



Università
Ca'Foscari
Venezia

Piano di Progetto

Data di consegna: 23/10/2017



CRBM

Callegher Gianmarco 854612@stud.unive.it

Ragazzo Alessio 860550@stud.unive.it

Benvenuto Francesco 858843@stud.unive.it

Makaj Aurelio 859759@stud.unive.it

Versione 1.1

Indice

1. Introduzione:

- 1.1. Overview del progetto
- 1.2. Deliverables del Progetto
- 1.3. Evoluzione del Progetto
- 1.4. Materiale di riferimento
- 1.5. Definizioni e Abbreviazioni

2. Organizzazione del progetto:

- 2.1. Modello del Processo
- 2.2. Struttura Organizzativa
- 2.3. Interfacce organizzative

3. Processi gestionali:

- 3.1. Obiettivi e Priorità
- 3.2. Assunzioni, Dipendenze, Vincoli
- 3.3. Gestione dei rischi
- 3.4. Meccanismi di monitoraggio e di controllo
- 3.5. Pianificazione dello staff

4. Processi tecnici:

- 4.1. Metodi, Strumenti e Tecniche
- 4.2. Documentazione del Software
- 4.3. Funzionalità di supporto al progetto

5. Pianificazione del lavoro, delle risorse umane e del budget

- 5.1. WBS
- 5.2. Dipendenze
- 5.3. Risorse Necessarie
- 5.4. Allocazione del Budget e delle Risorse
- 5.5. Pianificazione

1. Introduzione

1.1. Overview del progetto:

Lo scopo del progetto è sviluppare un'app per dispositivi mobili Android la quale fornirà un servizio che permetterà di navigare, rappresentare, valorizzare, incrociare i dati messi a disposizione dalle Pubbliche Amministrazioni.

In particolare verranno utilizzati i dati relativi ai bilanci e agli appalti delle ULSS della Regione Veneto in correlazione al bacino di utenza delle strutture ospedaliere da esse incorporate.

1.2. Deliverables del Progetto:

Durante questi quattro mesi di sviluppo del progetto vi saranno le seguenti deadline da rispettare:

- Piano di Progetto (23/10/2017)
- Documento di analisi e specifica (9/11/2017)
- Piano di testing (23/11/2017)
- Documento di Progettazione (22/12/2017)
- Realizzazione e messa in linea (31/01/2018)

1.3. Evoluzione del Progetto:

La scelta delle ULSS della Regione Veneto deriva da un semplice contatto diretto dei componenti del gruppo con il suddetto territorio.

Qualora la stessa tipologia di dati fosse messa a disposizione anche dalle altre regioni, l'uso dell'applicazione potrebbe essere esteso a livello nazionale.

1.4. Materiale di riferimento

- Slides del corso di Ingegneria del Software del Professor Cortesi
- Voci di bilancio: <http://soldipubblici.gov.it/it/home>
- Open Data offerti dai siti delle ULSS prese in considerazione
- Libro per approfondimenti sull'Ingegneria del Software: Ian Sommerville, Ingegneria del software, settima ed, Pearson Education

1.5. Definizioni e Abbreviazioni

Android: sistema operativo per dispositivi mobili

Beta tester: persona incaricata di effettuare prove e collaudi di software non ancora pubblicato, con lo scopo di trovare eventuali errori (bug)

Deadline: scadenza

Deliverables: oggetto materiale o immateriale realizzato (fornito/consegnato) come risultato di un'attività del progetto

IDE: ambiente di sviluppo integrato. È un software che, in fase di programmazione, aiuta i programmatori nello sviluppo del codice sorgente di un programma

Open Data: dati liberamente accessibili a tutti le cui eventuali restrizioni sono l'obbligo di citare la fonte o di mantenere la banca dati sempre aperta. La Pubblica Amministrazione ha l'obbligo di fornire i dati, anche tramite le nuove tecnologie dell'informazione e della comunicazione, in un'ottica di trasparenza

