

Ingegneria del Software: MyPollen Piano di progetto v1.0 - 20/10/2016

{IT}igli:

Gianluca Bigaglia 854406

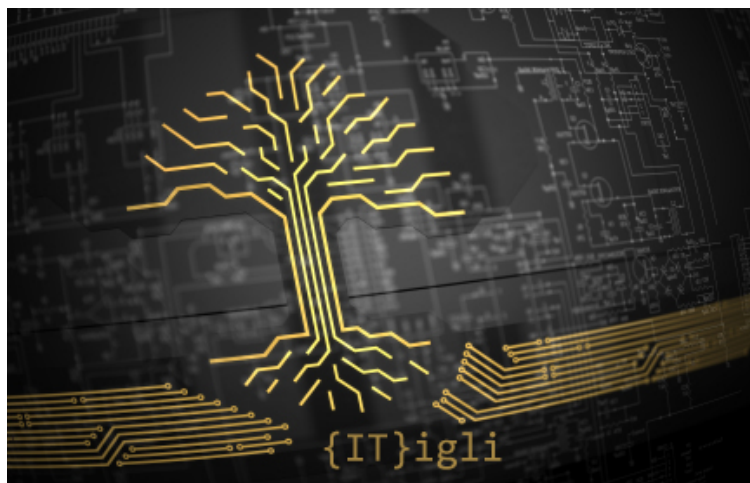
Luca Daniel 851269

Lorenzo Giudice 851032

Andrea Missiato 854305



Università Ca' Foscari Venezia



Indice

1	Introduzione	1
1.1	Overview del Progetto	1
1.2	Deliverables del progetto	1
1.3	Evoluzione del progetto	1
1.4	Materiale di riferimento	2
1.5	Definizioni e abbreviazioni	3
2	Organizzazione del progetto	4
2.1	Modello del processo	4
2.2	Struttura organizzativa	4
2.3	Interfacce organizzative	4
3	Processi gestionali	4
3.1	Obiettivi e priorità	4
3.2	Assunzioni, dipendenze, vincoli	5
3.3	Gestione dei rischi	6
3.4	Meccanismi di monitoraggio e di controllo	6
3.5	Pianificazione dello staff	7
4	Processi tecnici	7
4.1	Metodi, strumenti e tecniche	7
4.2	Documentazione del software	7
5	Pianificazione del lavoro, delle risorse umane, del budget	8
5.1	WBS(Work Breakdown Structure)	8
5.2	Dipendenze	10
5.3	Risorse necessarie	10
5.4	Allocazione del budget e delle risorse	11
5.5	Pianificazione	12
5.5.1	Diagramma di Gantt	12
5.6	Diagramma di Pert	14
5.6.1	Descrizione Pert	15
5.6.2	Milestones e Deadlines	15

1 Introduzione

1.1 Overview del Progetto

L'idea di progetto è quella di utilizzare gli open data forniti dall'ARPAV relativi ai "livelli di concentrazione in aria di pollini allergenici" raccolti a partire dall'anno 2008 fino allo scorso 2015.

L'obiettivo della nostra applicazione sarà quello di creare una mappa, per le varie città presenti nel dataset (Agordo, Belluno, Feltre, Padova, Rovigo, Treviso, Venezia, Verona, Vicenza), in cui saranno segnalate le varie aree a maggior rischio polline a seconda dei diversi periodi dell'anno. Oltre a questo l'applicazione prevederà ulteriori funzionalità che verranno specificate nel corso del documento. Sono state inoltre visionate applicazioni simili presenti all'interno del Play Store, ma a livello italiano nessuna risulta effettivamente funzionante o con buone recensioni. La nostra applicazione è rivolta principalmente a chi soffre di allergie da pollini, permettendo così agli utenti di sapere quali sono le aree che potrebbero mettere maggiormente in pericolo la loro salute.

Le features previste sono le seguenti:

- **Mappa:** visualizzazione in base all'utente / visualizzazione generica / visualizzazione con filtri della concentrazione dei pollini nel territorio veneto; essa verrà rappresentata per ogni zona tramite un alone il cui colore varierà in base ad una scala progressiva dal rosso al verde.
- **Area wiki:** per ogni polline saranno presenti informazioni sulle caratteristiche, i luoghi di diffusione e consigli vari per evitarli.

