



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



FITDIARY
DECIDE, COMMIT, SUCCEED

Software Project Management Plan

Riferimento 2021_SPMP_C07_FitDiary_Fasano-Spinelli_V1.0

Versione 1.0

Data 17/01/2022

Destinatario Prof.ssa Filomena Ferrucci

Presentato da Salvatore Fasano, Gianluca Spinelli

Approvato da



Revision History

Data	Versione	Descrizione	Autori
31/12/2021	0.1	Prima stesura	Salvatore Fasano, Gianluca Spinelli
17/01/2022	1.0	Completamento documento	Salvatore Fasano, Gianluca Spinelli



Team Members

Ruolo	Nome e Cognome	Acronimo	Email
PM	Salvatore Fasano	SF	s.fasano10@studenti.unisa.it
PM	Gianluca Spinelli	GS	g.spinelli18@studenti.unisa.it
TM	Daniele De Marco	DM	d.demarco11@studenti.unisa.it
TM	Ilaria De Sio	IS	i.desio7@studenti.unisa.it
TM	Rebecca Di Matteo	RM	r.dimatteo10@studenti.unisa.it
TM	Daniele Giaquinto	DG	d.giaquinto2@studenti.unisa.it
TM	Davide La Gamba	DL	d.lagamba@studenti.unisa.it
TM	Leonardo Monaco	LM	l.monaco11@studenti.unisa.it
TM	Simone Spera	SS	s.spera7@studenti.unisa.it
TM	Antonio Trapanese	AT	a.trapanese8@studenti.unisa.it



Sommario

Revision History	2
Team Members	3
1. Panoramica del progetto	6
1.1 Riepilogo del progetto	6
1.1.1 Scopo, ambito e obiettivi.....	6
1.1.2 Assunzioni e vincoli	6
1.1.3 Deliverables di progetto	6
1.1.4 Riepilogo dello schedule e del budget	8
1.2 Evoluzione dello SPMP	9
1.3 Riferimenti.....	9
2. Contesto del progetto.....	9
2.1 Modello di processo.....	10
2.2 Process Improvement Plan.....	10
2.3 Infrastructure Plan	11
2.4 Metodi, tool e tecniche.....	11
2.4.1 Metodi	11
2.4.2 Tools e tecnologie.....	11
2.5 Product Acceptance Plan	13
2.6 Organizzazione del progetto	13
3. Pianificazione del progetto	14
3.1 Avvio del progetto	14
3.1.1 Estimation Plan.....	14
3.1.2 Staffing Plan	15
3.1.3 Resource Acquisition Plan	15
3.1.4 Training Plan	15
3.2 Project Works Plans.....	16
3.2.1 Work activities.....	16
3.2.2 Schedule Allocation.....	16
3.2.3 Resource Allocation	16
3.2.4 Budget Allocation	16
4. Valutazione e controllo del progetto	17
4.1 Requirements Management Plan	17
4.2 Scope Change Control Plan.....	17
4.3 Schedule Control Plan.....	18



4.4 Budget Control Plan	18
4.5 Quality Assurance Plan.....	19
4.6 Project Closeout Plan	19
5. Consegna del prodotto.....	19
6. Supporting Process Plans	20
6.1 Supervisione e ambiente di lavoro del progetto	20
6.2 Decision Management.....	21
6.3 Risk Management.....	21
6.4 Configuration Management.....	21
6.5 Quality Assurance.....	21
6.6 Misure.....	22
7. Piani aggiuntivi	22



1. Panoramica del progetto

1.1 Riepilogo del progetto

1.1.1 Scopo, ambito e obiettivi

L'azienda specializzata nel settore del fitness intende consentire ai professionisti del campo, di supportare al meglio i propri clienti con lo scopo di raggiungere in modo agevole i risultati prefissati. L'azienda, quindi, vuole introdurre un cambiamento all'interno del settore, in modo da eliminare l'utilizzo di tutti i canali inadeguati attualmente utilizzati, quali WhatsApp, Instagram, Telegram ecc. . Data l'esistenza di tale necessità e riscontrata una mancanza di sistemi capaci di offrire soluzioni al problema, l'azienda ritiene interessante la possibilità di offrire una piattaforma in grado di supportare il lavoro di tali professionisti. L'obiettivo del progetto è fornire uno strumento di supporto alle attività di comunicazione e gestione dei clienti da parte di personal trainers e nutrizionisti garantendo una esperienza migliore ed una maggior probabilità di raggiungimento degli obiettivi prefissati. Il sistema proposto dovrà prevedere:

- tutte attività di creazione e gestione di un nuovo cliente;
- il processo di creazione dei protocolli creati dai professionisti;
- l'attività di consultazione dei contenuti da parte del cliente;
- la possibilità, per ogni cliente, di comunicare informazioni relative ai propri progressi;
- un meccanismo di previsione dei possibili obiettivi raggiungibili dal cliente.