PA: Pubbliche Amministrazioni

ULSS: Unità locale socio sanitaria

WBS: Elenco di tutte le attività di progetto

2. Organizzazione del progetto:

2.1. Modello del processo

Si è scelto di utilizzare un modello a cascata. Le fasi che andremo ad attraversare per poter concludere il nostro progetto sono le seguenti:

- Analisi e definizione dei requisiti
- Progettazione del sistema del software
- Implementazione e test delle singole unità
- Integrazione e test del sistema
- Installazione e mantenimento

Ciascuna fase potrà essere revisionata, ampliata ed integrata tramite feedback derivanti dalla fase successiva.

2.2. Struttura organizzativa

Vista la ridotta dimensione del gruppo, si seguirà una struttura organizzativa nella quale i vari componenti non sono vincolati a ruoli specifici, facilitando la comunicazione tra pari, e così sfruttando le abilità risolutive di tutti i membri riguardanti gli eventuali problemi riscontrati durante il ciclo di vita del progetto.

I componenti, i quali condividono lo stesso livello di conoscenza negli ambiti trattati, hanno optato per una gerarchia di tipo democratico decentralizzato; in questo modo l'assenza di un leader permanente permetterà un maggiore scambio di idee e collaborazione, rendendo impossibile un conflitto gerarchico.

2.3. Interfacce organizzative

Per la realizzazione del progetto, sarà necessario comunicare con altre entità esterne al gruppo di lavoro, principalmente per ottenere feedback riguardanti i materiali prodotti e informazioni utili alla stesura del codice.

Le principali figure di riferimento saranno:

- Docente Cortesi Agostino, per periodici feedback sul lavoro in corso
- Dott. Spanò Alvise, per consigli tecnici sullo sviluppo di applicazioni Android
- Beta-tester per poter verificare lo stato finale del prodotto

3. Processi gestionali

3.1. Obiettivi e Priorità

L'obiettivo consiste nella creazione di un'applicazione per dispositivi Android, funzionante, di pubblica utilità ed intuitiva, che permetta la visualizzazione di una mappa indicante le aree di copertura delle ULSS di cui verranno sfruttati gli Open Data e la possibilità di interazione da parte dell'utente per effettuare comparazioni fra essi in modo da poter osservare sotto quali aspetti differiscono.

Le priorità fissate dal team sono le seguenti:

- rispettare in modo puntuale le scadenze stabilite, consegnando dei deliverables di buona fattura
- mantenere costante e uniforme la partecipazione da parte di tutti i membri del team
- rilasciare un prodotto finale di qualità che possa eventualmente essere ripreso in futuro per ulteriori migliorie

3.2. Assunzioni, Dipendenze, Vincoli

Si assume che ciascun membro del gruppo svolga il proprio lavoro con responsabilità e che gli strumenti necessari allo sviluppo dell'applicazione siano disponibili ad ogni componente.

La qualità del prodotto finale dipende dal coinvolgimento di ogni membro e quindi da quanto tempo esso dedica all'apprendimento di nuove conoscenze.

I vincoli sono soprattutto di natura temporale dettati dalla necessità di rispettare le scadenze, e di natura organizzativa, dal momento che è necessario far conciliare le attività di progetto con gli impegni personali di ciascun componente del team.

