



Laurea Triennale in Informatica - Università di Salerno
Corso di *Ingegneria del Software* - Prof.ssa F.Ferrucci e Prof. F.Palomba



Software Project Management Plan

GreenLeaf

Riferimento	
Versione	1.0
Data	11/02/2023
Destinatario	Prof.ssa Filomena Ferrucci
Presentato da	Angelo Afeltra, Antonio Giametta, Raffaele Squillante
Approvato da	



Sommario

Revision History	4
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 Deliverables di progetto	6
1.1.4 Riepilogo dello schedule e del budget	7
1.2 Evoluzione dello SPMP	7
2 Riferimenti ad altri documenti	8
3 Contesto del progetto	8
3.1 Modello di processo.....	8
3.2 Process Improvement Plan	9
3.3 Infrastructure Plan	9
3.4 Metodi, tool e tecniche	10
3.5 Product Acceptance Plan.....	11
3.6 Organizzazione del progetto	12
4 Pianificazione del progetto	13
4.1 Avvio del progetto.....	13
4.1.1 Estimation Plan.....	13
4.1.2 Staffing Plan.....	13
4.1.3 Resource Acquisition Plan.....	14
4.1.4 Training Plan	14
4.2 Project Works Plans.....	15
4.2.1 Work activities.....	15
4.2.2 Schedule Allocation	15



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

4.2.3	Resource Allocation	15
4.2.4	Budget Allocation	16
5	Valutazione e controllo del progetto	16
5.1	Requirements Management Plan.....	16
5.2	Scope Change Control Plan	17
5.3	Schedule Control Plan.....	17
5.4	Budget Control Plan	17
5.5	Quality Assurance Plan	18
5.6	Project Closeout Plan.....	18
6	Consegna del prodotto	18
7	Supporting Process Plans	19
7.1	Supervisione e ambiente di lavoro del progetto	19
7.2	Decision Management.....	21
7.3	Risk Management	22
7.4	Configuration Management.....	22
7.5	Quality Assurance	22
7.6	Misure	22
8	Piani aggiuntivi	23



Revision History

Data	Versione	Descrizione	Autori
05/01/2023	0.1	Prima stesura	A. Afeltra A. Giametta R. Squillante
11/02/2023	1.0	Revisione Pre Consegna	A. Afeltra A. Giametta R. Squillante



1 Panoramica del progetto

1.1 Riepilogo del progetto

1.1.1 Scopo, ambito e obiettivi

Sviluppare una piattaforma che incentivi il pubblico a combattere la piaga delle deforestazioni, tema che viene gestito dall'Agenzia Europea dell'Ambiente.

Il sistema mira ad offrire un servizio user-friendly per l'adozione, e quindi la successiva piantumazione, di alberi nelle varie regioni italiane, con opportunità futura di essere scalato a tutta la comunità europea.

Inoltre, al fine di incentivare e coinvolgere in primis gli utenti più sensibili al problema, ma in maniera indiretta anche loro amici o parenti, il sistema permetterà anche di regalare alberi adottati ad uno o più dei suddetti.

Green Leaf fornirà, inoltre, un dispositivo IoT per ogni albero adottato al fine sia di potenziare il monitoraggio dell'area, attualmente basato su stazioni fisse che non coprono in maniera dettagliata e peculiare il territorio, e sia di monitorare lo stato di salute e di crescita dell'albero stesso.

Grazie a questi innovativi monitoraggi offerti, sarà inoltre possibile offrire un servizio di previsione della situazione ambientale nel futuro.

Al fine di sensibilizzare il più possibile il pubblico, verranno forniti servizi per calcolare la CO2 causata con l'utilizzo dei mezzi di trasporto, e verranno forniti supporti formativi per avvicinarli alla pratica della piantumazione.

1.1.2 Assunzioni e vincoli

Non essendoci un precedente sistema software, non vi sono particolari vincoli e assunzioni riguardanti il prodotto.

Nell'ambito del processo, al contrario, i vincoli principali si individuano nelle ore di lavoro a disposizione per team member, 50 per ognuno, e dalla scadenze fissate dal Top Manager. Tali dati saranno riportati nel dettaglio in seguito nel documento.