1.2 Deliverables del progetto

Fasi	Deadline
Definizione gruppi di lavoro	27/09/2016
Piano di progetto	21/10/2016
Documento di analisi e specifica	02/11/2016
Piano di testing	15/11/2016
Documento di progettazione	10/12/2016
Realizzazione e messa in linea	28/02/2017

1.3 Evoluzione del progetto

E' ancora da definire se nel progetto verranno integrate le seguenti funzionalità:

- **Alterazione dei dati in base al meteo:** ad esempio, un temporale potrebbe alterare la concentrazione di un certo tipo di polline.

- **Analisi predittiva dei dati:** avendo a disposizione dei dati non in tempo reale potrebbe risultare utile effettuare una stima di come potrebbe evolversi la concentrazione di polline durante gli anni.

1.4 Materiale di riferimento

Durante lo svolgimento del progetto verranno consultate le seguenti risorse:

- Il sito offerto da Google per gli sviluppatori android: (developer.android.com) verrà utilizzato per ottenere le conoscenze necessarie a programmare per Android.
- Il sito www.arpa.veneto.it fornirà i dati su cui sarà basata l'applicazione.
- Il sito blogs.unive.it/groups/software_engineering_2012 per la consultazione delle slides del corso e per la pubblicazione dei deliverables.

Sul Play Store sono già presenti alcune applicazioni che offrono alcune delle funzionalità da noi pensate, tra queste:

- **Meteo Allergie (ai pollini):** L'applicazione fornisce le previsioni in tempo reale sulle concentrazioni dei pollini sull'intero territorio italiano. Notiamo però che:
 - L'app non offre una visualizzazione a mappa.
 - Secondo le recensioni, oltre ad avere una grafica poco accattivante, l'applicazione funziona male.
- **Meteo Arpa Veneto:** L'applicazione fornisce diversi dati meteo relativi alla regione Veneto, nello specifico, per quanto riguarda i pollini, queste funzionalità sono a pagamento.
- **App ARPAV Pollini:** L'applicazione offre una visualizzazione dei livelli di concentrazione di pollini per le principali famiglie allergeniche presenti nel territorio della regione Veneto. Secondo le recensioni, l'app non funziona.
- **mappa pollini e meteo Italia:** L'applicazione offre il monitoraggio meteo della propria città, tra questo sono presenti indicazioni sul livello di pollini in base ai mesi dell'anno. Secondo le recensioni, l'applicazione non si avvia.
- **Pollen:** L'applicazione mette a disposizione un servizio di monitoraggio dei pollini molto simile a quello da noi pensato, tuttavia in questo caso le aree coperte sono: Austria, Germania, Svizzera, Svezia.

1.5 Definizioni e abbreviazioni

- **Open data:** sono dati liberamente accessibili a tutti, le cui eventuali restrizioni sono l'obbligo di citare la fonte o di mantenere la banca dati sempre aperta.
- **Telegram:** è un servizio di messaggistica istantanea basato su cloud disponibile gratuitamente per diverse piattaforme.
- **Android:** è un sistema operativo per dispositivi mobili sviluppato da Google.
- **API(Application Programming Interface):** un insieme di procedure disponibili al programmatore.
- **KitKat:** è la versione 4.4/4.4.4 del sistema operativo Android.
- **IDE:** ambiente di sviluppo integrato che aiuta i programmatori nello sviluppo del codice sorgente dei programmi.
- **Android Studio:** Android Studio è un ambiente di sviluppo integrato (IDE) per lo sviluppo per la piattaforma Android.
- **Git:** è un software di controllo versione distribuito(VCS).
- **Bitbucket:** è un servizio di hosting per progetti software che utilizzano Git come sistema di controllo versione.
- **Commit:** è un insieme di modifiche che hanno portato l'applicazione ad un nuovo stato.
- **Bug:** errore nella scrittura del codice sorgente che comporta comportamenti anomali del software.
- **Alpha:** versione di un software in fase di sviluppo le cui funzionalità non sono ancora state implementare completamente, spesso queste versioni sono affette a bug.
- **Beta:** versione non definitiva di un software, già testata da esperti disponibile ad un numero maggiore di utenti.
- **Testing:** indica l'attività di verifica e collaudo del software.
- **Release:** è una specifica versione di un software resa disponibile ai suoi utenti finali. E' univocamente identificata da un numero in modo da distinguerla dalle release del software precedenti e future.
- **Google Play:** è un negozio virtuale online di applicazioni, brani musicali, pellicole cinematografiche, libri e riviste sviluppato da Google per offrire servizi ai dispositivi mobili Android.