3.3. Gestione dei rischi

Il progetto è soggetto a diversi rischi che ne potrebbero intaccare la buona riuscita; quelli identificati sono i seguenti:

- Mancanza di collaborazione: qualora il rendimento di un membro del team calasse, bisognerà attivarsi al fine di migliorare l'affinità complessiva del gruppo, utilizzando tecniche comunicative più efficaci
- Abbandono da parte di un componente del gruppo: è possibile che un componente del gruppo abbandoni il progetto. Ne consegue che il carico di lavoro degli altri membri aumenterà al fine di sopperire tale mancanza. Per evitare che ciò avvenga occorre che vi sia una comunicazione frequente per identificare i problemi personali di ciascun membro e quindi risolverli di conseguenza
- Problemi di salute: come nel caso precedente si effettuerà una ripartizione del carico, la quale, però, sarà momentanea e tornerà ai livelli normali non appena tutti i componenti saranno operativi
- Malfunzionamento dei sistemi informatici: qualora vi fosse un guasto ad un PC di un componente, esso dovrà impegnarsi al fine di attivare una postazione di lavoro all'interno del laboratorio di informatica del campus scientifico e di usufruire di essa fino alla risoluzione del problema
- Difficoltà di utilizzo degli strumenti: è indispensabile che tutti i componenti siano in grado di utilizzare gli strumenti di sviluppo prefissati per evitare futuri ritardi. Coloro i quali hanno una maggiore padronanza dei sistemi di lavoro, dovranno condividere le proprie conoscenze con i colleghi, al fine di allineare al livello massimo le conoscenze del team di progetto (ciò avverrà nella fase di training)
- Difficoltà nello sviluppo delle componenti di progetto: è possibile che, in fase di progettazione, siano stati commessi errori di stima della complessità di alcune delle componenti SW e sarà dunque necessario un affiancamento momentaneo da parte di un altro componente del gruppo al responsabile dell'attività in questione allo scopo di trovare un'efficiente e rapida soluzione.
- Perdita dei dati: non è da escludere a priori la possibilità di perdita di dati. Per questa ragione verranno utilizzati spazi condivisi (cloud) su piattaforme consolidate che incorporano funzionalità automatiche di versionamento e backup (es drive per i documenti, github per il source code) oltre ad avere una copia in locale del progetto su ogni macchina utilizzata.

	RISCHIO	PROB	IMPATTO	SOLUZIONE
1	Mancanza di collaborazione	20%	medio-alto	sollecito al componente
2	Abbandono componente	5%	alto	ripartizione del carico
3	Salute	35%	medio-basso	ripartizione del carico
4	Malfunzionamenti strumentazioni	10%	medio-basso	sostituzione del mezzo
5	Difficoltà di utilizzo	10%	basso	training
6	Difficoltà di sviluppo	30%	medio	ricerca di una soluzione
7	Perdita dei dati	5%	alto	recupero da backup

Tabella dei rischi (1)

Probabilità/Impatto	Basso	Medio-Basso	Medio	Medio-Alto	Alto
Alta					
Media		3	6	1	
Bassa	5	4			2, 7

Tabella dei rischi (2)

3.4. Meccanismi di monitoraggio e di controllo

Durante lo svolgimento del progetto verranno svolte attività di controllo della correttezza e della qualità del lavoro svolto. Inoltre avranno luogo riunioni con l'obiettivo di: scambiare nuove idee, incentivare le proposte di miglioramento ed esporre le difficoltà e le relative soluzioni adottate. Gli incontri avverranno con cadenze periodiche, talvolta anche per via telematica.

Le decisioni dovranno essere prese di comune accordo effettuando, se necessario, dei compromessi al fine di non minare la compattezza del gruppo e dunque il relativo rendimento.

3.5. Pianificazione dello staff

Per sviluppare il progetto sono necessarie conoscenze relative alla programmazione orientata agli oggetti (in particolare del linguaggio Java) e praticità nell'utilizzo dell'ambiente di sviluppo ufficiale di Android (Android Studio).

Queste abilità, in parte già consolidate dai componenti del gruppo, saranno integrate e ampliate durante la funzione di training del team (periodi temporali dedicati appositamente a tale scopo).

4. Processi tecnici

4.1. Metodi, Strumenti e Tecniche

Come unici sistemi di calcolo verranno utilizzati i personal computer di ciascuno dei componenti del gruppo.

