



Università
Ca'Foscari
Venezia



ESIFunds

European Structural Investment
Funds made easy!

Universita' Ca' Foscari di Venezia

Dipartimento di Scienze Ambientali,
Informatica e Statistica

Corso di Ingegneria del
Software

A.A. 2017-2018

Docente: Prof. Agostino Cortesi

PIANO DI PROGETTO

Studenti

Gemin Francesco

Longhin Federico

Sponchiado Francesco

16/02/2018

Indice

Indice	2
1 Introduzione	3
1.1 Overview	3
1.2 Derivables	3
1.3 Evoluzione del Progetto	3
1.4 Materiale di Riferimento	3
1.5 Glossario	3
2 Organizzazione del Progetto	4
2.1 Modello del processo	4
2.2 Struttura organizzativa	4
2.3 Interfacce organizzative	5
2.4 Responsabilita' del progetto	5
3 Processi Gestionali	6
3.1 Obiettivi e priorit�	6
3.2 Assunzioni - dipendenze - vincoli	6
3.3 Tabella dei rischi	7
3.4 Grafico di Impatto	8
3.4.1 RMM	8
3.5 Meccanismi di monitoraggio e di controllo	10
3.6 Pianificazione dello staff	10
4 Metodi - Strumenti - Tecniche	11
4.1 Sistemi di calcolo	11
4.1.1 Android Studio - PRO	11
4.1.2 Android Studio - Contro	11
4.2 Metodi di Sviluppo	11
4.3 Documentazione del Software	12
4.4 Funzionalita' di supporto al progetto	12
4.4.1 Pianificazione della qualita'	12
4.4.2 Pianificazione della gestione delle configurazioni	12
5 Pianificazione del Lavoro	13
5.1 WBS	13
5.2 Dipendenze	14
5.3 Risorse necessarie	14
5.4 Allocazione del budget e delle risorse	15
5.5 Pianificazione	15

1 Introduzione

1.1 Overview

La proposta di progetto consiste nella realizzazione di un applicativo Android in ottica di miglioramento delle pubbliche amministrazioni. Nello specifico si andrà a realizzare uno strumento atto a rendere versatile ed interattiva la visualizzazione dei dati relativi alle opportunità di finanziamento dei fondi SIE.

1.2 Derivables

- 23/10/2017 - Piano di Progetto
- 09/11/2017 - Documento di Analisi e Specifica
- 23/11/2017 - Piano di Testing
- 15/12/2017 - Documento di Progettazione
- 31/01/2018 - Realizzazione e Messa in Linea

1.3 Evoluzione del Progetto

L'evoluzione del progetto prevede le seguenti fasi

- Progettazione dell'applicazione.
- Sviluppo dell'applicazione
 - Layout Grafico.
 - Elaborazione Open Data prescelti.
- Stesura documentazioni richieste.

Le varie macro aree verranno sviluppate in modo incrementale revisionando il piano di progetto durante lo sviluppo.

1.4 Materiale di Riferimento

- Open Data
- Slide del Corso di Ingegneria del Software
- Android Studio Developer Website

1.5 Glossario

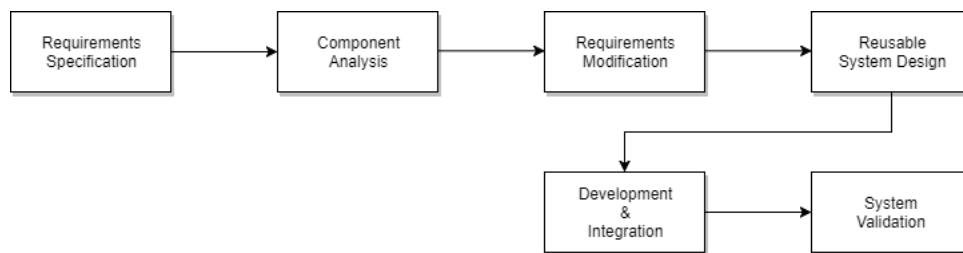
- **Fondi SIE (ESI)**: Fondi strutturali e di investimento europei.
- **GUI**: Graphical User Interface

2 Organizzazione del Progetto

2.1 Modello del processo

- L'applicativo da sviluppare e' di complessita' medio/alta.
- Solo parte del team ha precedentemente lavorato in ambiente Android Studio.
- Solo parte del team ha precedentemente realizzato un applicativo Android.