2 Organizzazione del progetto

2.1 Modello del processo

Il modello al quale vogliamo fare riferimento è una combinazione di caratteristiche appartenenti al "modello a cascata" e al "modello evolutivo". La buona visibilità offerta dal modello a cascata andrà a colmare le problematiche di visibilità presenti nel modello evolutivo grazie alla stesura di documenti il più dettagliati possibile riguardo allo sviluppo del progetto e, la presenza di uno o più prototipi durante la fase di sviluppo, farà in modo di rendere la creazione del nostro software un'attività controllata pur cercando di mantenere le linee guida delineate nel piano di progetto. Essendo il nostro sistema nuovo, il modello evolutivo assicurerà anche un basso rischio per quel che riguarda problemi di specifiche e di progettazione.

2.2 Struttura organizzativa

Il gruppo è formato da: Gianluca Bigaglia, Luca Daniel, Lorenzo Giudice, Andrea Missiato. Non sarà presente un Project Manager, essendo un gruppo ristretto e fidato non ci saranno problemi nella suddivisione del lavoro e si applicherà la democrazia decentralizzata.

Per le comunicazioni è stato creato un gruppo su Telegram che permetterà la condivisione dei documenti. Per quanto riguarda invece il codice utilizzeremo il tool Git con un repository su Bitbucket.

2.3 Interfacce organizzative

Le principali interazioni con esterni, durante la realizzazione del progetto saranno:

- **Committente:** nel nostro caso il prof. Agostino Cortesi.
- **Beta-tester:** abbiamo selezionato tra le persone presenti nell'ambiente universitario, colleghi con allergie ai pollini, che testeranno a fondo tutte le funzionalità della nostra applicazione, inoltre, verranno coinvolti i nostri familiari ed amici.
- **Utenti finali:** dai quali otterremo recensioni e consigli per analizzare e correggere i punti negativi della nostra applicazione.

3 Processi gestionali

3.1 Obiettivi e priorità

Riconosciamo i seguenti obiettivi:

- Sviluppo di un'applicazione funzionante, efficiente ed intuitiva nell'utilizzo, che vada in contro alle esigenze delle persone con patologie allergiche.
- Rispetto delle scadenze previste (vedi sottosezione 1.2).
- Venire in contro ai feedback negativi degli utenti finali con l'obiettivo di migliorare nel miglior modo possibile l'esperienza utente.

3.2 Assunzioni, dipendenze, vincoli

- **Assunzioni**

- Buona conoscenza della programmazione orientata agli oggetti.
- Avere tutti i dati necessari per lo sviluppo dell'applicazione.
- Ogni componente ha gli strumenti necessari allo sviluppo.

- **Dipendenze**

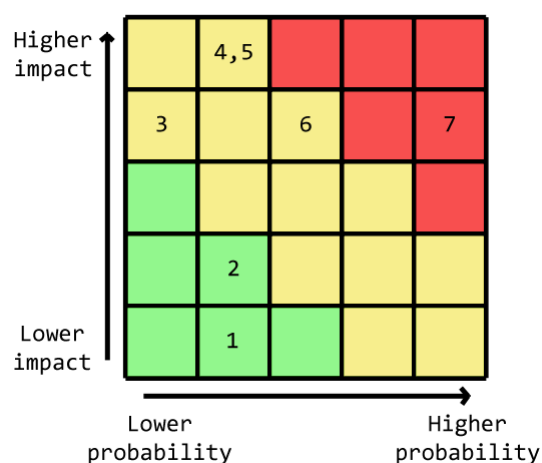
- Sviluppare il progetto in contemporanea allo studio per superare gli altri esami del semestre.
- Velocità nell'apprendimento dei linguaggi e degli ambienti di sviluppo a noi ora parzialmente sconosciuti e utili al raggiungimento degli obiettivi prefissati.
- Organizzazione del lavoro e degli incontri in base agli impegni dei vari componenti del gruppo.