Un browser sarà lo strumento principale per la ricerca degli Open Data necessaria alla realizzazione del prodotto, mentre l'IDE Android Studio verrà utilizzata come principale ambiente di sviluppo. Le linee guida definite da XP Core, basate sui principi dell'Agile Manifesto, verranno seguite come metodologia di sviluppo.

4.2. Documentazione del Software

Durante il ciclo di vita del progetto, verranno realizzati i seguenti documenti: Piano di Progetto, Documento di Analisi e Specifica, Piano di Testing e Documento di Progettazione. Questi, nella fase di chiusura, saranno racchiusi in un unico documento al fine di fornire un punto di accesso unico alla lesson learned acquisita.

Le milestones che si pianifica di raggiungere sono le seguenti:

- Piano di Progetto (23/10/2017)
- Documento di analisi e specifica (9/11/2017)
- Piano di testing (23/11/2017)
- Documento di Progettazione (15/12/2017)
- Realizzazione e messa in linea (31/01/2018)
- Istruzioni all'uso dell'applicazione (31/01/2018)

4.3. Funzionalità di supporto al progetto

4.3.1. Piano della Qualità

Il prodotto finale dovrà essere in grado di rappresentare, incrociare e operare sugli Open Data messi a disposizione dalle PA e dalle ULSS della Regione Veneto. L'applicazione dovrà essere sviluppata in ottica di prevenire e gestire ogni eventuale eccezione di sistema. Al fine di raggiungere tale scopo verranno effettuate periodiche attività di riesame, verifica e validazione.

4.3.2. Pianificazione della gestione delle configurazioni

L'applicazione che verrà sviluppata sarà solo lo scheletro di un progetto più ampio, sul quale altri programmatori potranno operare al fine di ampliare, una volta disponibili i necessari Open Data degli enti di sanità regionali, il "raggio d'azione". Sarà dunque necessaria un'adeguata documentazione al fine di guidare, nel modo più facile possibile, un programmatore esterno che si approccia per la prima volta al codice sorgente.

5. Pianificazione del lavoro, delle risorse umane e del budget

5.1. WBS (Work breakdown structure)

- 5.1.1. Specifiche del Progetto
 - 5.1.1.1. Definire gli obiettivi e lo scope del progetto
 - 5.1.1.2. Analizzare i requisiti
 - 5.1.1.3. Analizzare le risorse
 - 5.1.1.3.1. Identificazione delle risorse materiali
 - 5.1.1.3.2. Definizione della tipologia di skills richieste
 - 5.1.1.3.3. Definizione del team ottimale
 - 5.1.1.4. Definire l'architettura
 - 5.1.1.4.1. Definizione dell'architettura per lo sviluppo
 - 5.1.1.4.2. Definizione dell'architettura applicativa
 - 5.1.1.4.3. Definizione dell'architettura comunicativa
 - 5.1.1.5. Stimare i costi
 - 5.1.1.6. Predisporre il piano dei rischi
 - 5.1.1.6.1. Identificazione dei rischi di progetto
 - 5.1.1.6.2. Analisi di probabilità e impatto
 - 5.1.1.6.3. Stesura piano rischi
 - 5.1.1.7. Gestire la comunicazione
 - 5.1.1.7.1. Definire i soggetti interessati
 - 5.1.1.7.2. Definizione delle modalità di comunicazione
 - 5.1.1.7.3. Definizione di tipologia e contenuto delle informazioni
 - 5.1.1.8. Predisporre il piano della Qualità
 - 5.1.1.8.1. Definizione attività di riesame, verifica e validazione
 - 5.1.1.8.2. Definizione di strumenti di controllo
 - 5.1.1.8.3. Definizione responsabilità attività
 - 5.1.1.9. Predisporre documento di specifiche di progetto
- 5.1.2. Progettazione concettuale
 - 5.1.2.1. Effettuare analisi dei requisiti
 - 5.1.2.1.1. Requisiti funzionali
 - 5.1.2.1.1.1. Analisi organizzazione utente
 - 5.1.2.1.1.2. Analisi interfaccia utente
 - 5.1.2.1.1.3. Analisi funzioni utente