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 Jira e Microsoft Project. Oltre ciò, sia assume che lo sviluppo dell'applicativo avverrà tramite l'uso di tecnologie web, nella fattispecie utilizzando Maven, Spring Boot e ReactJS.

1.1.3 Deliverables di progetto

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

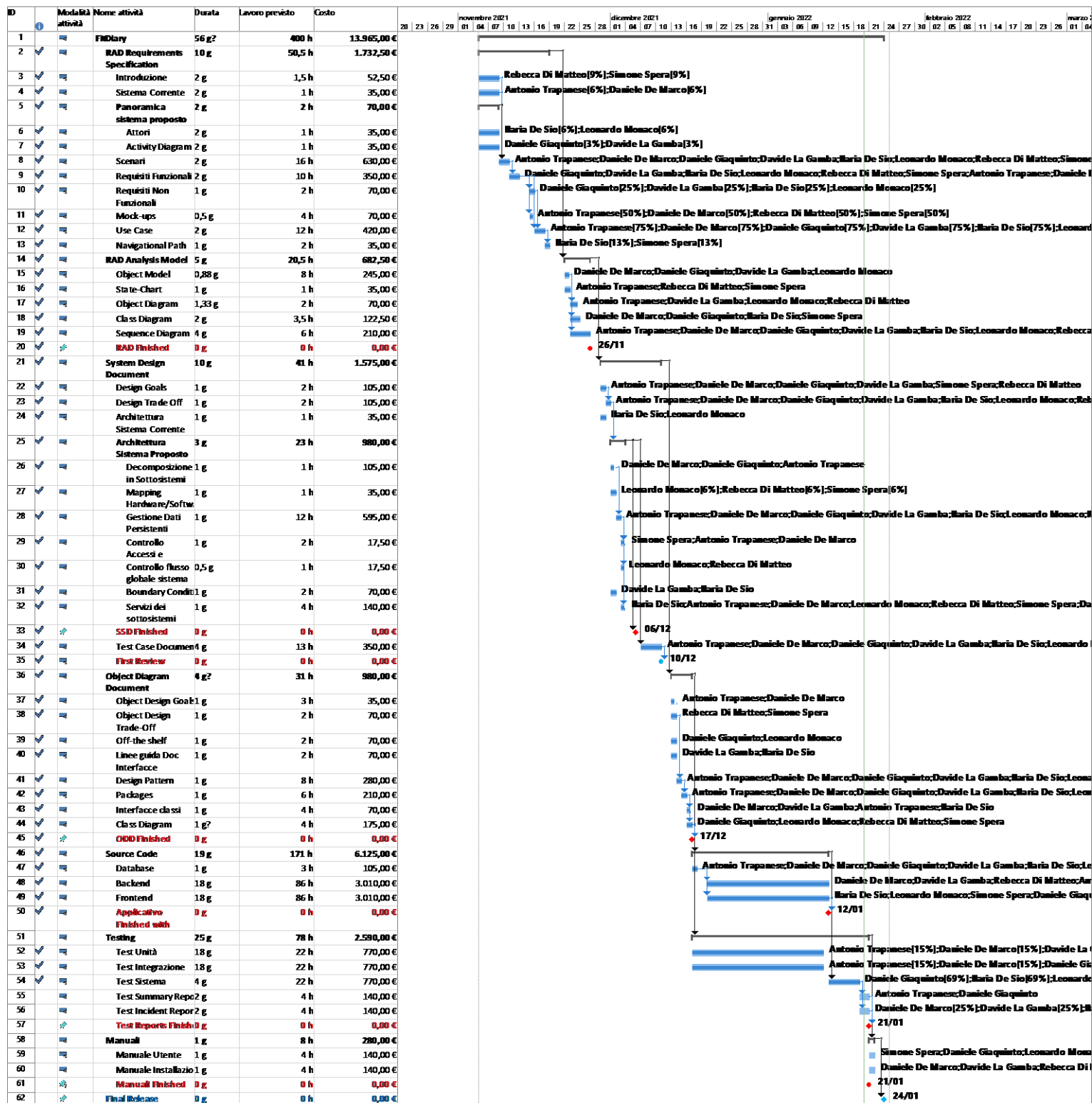


Deliverables/Milestones	WBS Id	Responsabile	Completamento Pianificato
RAD	1.2.6	Tutto il team	26/11/2021
SDD	1.3.5	Tutto il team	06/12/2021
TP e TCS	1.3.6	Tutto il team	10/12/2021
Consegna Intermedia	1.3.7	Tutto il team	10/12/2021
ODD	1.4.9	Tutto il team	17/12/2021
Applicativo, Javadoc	1.5.4	Tutto il team	12/01/2022
Test Summary Report, Test Incident Report	1.6.6	Tutto il team	21/01/2022
Manuale d'uso, Manuale d'installazione	1.7.3	Tutto il team	21/01/2022
Consegna finale	1.8	Tutto il team	24/01/2022



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.





Per maggiori info 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 e periodicamente ogni due settimane dall'inizio del progetto. L'aggiornamento avverrà seguendo le regole previste nel documento [Configuration Management Plan](#).

1.3 Riferimenti

Di seguito la lista agli altri documenti di management:

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

Nel presente documento si fa anche riferimento ad altri documenti, sviluppati nell'ambito del corso triennale di IS. Di seguito, si lasciano alcuni riferimenti a questi documenti:

- [Requirements Analysis Document](#)
- [System Design Document](#)
- [Object Design Document](#)
- [Test Plan](#)

2. 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.



2.1 Modello di processo

Il progetto FitDiary verrà eseguito combinando due diversi modelli di ciclo di sviluppo ben documentate in letteratura.

Dalla fase di Requirements Elicitation alla fase di Object Design sarà usato un tipico modello a V con possibilità di tornare indietro, solo se strettamente necessario e sostenibile di termini di budget. È stata effettuata tale scelta, in quanto si vuole sviluppare un'attitudine alla progettazione di buon testing nel team di sviluppo sin dalle prime fasi. Inoltre, un buon testing garantirà una buona qualità del codice.