Per quanto riguarda vincoli sulle tecnologie e i tools, si assume l'uso di strumenti di management quali Trello e Microsoft Project. Oltre ciò, sia assume che lo sviluppo dell'applicativo avverrà tramite l'uso di tecnologie web, nella fattispecie utilizzando Three Tier, Maven, e HTML5.

1.1.3 Deliverables di progetto

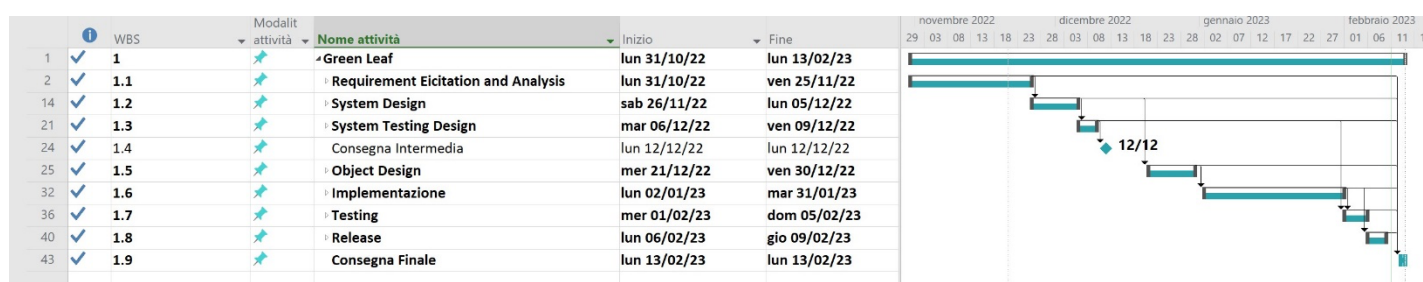
Di seguito si riporta una tabella contenente tutti i deliverables previsti per il progetto GreenLeaf.

Deliverables / Milestones	WBS id	Responsabile	Completamento pianificato
RAD	1.1	Tutto il team	25/11/2022
SDD	1.2	Tutto il team	05/12/2020
TP e TCS	1.3	Tutto il team	09/12/2022
Consegna intermedia	1.4	Tutto il team	12/12/2022
ODD	1.5	Tutto il team	30/12/2022
Applicativo, Javadoc	1.6	Tutto il team	31/01/2023
Test Incident report e Test Summary report	1.7	Tutto il team	05/02/2023
Release	1.8	Tutto il team	09/02/2023
Consegna finale	1.9	Tutto il team	13/02/2023



1.1.4 Riepilogo dello schedule e del budget

Di seguito si riporta un riepilogo dello schedule e del budget ad alto livello attraverso il Gantt Chart ottenuto tramite Microsoft Project. Non sono riportate le risorse umane essendo che il lavoro è stato separato verticalmente, per motivi didattici.



Per maggiori sullo schedule si rimanda ai documenti:

- [Schedule Management Plan](#);
- [Business Case](#):

1.2 Evoluzione dello SPMP

Il presente documento sarà aggiornato in caso di necessità. L'aggiornamento avverrà seguendo le regole previste nel documento [Configuration Management Plan](#).



2 Riferimenti ad altri documenti

Di seguito una lista agli altri documenti di management:

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

3 Contesto del progetto

Nella presente sezione vengono descritti il modello del processo adottato per lo sviluppo del progetto software in questione, la pianificazione del progresso del processo, la pianificazione dell'infrastruttura necessaria per lo sviluppo del progetto, i metodi, gli strumenti e le tecniche necessarie all'avanzamento, la pianificazione dell'accettazione del prodotto, e l'organizzazione del progetto.

3.1 Modello di processo

Il modello di ciclo di vita scelto per il progetto GreenLeaf è un modello a V con possibilità di retroazione. Tale modello consiste in un classico modello a cascata con maggior focus sulle fasi di design dei test. Tale scelta è stata presa in virtù di due principali caratteristiche del contesto di lavoro:

- L'impossibilità di avere un confronto reale e costante col cliente non presente nel progetto in questione, per motivi didattici.
- La necessità, sempre didattica, di produrre documentazione di test sin dalle prime fasi del processo.