- 5.1.2.1.1.4. Analisi navigazione utente
 - 5.1.2.1.1.5. Analisi ambiente di accoglienze
 - 5.1.2.1.2. Requisiti non funzionali
 - 5.1.2.1.2.1. Analisi volumi
 - 5.1.2.1.2.2. Analisi carico
 - 5.1.2.1.2.3. Analisi performance
 - 5.1.2.1.2.4. Analisi usabilità
 - 5.1.2.1.3. Analisi dei dati
 - 5.1.2.1.3.1. Identificazione delle entità
 - 5.1.2.1.3.2. Identificazione delle relazioni fra le entità
 - 5.1.2.1.3.3. Definizione del modello Entity-Relationship
 - 5.1.2.2. Effettuare analisi dei processi
 - 5.1.2.3. Effettuare analisi delle funzioni
 - 5.1.2.4. Predisporre documento di progettazione concettuale
- 5.1.3. Progettazione tecnica
 - 5.1.3.1. Effettuare una verifica della progettazione concettuale
 - 5.1.3.2. Definizione versioni Android target
 - 5.1.3.3. Definizione ambiente di sviluppo
 - 5.1.3.3.1. Scelta librerie
 - 5.1.3.3.2. Scelta tools
 - 5.1.3.4. Progettazione interfacce
 - 5.1.3.5. Definire standard da adottare
 - 5.1.3.6. Predisporre documento di progettazione tecnica
- 5.1.4. Sviluppo
 - 5.1.4.1. Effettuare una verifica della progettazione tecnica
 - 5.1.4.2. Sviluppare l'applicazione di anticorruzione
 - 5.1.4.2.1. Sviluppo interfaccia utente
 - 5.1.4.2.2. Sviluppo logica applicativa
 - 5.1.4.3. Produrre documentazione del software sviluppato
 - 5.1.4.3.1. Documentazione dell'applicazione di anticorruzione
- 5.1.5. Test
 - 5.1.5.1. Effettuare test di modulo
 - 5.1.5.1.1. Preparazione piano di test
 - 5.1.5.1.2. Preparazione dell'ambiente di test
 - 5.1.5.1.3. Preparazione casi di prova ed esiti attesi
 - 5.1.5.1.4. Esecuzione del test e tracciatura dei risultati
 - 5.1.5.2. Effettuare test di integrazione
 - 5.1.5.2.1. Preparazione piano di test
 - 5.1.5.2.2. Preparazione dell'ambiente di test
 - 5.1.5.2.3. Preparazione casi di prova ed esiti attesi
 - 5.1.5.2.4. Esecuzione del test e tracciatura dei risultati
 - 5.1.5.3. Effettuare test di sistema

- 5.1.5.3.1. Preparazione piano di test
- 5.1.5.3.2. Preparazione dell'ambiente di test
- 5.1.5.3.3. Preparazione casi di prova ed esiti attesi
- 5.1.5.3.4. Esecuzione del test e tracciatura dei risultati
- 5.1.5.4. Effettuare collaudo con il docente
 - 5.1.5.4.1. Supporto nella preparazione piano di test
 - 5.1.5.4.2. Supporto nella preparazione dell'ambiente di test
 - 5.1.5.4.3. Supporto nella preparazione dei casi di prova
 - 5.1.5.4.4. Supporto nella preparazione del collaudo
 - 5.1.5.4.5. Accettazione esiti collaudo da parte del docente
- 5.1.6. Pianificazione e controllo del progetto
 - 5.1.6.1. Analizzare e sviluppare la WBS
 - 5.1.6.2. Assegnare le risorse alle attività
 - 5.1.6.3. Predisporre il piano di progetto
 - 5.1.6.3.1. Definizione della baseline
 - 5.1.6.3.2. Definizione attività inizio, fine e successori
 - 5.1.6.3.3. Identificazione delle milestone del progetto
 - 5.1.6.4. Predisporre il Piano della Qualità
 - 5.1.6.4.1. Definizione attività di riesame, verifica e validazione
 - 5.1.6.4.2. Definizione di strumenti di controllo
 - 5.1.6.4.3. Definizione responsabilità attività
 - 5.1.6.5. Verificare la schedulazione
 - 5.1.6.6. Verificare l'ambito
 - 5.1.6.7. Verificare i costi
 - 5.1.6.8. Verificare la Qualità
 - 5.1.6.9. Effettuare riunioni di coordinamento con il project team
 - 5.1.6.10. Effettuare riunioni di Coordinamento con il docente e il Dott. Alvisè Spanò