Per quanto riguarda la validazione dei requisiti, questa avverrà producendo dei Wire-frame dopo una conversazione con il cliente rappresentativo "Mario Rossi", a cui verranno inviati per avere dei feedback sull'impostazione del sistema in termini di features.

Invece, per le fasi di sviluppo, verrà utilizzato l'approccio Agile Scrum, documentato nello Scrum Product Backlog.

Nello specifico, era desiderio dei Project Managers fare in modo che i team members potessero vivere l'esperienza dello Scrum, per poterne trarre beneficio in termini di conoscenze.

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 (GitHub Action).

Per ciò che concerne il configuration management si fa riferimento al documento di Configuration Management Plan.

2.2 Process Improvement Plan

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

- **Training:** prima di ogni fase del progetto FitDiary ci sarà un meeting in cui verranno presentati e spiegati degli esempi di artefatti che dovranno essere prodotti, in modo da evitare dubbi o confusioni future. Si precisa che, per quanto riguarda gli aspetti circa lo sviluppo del codice, oltre ai normali meeting verrà fornito ulteriore materiale sotto forma di video e link di documentazione.
- **Sprint Retrospective:** durante la fase di sviluppo si seguirà un modello agile, lo Scrum. Tale modello prevede l'esecuzione di analisi a posteriori di ogni fase di lavoro per lo sviluppo di una versione dell'applicativo. Tali retrospettive saranno fatte attraverso una riunione formale e le informazioni ottenute saranno usate per migliorare il processo. Tali informazioni saranno contenute in una serie di documenti separate dagli altri, facenti parte di un diverso modello di ciclo di vita.



2.3 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 IDEA, e uno strumento di versioning locale, Git.

Per garantire una migliore comprensione delle modifiche al codice sorgente, si favoriranno commit di piccola taglia (poche modifiche) e si utilizzerà, per la stesura del messaggio di commit, il seguente standard ideato da Google <https://www.conventionalcommits.org/en/v1.0.0/>. Inoltre, si utilizzeranno feature messe a disposizione dalle tecnologie selezionate per differenziare le impostazioni tra l'ambiente di sviluppo e l'ambiente di produzione (ad esempio verrà utilizzato un database locale in caso di development, mentre per la produzione viene utilizzato un database PostgreSQL fornito su Heroku).