Nonostante la mancanza della figura del cliente, ogni conclusione di fase si è comunque seguito un processo di Verifica, atto a garantire che il prodotto che si sta costruendo rispetti la specifica data, e di Validazione, atto a stabilire se il prodotto che si sta costruendo adempia ai suoi reali obiettivi operativi.

Sempre durante lo sviluppo sarà seguito un processo di Continuous Integration e Deployment (CI/CD) tramite l'uso di tools appositi e riconosciuti nel mondo professionale. Motivo di tale scelta risulta essere



principalmente la necessità di automatizzare processi di verifica del codice prodotto, onde evitare problemi inaspettati e di impatto elevato sul progetto durante le fasi di implementazione più concitate.

Per ciò che concerne il configuration management si fa riferimento al documento di [Configuration Management Plan](#).

3.2 Process Improvement Plan

Per ciò che riguarda le attività atte a migliorare la qualità dei processi stessi, esse possono essere elencate di seguito:

- **Training:** al fine di migliorare e velocizzare i processi di design e sviluppo, ogni fase del ciclo di vita sarà preceduta da una fase di training organizzata dal PM con l'obiettivo di istruire i team members su cosa deve essere fatto e come.
- **Questionari di feedback:** al fine di raccogliere opinioni e dati sulla qualità dei processi percepita dai team members, saranno organizzati questionari di feedback settimanali alla fine di ogni settimana di lavoro. Tali questionari avranno lo scopo di valutare il processo, valutare i tools usati per il management del processo, e portare i team members ad auto-valutarsi, stimolando in loro un pensiero auto-critico. I PM potranno usare le informazioni ottenute per apportare cambiamenti in corso d'opera.

3.3 Infrastructure Plan

Nella presente sezione si andrà a descrivere come avverrà la configurazione dell'ambiente di sviluppo del prodotto. Nello specifico, le macchine usate dai team members saranno principalmente macchine personali sulle quali verrà installata una IDE per lo sviluppo, preferibilmente IntelliJ, e uno strumento di versioning locale, Git. Il setup di tali macchine avverrà in un meeting ad hoc organizzato prima dell'inizio della fase di implementazione.



3.4 Metodi, tool e tecniche

Metodi

Nella presente sezione si andranno a elencare le metodologie adottate durante la fase di implementazione e sviluppo. Nello specifico, saranno utilizzate le seguenti metodologie:

- Modello a cascata, come modello di ciclo di vita per lo sviluppo;
- Pair Programming, per lo sviluppo di alcune parti dell'applicativo;
- Pull based development, per la gestione dell'avanzamento del lavoro;
- Revisioni, per lo accounting della qualità del codice.

Tools e tecnologie

Nella presente sezione si andranno a elencare i principali tools usati durante l'intero ciclo di vita del progetto GreenLeaf e le tecnologie per lo sviluppo.

Per ciò che concerne l'organizzazione del lavoro del team saranno utilizzati i seguenti tools:

- Trello, per la gestione e coordinazione riguardanti i tasks di progetto e per l'implementazione;
- Slack, per la comunicazione formale soprattutto dal PM verso gli altri team members;
- Whatsapp, per la comunicazione informale di tutto il team di progetto;
- Discord, per il lavoro di gruppo e i meeting formali.;

Per ciò che concerne lo sviluppo di artefatti, saranno utilizzati i seguenti tools:

- One Drive, per la gestione delle cartelle contenenti i documenti di progetto;
- Microsoft Word, per la scrittura dei documenti;
- Microsoft Excel, per i fogli di lavoro;
- Draw.io per la creazione dei diagrammi UML;
- Adobe XD, per la creazione dei mockups delle interfacce grafiche;
- IntelliJ IDEA Ultimate Edition, per lo sviluppo del codice sorgente tramite IDE;



- Katalon, per il testing di sistema;

Per ciò che concerne lo sviluppo del software, saranno utilizzate le seguenti tecnologie:

- Java, come linguaggio di programmazione back-end;
- JUnit, come framework per il test di unità e integrazione;
- Mockito, come framework per il mocking per il test di unità;
- HTML5, come linguaggio di programmazione front-end;
- CSS, come linguaggio di styling del front-end;
- Javascript, come linguaggio di scripting per il front-end;
- Bootstrap, come framework front-end per lo sviluppo di view grafiche.