Sulla base di tali premesse si e' optato per l'adozione di un modello di processo basato sul riuso. Questo modello permette la facile ed immediata integrazione di componenti "Off the Shelf" largamente utilizzati in un ambiente votato alla modularita' come Android. L'evoluzione dell'applicativo non richiedera' un eccessivo aumento dei costi aggiuntivi richiedendo lo studio da parte del team di librerie in continua evoluzione con tempi relativamente brevi.



2.2 Struttura organizzativa

Il team ha optato per un'organizzazione di tipo democratico decentralizzato. Questa organizzazione e' sconsigliata in caso di un gran numero di componenti (Problema di scalabilita'), essendo pero' il gruppo formato da solo tre membri, probabilmente per tutto il ciclo di produzione del progetto, pensiamo sia la scelta migliore per prendere decisioni in maniera semplice ed efficace. Il ruolo del leader e' ricoperto dall'intero team che per ogni decisione si affidera' all'esito di una votazione tra tutti i membri. Durante il processo di sviluppo non verranno assegnati ruoli specifici, in base alle necessita' ogni membro potra' essere assegnato a compiti differenti. Nel caso in cui un membro del gruppo risultasse piu' competente degli altri a svolgere determinate attivita' il resto del team puo' prendere la decisione di affidargli un potere decisionale maggiore in quella determinata area di competenza.

2.3 Interfacce organizzative

Durante la realizzazione del progetto potrà risultare necessaria la comunicazione con entità esterne al team per motivi quali richieste di informazioni o feedback.

- Forum di supporto Android Developers
- Feedback da parte del professor A. Cortesi.
- Bacino di utenti (Tecnici) per la fase di testing dell'applicativo.
- Bacino di utenti (Non tecnici) per la fase di testing dell'applicativo.

2.4 Responsabilità del progetto

Avendo adottato un'organizzazione di tipo democratico decentralizzato la responsabilità sarà suddivisa in maniera equa tra tutti i membri. Ogni componente avrà comunque delle scadenze da rispettare che verranno controllate dagli altri.

3 Processi Gestionali

3.1 Obiettivi e priorit 

Obiettivi

- Sviluppare un applicativo funzionante in Android.
- Rispettare i tempi di consegna del prodotto e dei relativi derivatives entro le varie date prefissate.

Priorit 

- Parserizzare nel modo corretto gli Open Data che si e' deciso di utilizzare.
- Portare tutto il team al medesimo livello di conoscenza dell'ambiente di sviluppo Android Studio.
- Pubblicare un software di qualita'

3.2 Assunzioni - dipendenze - vincoli

Assunzioni

- Tutti gli utenti utilizzano dispositivi Android.
- Il dispositivo utilizzato dagli utenti possiede un accesso alla rete Internet.

Dipendenze

- Utilizzo di Smartphone/Tablet forniti di sistema operativo Android.
- La composizione del team non subira' cambiamenti. (Se non minimi e temporanei)

Vincoli

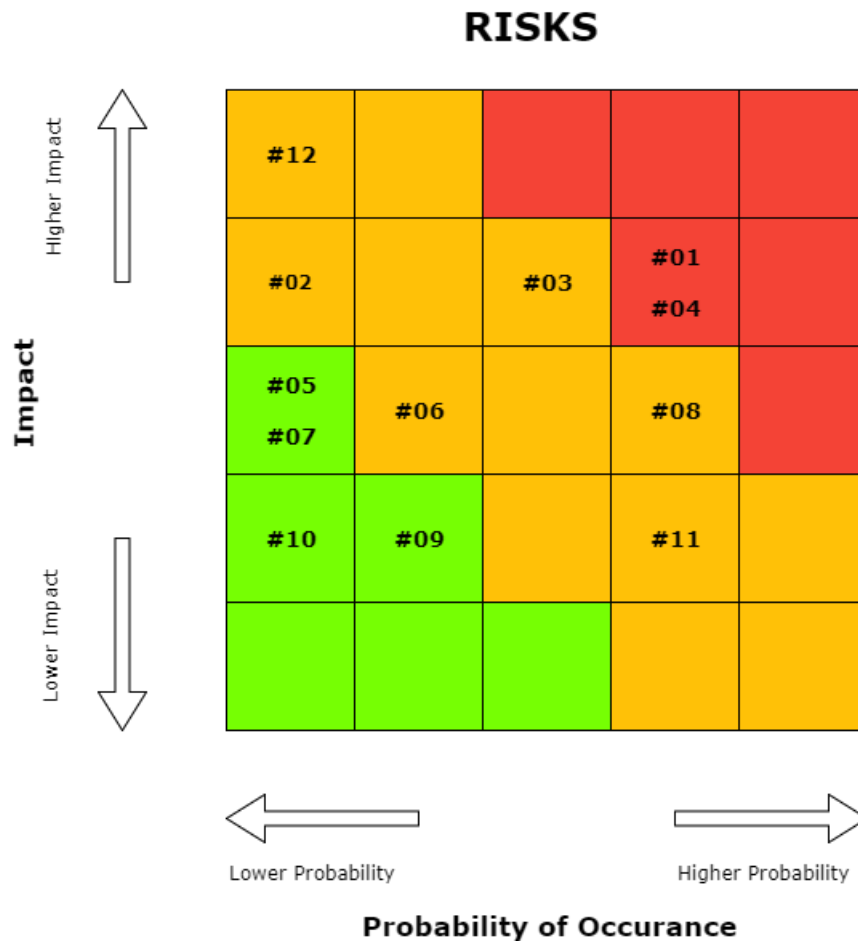
- Preparazione di esami per altri corsi.
- Il corretto funzionamento dell'applicativo e' vincolato dalla correttezza dei dati prescelti per svilupparlo.
- Conoscenza di altri linguaggi di programmazione per implementare servizi aggiuntivi.
- Realizzazione e messa in linea in data 31 gennaio.