2.4 Metodi, tool e tecniche

2.4.1 Metodi

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

- **Scrum**, come metodologia di sviluppo Agile;
- **Pair Programming**, per lo sviluppo di parti complesse dell'applicativo;
- **Pull based development**, per la gestione dell'avanzamento del lavoro;
- **Revisioni**, per l'accounting della qualità del codice.

2.4.2 Tools e tecnologie

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

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

- **Jira**, per la gestione e coordinazione riguardanti i tasks di progetto e per l'implementazione delle Scrum Board per la fase di implementazione;
- **Slack**, per la comunicazione formale tra Project Managers e team members;
- **Telegram**, per la comunicazione informale tra tutti i membri del team;
- **Microsoft Teams**, per meeting formali settimanali, eventuali meeting straordinari e per videoconferenze;



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, sia Deliverables di prodotto che documenti di Management;
- **Microsoft Word**, per la scrittura dei documenti;
- **Microsoft Excel**, per i fogli di lavoro, matrice di tracciabilità ed altri documenti di analisi dei dati.
- **Draw.io**, per la creazione dei diagrammi UML;
- **Balsamiq**, per la creazione dei wire-frames delle interfacce grafiche;
- **IntelliJ IDEA**, per lo sviluppo del codice sorgente;
- **Selenium IDE**, per il testing di sistema;

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

- **Java 11**, come linguaggio di programmazione backend;
- **Spring Boot framework**, come framework java per facilitare lo sviluppo backend;
- **Hibernate**, come tecnologia semantica per la conversione da oggetti a entità SQL;
- **JUnit**, come framework per il test di unità e integrazione;
- **Mockito**, come framework per il mocking per il test di unità;
- **React JS**, come linguaggio di programmazione front-end;
- **CSS**, come linguaggio di styling del front-end;
- **Chakra UI**, come libreria di componenti per la realizzazione delle interfacce grafiche del sistema.

Per ciò che concerne la 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/Continuous Deployment automatici ad ogni Pull request;
- **Jacoco**, per il calcolo delle metriche relative alle branch coverage;
- **Checkstyle**, per il rispetto della convenzione Sun di Java;
- **Heroku**, piattaforma per il deploy del backend;
- **Vercel**, piattaforma per il deploy del frontend.



2.5 Product Acceptance Plan

La prima release del prodotto sarà consegnata e mostrata al Top Manager nella settimana dopo il 24/01/2022. Tale consegna prevederà una presentazione sull'approccio di management, approccio allo sviluppo dell'intero progetto e presentazione dei 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:

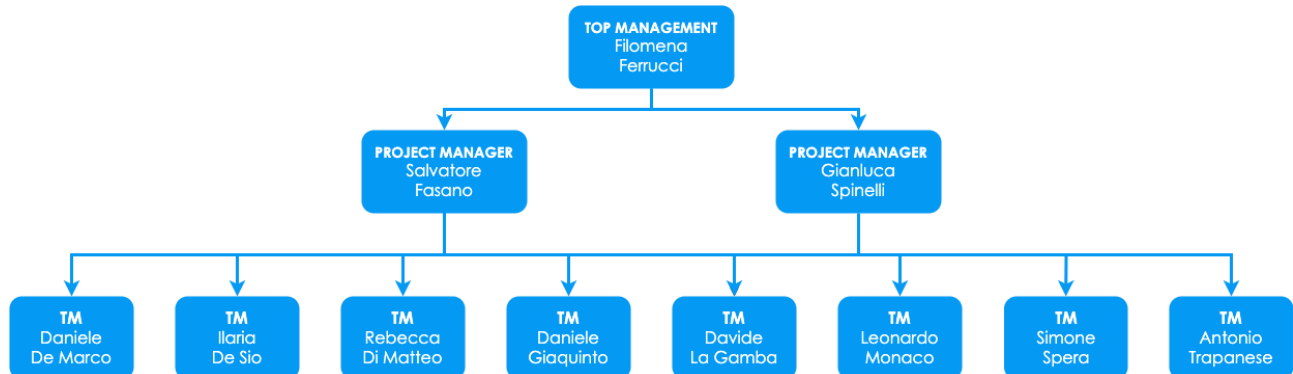
- Consegna dei deliverable di progetto entro le date fissate dal Top Manager;
- Consegna dell'applicativo con le funzionalità a priorità elevata entro la data ultima del progetto;
- Elevata qualità di tutti gli artefatti prodotti, dimostrata attraverso l'uso di check-list di qualità e tools per il calcolo di metriche sul codice;
- Branch-coverage del test dell'applicativo superiore al 75%, calcolata tramite l'uso del tool Jacoco;
- Numero di errori dati in output da Checkstyle, calcolati in base al rispetto della convenzione Sun di Java, inferiore a 20.