Per ciò che concerne il build, le verifiche, e il deployment del software, saranno utilizzate i seguenti tools:

- Maven, per il building e il testing del software;
- GitHub, per il versioning;
- GitHub Action, per la Continuous Integration;
- Jacoco, per il calcolo delle metriche;
- Checkstyle, per il rispetto della convenzione Google java Style.

3.5 Product Acceptance Plan

La prima versione release del prodotto sarà consegnata e mostrata al Top Manager nella settimana dopo il 13/02/2023. Tale consegna prevederà una presentazione sull'approccio di management e sviluppo dell'intero progetto e sui principali artefatti prodotti. Oltre ciò, sarà mostrata una breve demo atta a validare alcune funzionalità dell'applicativo.

Il prodotto in questione sarà accettato sulla base dei seguenti criteri:

- Consegna dei deliverable di progetto entro le date fissate dal Top Manager;
- Consegna dell'applicativo con le funzionalità a priorità elevata entro la data ultima del progetto;
- Elevata qualità di tutti gli artefatti prodotti, dimostrata attraverso l'uso di check-list di qualità e



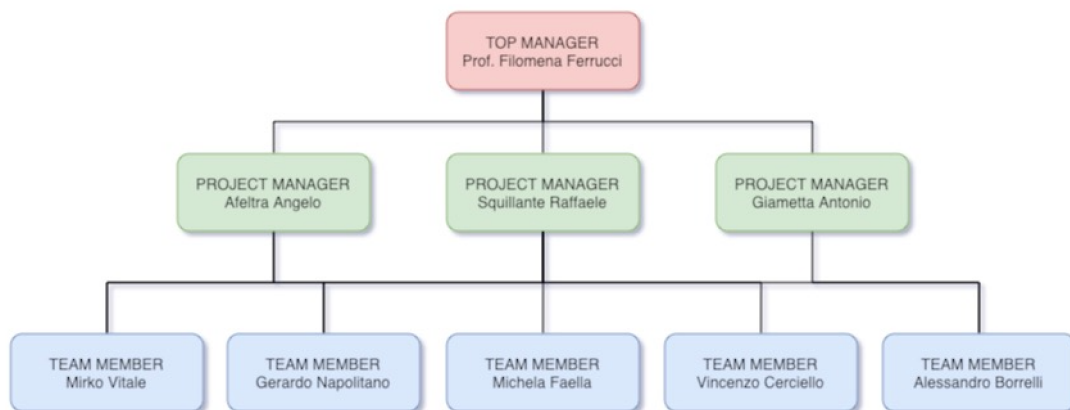
tools per il calcolo di metriche sul codice;

- Numero di warning dati in output da Checkstyle, calcolati in base al rispetto della convenzione Google di Java, inferiore a 20.

3.6 Organizzazione del progetto

Nella presente sezione si andrà a descrivere la struttura dell'organizzazione del progetto. Tale struttura è di tipo gerarchico e vede al suo apice il Top Manager, che si occupa di supervisionare il progetto e comunicare con i Project Manager. I Project Manager hanno la responsabilità di portare il progetto a compimento con successo, facendo lavoro di management nei confronti dei team members, che rispondono direttamente a loro. Tra i team member poi vi è la figura del review leader, che ha il compito di eseguire revisioni periodiche di tutti i documenti e filtrare richieste di modifiche importanti verso il project manager.

Di seguito, l'organigramma dell'organizzazione.





4 Pianificazione del progetto

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 Avvio del progetto

Di seguito si andranno a descrivere i piani per la stima, il reclutamento del team, l'acquisto delle risorse, e il training del team di progetto.

4.1.1 Estimation Plan

Il budget a disposizione risulta essere di 50 ore a team member per un totale di 250 ore a cui vanno ad aggiungersi le 50 ore dei Project Manager. Di seguito, si mostrano come tali ore sono state stimate in relazione alle diverse macro-attività del progetto GreenLeaf:

- Requirements Elicitation and Analysis: 60h;
- System Design: 25h;
- Object Design: 30h;
- System Implementation: 100h;
- Testing: 15h.