- **Vincoli**

- Rispettare i termini di consegna.
- Seguire le specifiche linee guida definite all'interno del suddetto documento.
- Riuscire a completare il progetto entro i primi giorni di Gennaio, poiché un componente del gruppo ha bisogno di laurearsi a Febbraio.

3.3 Gestione dei rischi

N	Rischi	Prob.	Impatto	Soluzioni
1	Malattia di un componente del gruppo	Bassa	Irrelevante	Lavoro da casa
2	Malfunzionamento del PC personale	Bassa	Marginale	PC alternativi
3	Dimissioni di un componente	Molto Bassa	Critico	Aumentare il carico di lavoro
4	Litigi interni	Bassa	Catastrofico	Scisma del gruppo
5	Corruzione dei dati	Bassa	Catastrofico	Backup periodico dei dati
6	Ritardo nella consegna	Media	Critico	Impostazione di scadenze interne
7	Impossibilità di incontro fisico	Alta	Critico	Incontro virtuale



3.4 Meccanismi di monitoraggio e di controllo

Utilizzando *Telegram* riusciremo a tenerci aggiornati sullo sviluppo del progetto, inoltre, prefissando delle scadenze interne al gruppo almeno settimanali, sarà possibile rispettare le deadline di progetto. Ogni martedì ed ogni giovedì ci incontreremo per controllare la giusta evoluzione del progetto (sia a livello di documentazione che a livello di applicazione).

Lo sviluppo dell'applicazione verrà monitorato attraverso l'albero dei commit di Git.

Oltre alle riunioni che organizziamo settimanalmente per aggiornare lo stato di avanzamento del progetto, il nostro gruppo si frequenta al di fuori

dell'orario universitario, offrendo quindi la possibilità di un monitoraggio giornaliero, magari meno specifico degli incontri programmati.

3.5 Pianificazione dello staff

Il nostro staff è composto dai seguenti ruoli:

- **Costruttori:** Sono i programmatori veri e propri con il compito di implementare le varie funzionalità necessarie alla buona esecuzione dell'applicazione.
- **Progettisti:** Hanno voce in capitolo riguardo le decisioni su *Milestones*, suddivisione dei compiti e preferenze riguardanti le varie strade da percorrere.
- **Coordinatore/Progettista:** Un progettista che oltre ai suoi compiti standard, svolge un'attività di controllo qualità/correttezza e si occupa di impostare delle scadenze interne. Pur facendo ciò non ricopre un ruolo da Manager ma si occupa solamente della buona organizzazione delle attività. Inoltre, si occupa delle interazioni con il committente, i beta-tester e gli utenti finali.

Al di là del **coordinatore/progettista**, che viene definito all'inizio del progetto e rimane tale fino alla consegna finale, gli altri membri del gruppo possono ricoprire sia il ruolo di **costruttore** che il ruolo di **progettista**.

4 Processi tecnici

4.1 Metodi, strumenti e tecniche

Ogni componente del gruppo utilizzerà il proprio PC per lavorare, non tutti i componenti utilizzano lo stesso sistema operativo, nello specifico verranno impiegati: Windows e GNU/Linux. I linguaggi utilizzati per sviluppare l'applicazione sono Java e XML; per la gestione del database SQLite (interno all'app) verrà utilizzato invece il linguaggio SQL.

Per facilitare la programmazione verrà utilizzato l'ambiente di sviluppo Android Studio. Per le comunicazioni verrà utilizzato Telegram e per la condivisione del codice Git. Ogni settimana verranno pianificate le riunioni che possono essere fisiche o virtuali, nel secondo caso tramite i tool TeamViewer/TeamSpeak3.