2.6 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 Managers. Questi 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. Per supportare il processo di qualità è stato realizzato un workflow sul tool utilizzato per gestire i task, Jira, che include anche uno stato "In Revisione": i PM effettueranno una breve revisione dei task in tale stato e potranno decidere se la qualità è soddisfacente, di segnalare il task come completato oppure di rimetterlo in corso documentandone la motivazione. Per ogni task verranno stabiliti un assegnatario ed un reporter: l'assegnatario sarà direttamente responsabile del prodotto relativo al task, mentre il reporter si occuperà di effettuare dei controllo sul lavoro prodotto prima di mandarlo in revisione ai PM.



Di seguito, l'organigramma dell'organizzazione.



3. 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.

3.1 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.

3.1.1 Estimation Plan

Il budget a disposizione risulta essere di 50 ore a team member per un totale di 400 ore a cui vanno ad aggiungersi le 100 ore totali dei Project Managers. Di seguito, si mostrano come tali ore sono state allocate in relazione alle diverse macro-attività del progetto FitDiary:

- RAD Requirements Specification: 50.5h;
- RAD Analysis Model 20.5h;
- System Design Document: 41h;
- Object Design Document: 31h;
- Source Code: 175h;
- Testing: 74h;
- Manuali: 8h.

Come da budget, sono state stimate esattamente 400 ore totali per la realizzazione del progetto.



3.1.2 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. Tutte le abilità/soft skills richieste dai Project Managers saranno presentate nella presentazione introduttiva agli studenti triennali.

Le abilità di tipo tecnico riguardano prettamente conoscenze in ambito di programmazione orientata agli oggetti, costruzione di basi di dati e schermate web. Oltre ciò, vengono richieste competenze ingegneristiche nella modellazione, formalizzazione e risoluzione di problemi. Ciò nonostante, si prevede che tali abilità andranno a migliorare nel corso del progetto.

3.1.3 Resource Acquisition Plan

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

1. Gli studenti triennali compileranno un questionario sulle loro capacità tecniche, soft skill ed esperienze passate;
2. I Project Managers faranno una presentazione dei propri progetti e della propria persona a tutti gli studenti triennali;
3. Gli studenti formeranno dei gruppi autonomi ed esprimeranno le proprie preferenze in modo da assegnare una di questa, in ordine decrescente, ad ogni progetto proposto;
4. I Project Managers visioneranno le preferenze e sceglieranno un gruppo, risolvendo eventuali conflitti tra loro o prendendo singoli membri da gruppi divisi.

3.1.4 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, i Project Managers potrà decidere di organizzare sessioni di training addizionali e non pianificate. Altri training ancora potrebbero essere concordati tra i team member e i Project Managers a discrezione dei primi.

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

- Training sui tools di management come Jira e Slack;
- Training sugli artefatti del RAD e la loro utilità;
- Training sugli artefatti dello SDD e la loro utilità;
- Training sugli artefatti dell'ODD;
- Training su Spring, Spring Boot, Spring MVC, Maven;
- Training su React JS;
- Training sul Rest API, HTTP Code, HTTP Methods ed JSend Standard;



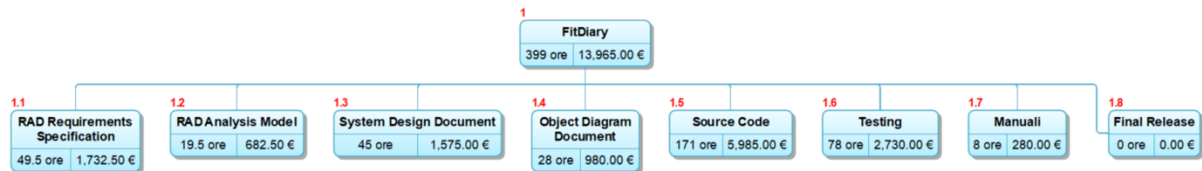
- Training su Continuous Integration;
- Training su conventional commit per commit message;
- Training sul test di unità e integrazione;

3.2 Project Works Plans

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

3.2.1 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](#) e [Schedule Management Plan](#).