4.1.2 Staffing Plan

Il piano per la selezione del team di progetto prevede l'identificazione di soggetti con abilità di varia natura. Consapevoli del fatto che non sia possibile avere un team eccellente in ogni campo, si cercherà di ottenere figure che possano compensarsi a vicenda. Ciò nonostante, prettamente di tipo comportamentale, dovranno essere possedute da tutto il team. Per i presenti motivi, tutte le abilità richieste dal PM saranno presentate nell'introduzione dello stesso agli studenti triennali.

Le abilità di tipo tecnico riguardano prettamente conoscenze in ambito di programmazione orientata agli oggetti e costruzione di basi di dati e schermate web. Oltre ciò, vengono richieste competenza ingegneristiche nella modellazione, formalizzazione, e risoluzione di problemi. Ciò nonostante, si prevede che tali abilità andranno a migliorare nel corso del progetto. Per concludere, tutti i team members dovranno essere predisposti all'apprendimento di nuove tecnologie e flessibili al cambiamento.



Di seguito, si mostra una tabella contenente le abilità richieste con relativa importanza:

Abilità	Importanza (da 1 a 5)
Predisposizione all'apprendimento di nuove tecnologie	5
Flessibilità al cambiamento	5
Rispetto e serietà	5
Capacità di lavorare in gruppo	4
Formalizzazione di problemi	4
Risoluzione di problemi	4
Programmazione orientata agli oggetti	5
Progettazione e sviluppo di DB relazionali	4
Programmazione Front-End	4

4.1.3 Resource Acquisition Plan

L'acquisizione dei membri del gruppo avverrà in passi successivi:

1. Gli studenti triennali compileranno un questionario sulle loro capacità tecniche e esperienze passate;
2. I Project Manager faranno una presentazione delle proprie idee e della propria persona a tutti gli studenti triennali;
3. Gli studenti esprimeranno le proprie preferenze su un certo numero di Project Manager;
4. I Project Manager visioneranno le preferenze e sceglieranno un gruppo, resolvendo eventuali conflitti tra loro o prendendo singoli membri da gruppi divisi.

4.1.4 Training Plan

Nel corso del progetto saranno organizzate sessioni di training precedenti ogni inizio di macro-attività atte a facilitare il lavoro dei team members. Se necessario, il PM potrà decidere di organizzare sessioni di training addizionali e non pianificate. Altri training ancora potrebbero essere concordati tra i team member e il PM a discrezione dei primi.



Di seguito sono riportate le principali attività di training previste per il progetto:

- Training sui tools di management come Trello e Slack;
- Training sugli artefatti del RAD e la loro utilità;
- Training sugli artefatti dello SDD e la loro utilità;
- Training sul testing in particolare Category Partition;
- Training sul test di unità e integrazione;
- Training su Git e Github.

4.2 Project Works Plans

Di seguito si andranno a descrivere i piani riguardanti lo schedule e il budget di progetto.

4.2.1 Work activities

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



Per una vista più dettagliata della WBS e dei work packages si rimanda ai documenti di WBS Dictionary [C09_WBSD_v1.0](#) e [C09_SMP_v1.0](#).

4.2.2 Schedule Allocation

Lo schedule delle attività è descritto nei documenti di schedule ([Schedule Management Plan](#)) e nei files ottenuti tramite Microsoft Project 2019 (allegati 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 prevalentemente delle ore di lavoro per team member. L'assegnazione di tali ore è decisa basandosi su esperienze pregresse e numero di artefatti per macro-attività. Nel dettaglio, si stima che le attività richiedenti maggior numero di risorse siano fare la requirements analysis, perché è la prima affrontata dal team e la fase implementativa.

Nel documento relativo allo schedule è possibile vedere come i costi sono stati assegnati alle varie attività.

5 Valutazione e controllo del progetto

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 verifichino.

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 dal 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, il PM aggiornerà 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 il PM terrà 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 milestones.

5.4 Budget Control Plan



Anche per il controllo dei costi e del budget si farà uso degli strumenti forniti da Microsoft Project 2019. Anche nel presente caso, attraverso il calcolo dello Earned Value, si andranno a calcolare metriche quali l'indice di performance dei costi (CPI) e altre relative al costo. Sempre sulla base di tali metriche, saranno prese decisioni da parte del PM.