Parallelizzazione delle attività:

- Specifiche del progetto (5.1.1):
 - Le attività da 5.1.1.1 a 5.1.1.4 dovranno essere serializzate.
 - Le attività 5.1.1.5 e 5.1.1.6 potranno essere parallelizzate dopo che le precedenti saranno state completate.
 - Le attività da 5.1.1.7 e 5.1.1.8 potranno essere parallelizzate dopo che le precedenti saranno state completate.
 - L'attività 5.1.1.9 dovrà essere effettuata alla fine come completamento della definizione delle Specifiche di Progetto.
- Progettazione Concettuale (5.1.2):
 - Tutte le attività potranno essere parallelizzate.
- Progettazione Tecnica (5.1.3):
 - L'attività 5.1.3.1 dovrà precedere le le altre.
 - L'attività 5.1.3.2 dovrà seguire alla 5.1.3.1.
 - L'attività 5.1.3.3 dovrà seguire alla 5.1.3.2.
 - Le attività 5.1.3.4 e 5.1.3.5 potranno essere parallelizzate dopo il completamento della 5.1.3.3.
 - L'attività 5.1.3.6 dovrà essere eseguita per ultima.
- Sviluppo (5.1.4):
 - L'attività 5.1.4.1 dovrà precedere le le altre.
 - L'attività 5.1.4.2 dovrà seguire alla 5.1.4.1.
 - L'attività 5.1.4.3 dovrà essere eseguita per ultima.
- Test (5.1.5):
 - Le attività di testing dovranno seguire il seguente ordine:
 - Test di modulo
 - Test di Integrazione
 - Test di sistema
 - Collaudo con il docente
 - Ogni attività di testing è composta da più fasi che dovranno essere eseguite nel seguente ordine:
 - Preparazione piano di test
 - Preparazione dell'ambiente di test
 - Preparazione casi di prova ed esiti attesi
 - Esecuzione del test e tracciatura dei risultati
- Pianificazione e controllo del progetto (5.1.6):
 - L'attività 5.1.6.1 dovrà precedere le le altre.
 - L'attività 5.1.6.2 dovrà seguire alla 5.1.6.1.
 - L'attività 5.1.6.3 dovrà seguire alla 5.1.6.2.

- L'attività 5.1.6.3 dovrà seguire alla 5.1.6.4.
- Le attività da 5.1.6.5 a 5.1.6.10 potranno essere parallelizzate dopo il completamento delle precedenti.