3.2.2 Schedule Allocation

Lo schedule delle attività è descritto nei documenti di schedule (Schedule Management Plan) e nei files ottenuti tramite Microsoft Project 2016 (allegati alla consegna).

3.2.3 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.

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



4. Valutazione e controllo del progetto

4.1 Requirements Management Plan

La raccolta dei requisiti risulta essere una parte essenziale nell'ambito dello sviluppo di un buon prodotto software. Nel nostro caso la raccolta dei requisiti sarà fatta sfruttando l'Activity Diagram del sistema proposto, suddiviso opportunamente per attori, informazioni reperite dai Project Managers intervistando il cliente rappresentativo e scenari. I requisiti così ottenuti saranno poi raffinati e validati attraverso use case e wire-frame. In particolare, la realizzazione di wire-frame consentirà a tutto il team di avere una visione generale del flusso di esecuzione dell'intero sistema e potranno essere quindi usati per modificare eventualmente i requisiti funzionali, prima di procedere con le altre attività del RAD.

Si precisa che i requisiti funzionali saranno elencati seguendo lo standard IEEE, associando loro una priorità ed alcuni di questi verranno specificati anche mediante User Story.

Tutti i requisiti a priorità elevata saranno completamente documentati e implementati nella prima versione. I requisiti a priorità media saranno documentati fino al documento di System Design e infine, i requisiti a priorità bassa non saranno documentati. 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.

4.2 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:

- **Proposta di cambiamento:** il proponente, che può essere qualsiasi stakeholder, deve compilare un modulo contenente il titolo del progetto, l'id identificativo del cambiamento e la descrizione dettagliata del cambiamento;
- **Valutazione del cambiamento:** i Project Managers analizzeranno l'impatto del cambiamento, in termini di modifiche da effettuare, di rischi associati ed il corrispondente valore di business;
- **Condivisione di opinioni:** la modifica sarà valutata con l'intero team di sviluppo e ogni team member potrà esprimere la propria opinione;
- **Accettazione o rifiuto del cambiamento:** la richiesta potrà essere accettata o rifiutata dai Project Managers sulla base delle analisi effettuate;



- **Annotazione su Jira ed implementazione:** il team individua assieme tutti gli artefatti che si devono modificare per implementare la Change Request. Successivamente i Project managers caricano un task di modifica su Jira ed una volta che tutti gli artefatti saranno modificati e il task sarà chiuso e la modifica sarà considerata implementata.

4.3 Schedule Control Plan

La pianificazione e gestione dello schedule avverrà prevalentemente attraverso l'uso di Microsoft Project 2016, il quale rappresenta uno standard per questa tipologia di attività. Durante il corso del progetto, i Project Managers aggiorneranno manualmente i progressi tramite il software, visualizzando l'andamento del progetto rispetto alle previsioni, prendendo così decisioni in caso di necessità. Per quanto concerne l'Earned Value Management, le varie misurazioni verranno racchiuse all'interno del documento Excel "2021_EVM_C07_FitDiary_Fasano-Spinelli". Oltre quanto precedentemente detto, dopo l'assegnazione dei singoli tasks i Project Managers terranno traccia dello stato tramite l'uso del tool Jira (mediante Kanban Board), andando a fissare scadenze, priorità, effort stimato, assegnatario, revisore ed eventuali info e allegati, per ognuno di essi. Con cadenza settimanale, saranno effettuati meeting formali, generalmente il lunedì dalle ore 17 alle ore 18 nel canale C07 su Teams, comprendenti di una parte di status sui tasks in corso per la prossima milestones.

4.4 Budget Control Plan

Anche per il controllo dei costi e del budget si farà uso degli strumenti forniti da Microsoft Project 2016. Anche in questo caso, grazie all'Earned Value, si otterranno metriche come, ad esempio, l'indice di performance dei costi (CPI). Sempre sulla base di tali metriche, saranno prese decisioni da parte dei Project Managers.