4.2 Documentazione del software

Attualmente, non avendo ancora implementato alcuna funzionalità, non è presente una documentazione che spieghi l'utilizzo della nostra applicazione. Successivamente, verrà redatta sulla base di quanto prodotto e consegnata

nel momento in cui verrà richiesta un'implementazione più accurata. Contiamo di sviluppare la documentazione man mano che ogni funzionalità viene implementata, ottenendo fin da subito una versione provvisoria e comunque completa per spiegare le funzionalità del nostro prodotto al cliente.

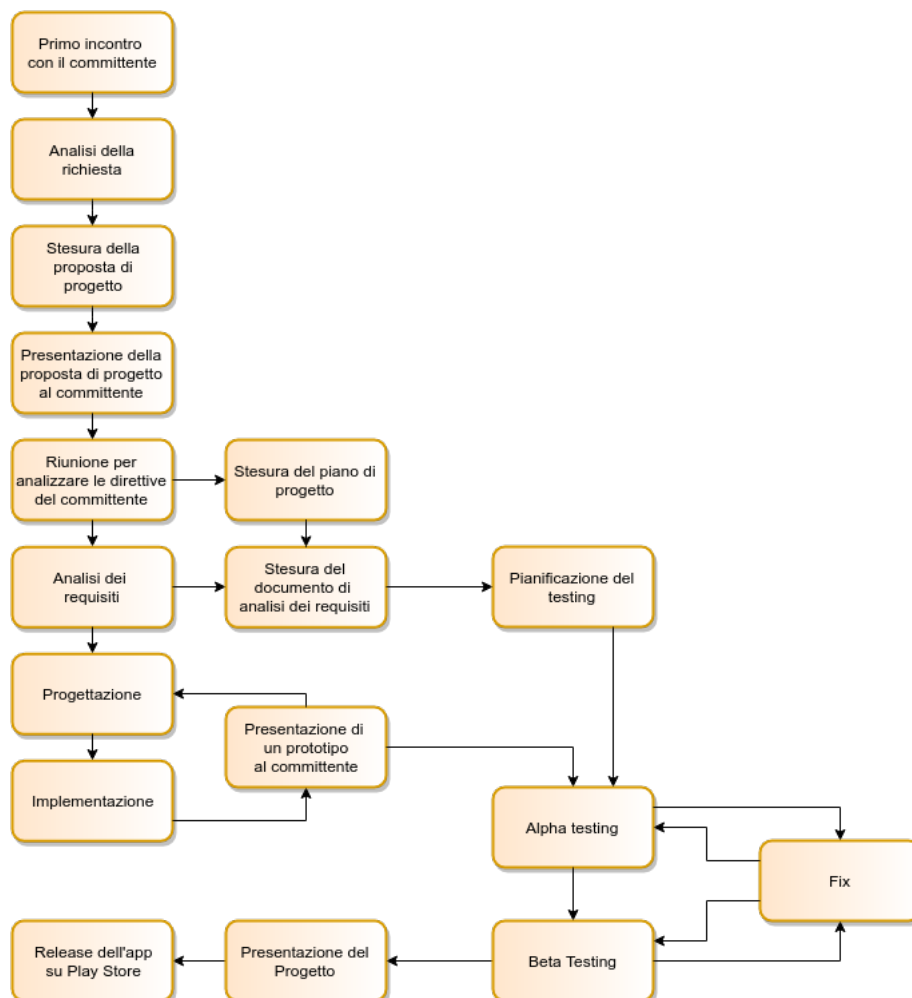
Verrà redatto un documento generale che illustrerà gli agenti dell'applicazione, a livello più basso troviamo per ogni attività/classe implementata un commento iniziale che ne descrive l'utilizzo, allo stesso modo prima di ogni metodo verrà scritto un commento che ne spiega il compito e definisce come vengono utilizzate le variabili.