5.2. Dipendenze

Diagramma di Gantt

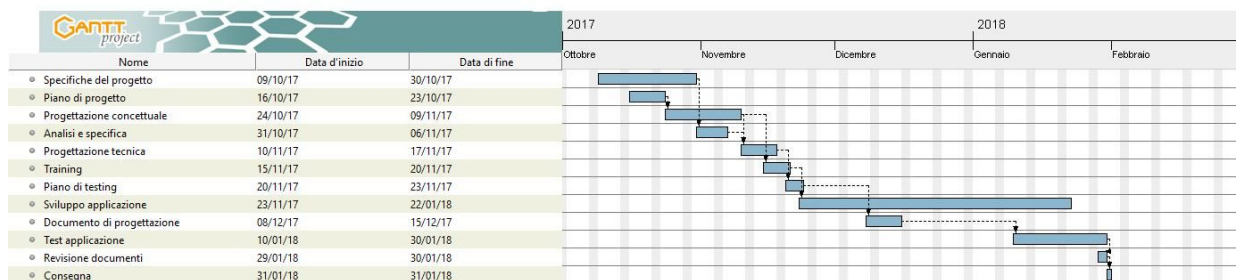
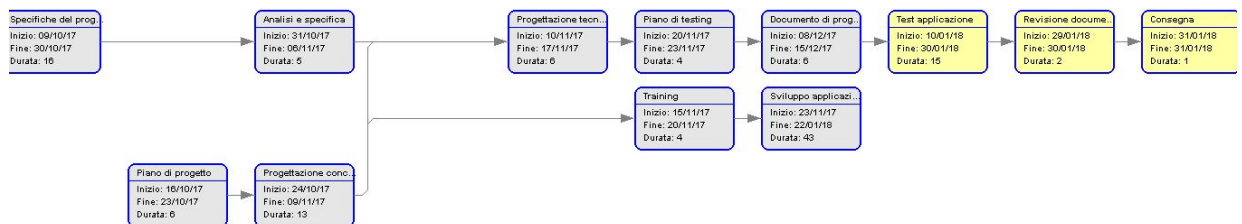


Diagramma di Pert



Nome attività	Giorni	Ore
Specifiche del progetto	16	16
Piano di progetto	6	12
Progettazione concettuale	13	20
Analisi e specifica	5	10
Progettazione tecnica	6	12
Training	4	16
Sviluppo applicazione	43	200
Piano di testing	4	8
Documento di progettazione	6	4
Test applicazione	15	30
Revisione documenti	2	4
Consegna	1	1

Giorni durata progetto totali: 121 gg.

Ore durata progetto totali: 333 ore.

Ore medie CAD: 83 ore

5.3. Risorse necessarie

Per una buona riuscita del progetto le risorse necessarie sono: un personale adeguatamente ampio, dei PC con l'IDE Android Studio installata, conoscenze di Project Management, conoscenze della programmazione Android, capacità di redazione dei documenti, e tempo da investire nelle attività di cui si compone il progetto.

Le persone coinvolte sono quattro studenti e, al fine di ottimizzare il tempo di ciascuno di essi in correlazione ai propri compiti di progetto, verrà redatto un piano il quale tenga conto delle scadenze di progetto e delle milestones prefissate.

Le conoscenze della Programmazione Android e le capacità di redazione dei documenti saranno ampliate tramite gli incontri con il Dott. Alvise Spanò le prime e con la partecipazione alle lezioni del corso di Ingegneria del Software le seconde.

5.4. Allocazione del Budget e delle Risorse

L'allocazione delle risorse alle attività di cui è composta ogni fase di progetto avverrà dopo il raggiungimento della milestone precedente, basando le proprie scelte sulla lesson learned acquisita task dopo task. Le risorse saranno assegnate al progetto a rotazione, al fine di ampliare le conoscenze di ciascuno dei membri a 360°.

5.5. Pianificazione

Il ciclo di vita del progetto prevede le seguenti quattro scadenze:

- Piano di Progetto (23/10/2017)
- Documento di analisi e specifica (9/11/2017)
- Piano di testing (23/11/2017)
- Documento di Progettazione (22/12/2017)
- Realizzazione e messa in linea (31/01/2018)

Inoltre il gruppo prevede il conseguimento delle seguenti milestone intermedie

- Progettazione concettuale
- Progettazione tecnica
- Sviluppo Applicazione
- Test Applicazione
- Rilascio Applicazione (31/01/2018)