4.5 Quality Assurance Plan

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

4.6 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 Project Managers ai team member e sarà fatta una presentazione sul progetto sia dai Project Managers che dai team members per illustrare l'intero progetto e le attività svolte.

5. Consegna del prodotto

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

Di seguito la lista dei deliverables di management:

- Statement of Work (SOW)
- Business Case (BC);
- Project Charter;
- Software Project Management Plan (SPMP);
- Team Contract;
- Scope Statement (SSD);
- WBS Dictionary (WBS);
- Schedule Management Plan (SPM);
- Configuration Management Plan (CMP);
- Cost Baseline (CB);
- Status Reports;
- Final Project Presentation;
- Final Status Report;
- Lesson Learned Report;
- Quality Management Plan (QMP);
- Risk Management Plan (RMP);
- Risk Register;



- Documentazione di Scrum;
- Agende di meeting;
- Minute di meeting.

Di seguito la lista dei deliverables di prodotto:

- Requirements Analysis Document (RAD);
- System Design Document (SDD);
- Test Plan (TP);
- Test Case Specification (TCS);
- Object Design Document (ODD);
- Matrice di tracciabilità;
- Test Incident Report (TIR);
- Test Summary Report (TSR);
- Source Code;
- Manuale d'uso (MU);
- Manuale d'installazione (MI),
- Documenti di revisione;
- Foglio di lavoro (FdL).

6. Supporting Process Plans

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

6.1 Supervisione e ambiente di lavoro del progetto

Durante il corso della settimana, ogni team member lavorerà, eventualmente in gruppo, ai task previsti per quella stessa settimana. Una volta a settimana, durante il meeting formale, ogni team member dovrà aggiornare i Project Managers sullo status dei task assegnatigli. In caso di problemi, i Project Managers saranno disponibili per eventuali incontri straordinari. I Project Managers controlleranno gli artefatti prodotti ogni settimana, per aumentare la qualità e aiutare i Team Members a comprendere bene gli standard fissati per la documentazione. Al raggiungimento di una milestone, la revisione degli artefatti, in concomitanza con il controllo della Checklist compilata sarà assegnata ad alcuni Team Members e, una volta conclusa, i Project Managers procederanno ad una seconda ed ultima revisione. Il lavoro verrà principalmente svolto a distanza, tramite il tool Microsoft Teams.



La valutazione del lavoro svolto dai Team Members si baserà su una serie di parametri valutativi proposti dai Project Managers ed accettati dal team di sviluppo durante la firma del Team Contract.

Si lasciano i riferimenti al [Team Contract](#) per la griglia di valutazione ed al documento di [valutazioni](#).

6.2 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 Managers 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, saranno prese da tutto il gruppo durante i meeting formali e le riunioni di gruppo. In tale sede qualsiasi membro del team può presentare proposte, le quali saranno discusse durante i meeting. La decisione finale spetta comunque ai Project Managers, i quali specificheranno le motivazioni di accettazione e di rifiuto della proposta. Ogni team member inoltre sarà incaricato a turno di dover effettuare la stesura di una minuta per tenere traccia delle decisioni intraprese durante l'avanzamento del progetto.

6.3 Risk Management

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

6.4 Configuration Management

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

6.5 Quality Assurance

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



6.6 Misure

Le metriche di qualità sono racchiuse nel documento di [Quality Management Plan](#).

Le metriche dimensionali utilizzate sono le seguenti:

Nome metrica	Descrizione	Motivazione
Lines of Code LOC	Numero totale di linee di codice	Metrica dimensionale per caratterizzare la dimensione dell'applicativo
Non Comment 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

Oltre le metriche precedentemente citate si è cercato di misurare la produttività del team attraverso l'uso di Burndown charts in fase di implementazione, avendo usato un approccio Scrum. In tali diagrammi si è tenuto conto del Focus Factor, che a causa delle nuove tecnologie è stato tenuto basso (0.6) inizialmente ed incrementato poi a 0.8 grazie all'esperienza ed alla produttività in aumento. Questo fattore è servito per stimare il carico di lavoro e i tempi di completamento nel singolo Sprint. Tali documenti possono essere trovati nella cartella Scrum.

7. 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, tale attività sarà utilizzata per arricchire la presentazione finale del progetto, dal lato dei Project Managers.