5 Pianificazione del lavoro, delle risorse umane, del budget

5.1 WBS(Work Breakdown Structure)

1. Proposta di progetto
 - (a) Primo incontro con il committente.
 - (b) Analisi della richiesta.
 - (c) Stesura della proposta di progetto.
 - (d) Presentazione della proposta di progetto al committente.
2. Piano di Progetto
 - (a) Riunione per analizzare le direttive del committente.
 - (b) Stesura del piano di progetto.
3. Analisi dei requisiti
 - (a) Raccolta e analisi dei requisiti.
 - (b) Stesura del documento riguardante i requisiti.
4. Progettazione
 - (a) Organizzazione della progettazione.
 - (b) Stesura di un documento di progettazione.
 - (c) Progettazione architettura.
 - (d) Progettazione interfaccia.
5. Implementazione dell'app.
 - (a) Implementazione delle funzionalità di base.
(Integrazione orizzontale)
 - (b) Presentazione di un prototipo.
 - (c) Adattamento delle funzionalità alle impressioni del committente.

- (d) Vai al punto 5 finché non finisce l'implementazione.
- 6. Fase di testing.
 - (a) Pianificazione del testing.
 - (b) Alpha-testing(interno al gruppo).
 - (c) Beta-testing(esterno al gruppo).
 - (d) Fix.
 - (e) Vai al punto 6b o 6c finché i test non sono soddisfacenti.
- 7. Presentazione del progetto.
- 8. Release dell'app sul Play Store.



Per ogni attività presente troviamo una sessione di monitoraggio.

5.2 Dipendenze

Dati i punti del capitolo precedente (sottosezione 5.1) sono presenti le seguenti dipendenze:

Attività	Dipendenza
1.a	-
1.b	1.a
1.c	1.b
1.d	1.c
2.a	1
2.b	2.a
3.a	2
3.b	3.a
4	3
4.b	4.a
5.a, 5.b	4,2
5.c	5.a, 5.b
6.a	3.b
6.b, 6.c	6.a
7	6
8	7,6

5.3 Risorse necessarie

Risorse Umane: Le persone che parteciperanno allo sviluppo dell'applicazione sono i quattro componenti del gruppo.

Software:

- Per la redazione della documentazione viene impiegato il linguaggio di markup $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$.
- Per la realizzazione di (eventuali) grafici verrà utilizzato il portale **draw.io** per il diagramma WBS e Pert, mentre per il diagramma di Gantt sarà utilizzato il software **GanttProject**.
- Per lo sviluppo dell'applicazione verrà utilizzato l'IDE **Android Studio**.
- Per la compilazione verrà impiegata la JDK Android.
- Per la gestione dei database SQLite oltre al gestore standard "SQLite3" verrà impiegato il tool grafico **SQLiteMan**.

Hardware: L'applicazione in fase di sviluppo verrà testata sui seguenti dispositivi android:

- Xiaomi Mi5 con Android 6.0.x
- Nexus 5x con Android 7.0.x
- Dispositivi virtuali forniti da Android Device Manager.

Mentre il software verrà sviluppato sui seguenti PC:

- ASUS N550JK
- ASUS N552VW
- Dell Inspiron 7559

5.4 Allocazione del budget e delle risorse

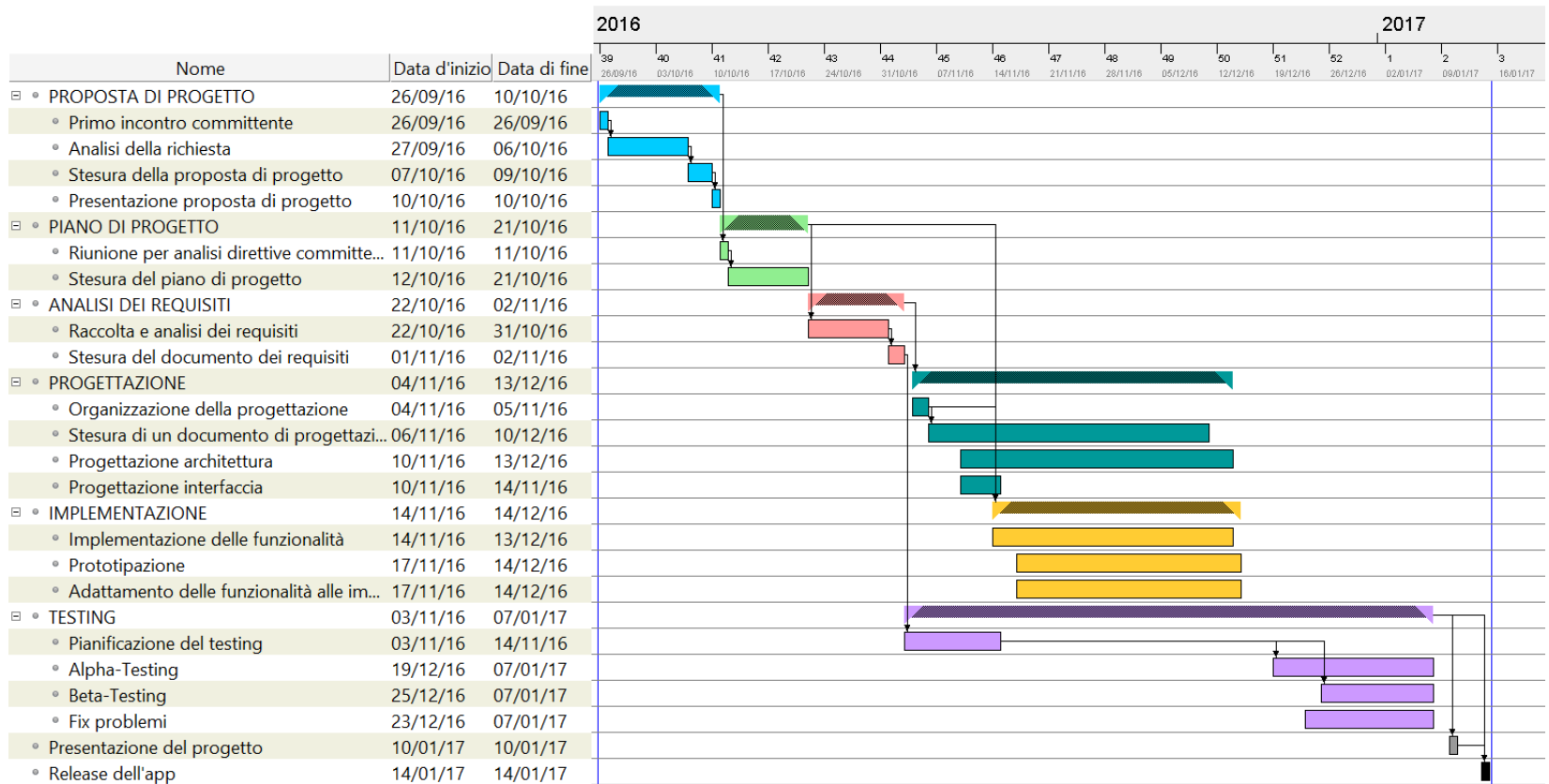
Lo sviluppo dell'applicazione non richiede nessun tipo di risorsa economica, in quanto i software impiegati (Android Studio, Git, Telegram, TeamViewer, TeamSpeak3) sono tutti gratuiti. I dati su cui si baserà l'applicazione sono anch'essi ad accesso libero. Eventualmente l'unico costo sarà quello relativo all'iscrizione al PlayStore come sviluppatore al fine di pubblicare l'applicazione finale.

Oltre ad un eventuale costo economico, la realizzazione dell'applicazione richiederà sicuramente un costo temporale indicato in Giorni/Uomo(g/u):

