plan.

7.6. Misure

Le metriche di qualità sono quelle utilizzate per valutare la qualità del prodotto o del processo di sviluppo del progetto. Le prime sono descritte in dettaglio nel Quality Management Plan. Le seconde, invece, sono:

Nome Metrica	Descrizione	Motivazione
Lines of Code (LOC)	Numero totale di linee di codice	Metrica dimensionale per determinare la dimensione dell'intero progetto software.
Non-Comment Line of Code (NCLOC)	Numero totale di linee di codice effettivo	Metrica dimensionale per determinare la dimensione del progetto software effettivo.
Comment Lines of Code (CLOC)	Numero totale di linee commentate	Metrica dimensionale per determinare la dimensione delle linee commentate del progetto.

Nel progetto, oltre alle metriche di qualità, si è cercato di misurare la produttività del team attraverso l'uso di burndown charts durante la fase di implementazione. Questi diagrammi sono stati utilizzati seguendo un approccio Scrum e tengono conto del Focus Factor (che è alto all'inizio e diminuisce progressivamente) per stimare il carico di lavoro e i tempi di completamento. I documenti relativi a queste metriche possono essere trovati nella cartella Scrum.



8. Piani aggiuntivi

Al termine del progetto, è prevista una riunione per un'attività di lessons learned più approfondita, che coinvolgerà tutti i membri del gruppo. Prima della consegna finale, sarà svolta un'attività preliminare di lessons learned, che sarà utilizzata dai PM per arricchire la presentazione finale del progetto.