5.5 Quality Assurance Plan

I processi di quality assurance sono riportati nel dettaglio nel documento di [Quality Management Plan](#).

5.6 Project 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);
- Project Charter (PC);
- Software Project Management Plan (SPMP);
- Quality Management Plan (QMP);
- Risk Management Plan (RMP);
- Risk Register;
- WBS Dictionary (WBSD);
- Schedule Management Plan (SMP);
- Configuration Management Plan (CMP);



Di seguito la lista dei deliverables di sviluppo:

- 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 di meeting;
- Minute di meeting;

7 Supporting Process Plans

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

7.1 Supervisione e ambiente di lavoro del progetto

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. Oltre ciò, i PM leggeranno gli artefatti prodotti ogni settimana, almeno nelle prime fasi di progetto, per aumentare la qualità e aiutare i team members a comprendere bene gli standard fissati per la documentazione. Al raggiungimento



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

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.

Durante il corso del progetto, gli ambienti di lavoro principali saranno le aule studio dell'università ed i canali online come Discord. Durante tali lavori, la supervisione del team members sarà affidata ai Project Manager. La supervisione dei Project Manager spetterà al Top Manager tramite meeting occasionali.

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

Nome campo	Descrizione campo	Significato valutazione minima	Significato valutazione media	Significato valutazione massima
Proattività	Capacità di anticipare il cambiamento e le problematiche.	Si agisce senza la minima capacità di iniziativa	Si agisce cercando di evitare problematiche ovvie	Si agisce pensando sempre al futuro e di conseguenza
Partecipazione	Interesse e partecipazione alle attività di gruppo e, più nello specifico, di problem solving.	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
Qualità	Qualità del contenuto degli artefatti prodotti.	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
	Correttezza formale e grammaticale degli artefatti prodotti.			
	Numero di correzioni necessarie.			
	Capacità di lavorare in team.	Lo studente si	Lo studente si	Lo studente è
	Capacità di rispettare le decisioni.			



Relazioni	Numero di interventi costruttivi.	comporta in modo pessimo, mancando di rispetto e causando danni ai suoi compagni	comporta in un modo accettabile, senza dare problemi ai colleghi	impeccabile e si comporta in modo educato e disponibile verso i suoi compagni.
	Rispetto dei ruoli.			
	Favorire un ambiente di lavoro sano.			
Rispetto	Rispetto di quanto scritto nel team contract.	Lo studente non rispetta ad alcuna direttiva prefissata dai PM e dal team.	Lo studente rispetta in modo blando le direttive prefissate dai PM e dal team.	Lo studente rispetta con vigore tutte le direttive prefissate dai PM e dal team
	Rispettare le scadenze fissate.			
	Rispetto sull'utilizzo degli strumenti.			
	Rispetto delle date fissate per i meeting.			

In conclusione, si lascia il link al foglio di lavoro usato per le valutazioni: [C09_Valutazioni.xlsx](#).

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 soli Project Manager e comunicate ai team members 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 del 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 members 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 Qualità Assurance, si rimanda al documento di [Quality Management Plan](#).

7.6 Misure

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 documento di [Quality Management Plan](#).

Le seconde sono esposte di seguito.

Nome metrica	Descrizione	Motivazione
Lines of Code LOC	Numero totale di linee di codice.	Metrica dimensionale per caratterizzare la dimensione dell'applicativo
Non <u>Comment</u> LOC NCLOC	Numero totale di linee di codice effettivo	Metrica dimensionale per caratterizzare la dimensione dell'applicativo



Comment Lines of Code CLOC	Numero totale di linee di commenti	Metrica dimensionale per caratterizzare la dimensione dell'applicativo
Cosmic Function Point COSMIC	Metrica funzionale per il calcolo della dimensione dei requisiti funzionali	Metrica usata per stabilire la dimensione del sistema dal punto di vista delle funzionalità
Complessità Ciclomatica	Metrica per il calcolo del numero di decisioni del programma.	Usata per stabilire la complessità del programma.

8 Piani aggiuntivi

Si prevede di organizzare un meeting dopo la fine del progetto per un'attività di lessons learned più approfondita (una attività preliminare sarà fatta prima della consegna) con tutti i team members che sarà usata per arricchire la presentazione finale del progetto, dal lato del PM.