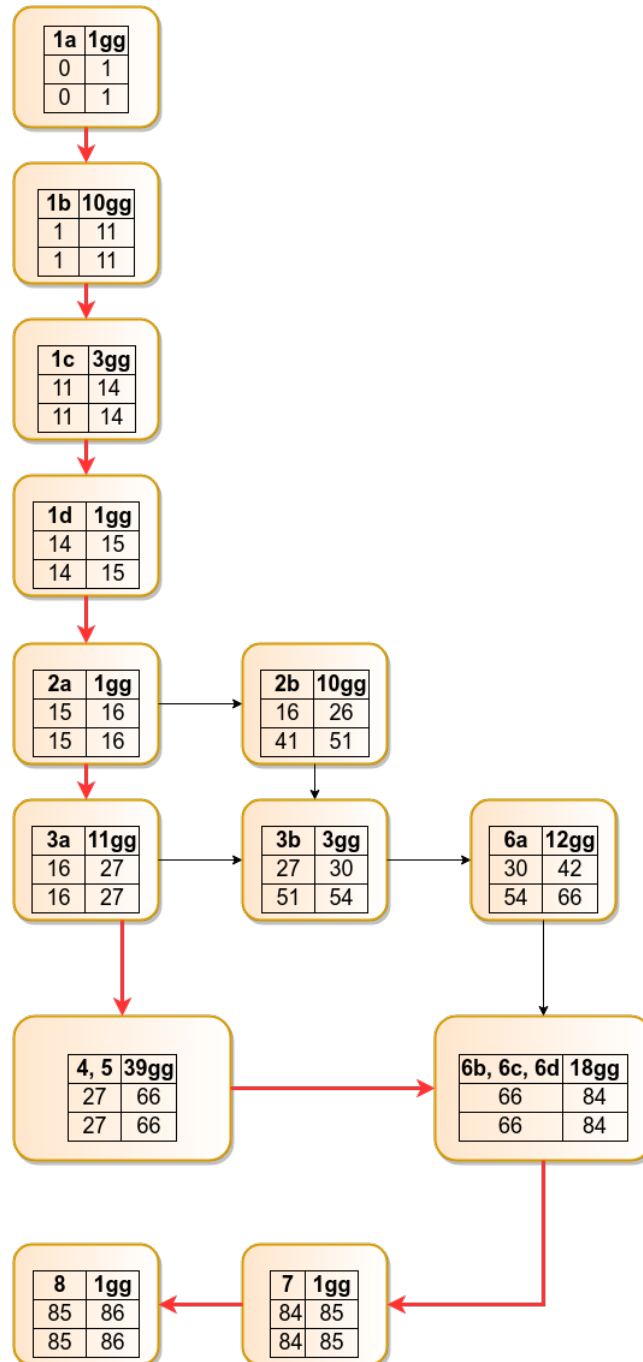
Attività	Durata(GG)	GG/Uomo
Primo incontro con il committente	1	4
Analisi della richiesta	10	40
Stesura della proposta di progetto	3	12
Presentazione proposta di progetto	1	4
Riunione per analisi delle direttive del committente	1	4
Stesura del piano di progetto	10	40
Raccolta e analisi dei requisiti	10	40
Stesura del documento riguardante i requisiti	2	8
Organizzazione della progettazione	2	8
Stesura di un documento di progettazione	35	140
Progettazione architettura	34	136
Progettazione interfaccia	5	20
Implementazione delle funzionalità	30	120
Prototipazione	28	112
Adattamento delle funzionalità alle impressioni del committente	28	112
Pianificazione del testing	12	48
Alpha-testing	20	80
Beta-testing	14	56
Fix Problemi	16	64
Presentazione del progetto	1	4
Release dell'app	1	4

5.5 Pianificazione

5.5.1 Diagramma di Gantt



5.6 Diagramma di Pert



5.6.1 Descrizione Pert

Nome attività	Giorni
ES	EF
LS	LF

- **Nome:** Nome dell'attività (si faccia riferimento a sottosezione 1.2).
- **Giorni:** Giorni per lo svolgimento e completamento dell'attività.
- **ES:** Earliest Start Time, è il minimo giorno di inizio dell'attività a partire dal minimo tempo necessario per le attività che precedono.
- **EF:** Earliest Finish Time, è il minimo giorno in cui l'attività può terminare ($ES + \text{Giorni}$).
- **LS:** Latest Start Time, è il giorno massimo in cui l'attività deve iniziare senza provocare ritardo ($LF - \text{Giorni}$).
- **LF:** Latest Finish Time, è il giorno massimo in cui l'attività deve iniziare senza provocare ritardo.

5.6.2 Milestones e Deadlines

Milestones	Deadline
Piano di progetto	21/10/2016
Documento di analisi e specifica dei requisiti	02/11/2016
Pianificazione testing con annesso documento	15/11/2016
Termine fase di testing	04/12/2016
Documento di progettazione	10/12/2016
Release	14/01/2017