3.3 Tabella dei rischi

- **PS:** Product size.
- **ST:** Staff size and experience.
- **DE:** Development environment.

	Rischio	Categoria	Probabilità	Impatto	RMM
# 01	Cattiva stima della taglia del progetto	PS	80 %	2	# 01
# 02	Mancanza di figure specializzate aumentano il numero di difetti richiedendo un complesso processo di debugging	ST	15 %	2	# 02
# 03	Parti del prodotto richiedono più tempo per essere implementate dato la poca familiarità con quest'ultime	DE	50 %	2	# 03
# 04	L'ambiente richiede uso di un Database	DE	80 %	2	# 04
# 05	Parti di codice separatamente sviluppate non possono essere integrate facilmente richiedendo modifiche anche sostanziali	DE	10 %	3	# 05
# 06	Parti di codice separatamente sviluppate non sono corrette richiedendo una nuova fase di stesura	DE	25 %	3	# 06
# 07	Sviluppo di funzionalità ausiliarie non necessarie al prodotto finito	DE	05 %	3	# 07
# 08	Rigorose richieste di compatibilità con sistemi esistenti richiedono più tempo di testing, design e implementazione di quanto preventivato	DE	80 %	3	# 08
# 09	Il processo di sviluppo in un ambiente non familiare causa problematiche impreviste	DE	35 %	4	# 09
# 10	I vari membri del team non lavorano bene insieme	ST	05 %	4	# 10
# 11	Personale chiave non è sempre disponibile	ST	80 %	4	# 11
# 12	Perdita di membri del team durante lo sviluppo del progetto	ST	05%	1	# 12

3.4 Grafico di Impatto



3.4.1 RMM

- Rischio 01
 - Nuova analisi delle specifiche necessarie per la realizzazione del progetto
 - Nuova stima della complessita' del progetto
 - Nuova stima della taglia del progetto
- Rischio 02
 - Riconoscimento della mancanza di conoscenza della figura

- Acquisizione di conoscenze specifiche da parte della figura interessata
- Conseguente stesura delle funzionalita' precedentemente errate
- Rischio 03
 - Acquisizione di familiarita' con le parti del prodotto in questione
 - Conseguente stesura delle parti in tempo minore
- Rischio 04
 - Progettazione di un Database adatto all'applicativo
 - Creazione del Database dato il progetto
 - Messa in linea del Database in questione
- Rischio 05
 - Analisi delle problematiche di integrazione
 - Risoluzione delle problematiche di integrazione
 - Integrazione delle parti di codice separatamente create
- Rischio 06
 - Analisi delle parti di codice non corrette
 - Correzione degli errori
 - Conseguente stesura delle parti in modo tale che siano corrette
- Rischio 07
 - Aggiornamento dei documenti per integrare le funzionalita' aggiuntive
- Rischio 08
 - Analisi dei problemi di compatibilita' con i sistemi esistenti
 - Risoluzione dei problemi di compatibilita' con i sistemi esistenti
 - Implementazioni delle funzionalita' che precedentemente avevano creato problemi di compatibilita'
- Rischio 09
 - Studio approfondito dell'ambiente di sviluppo
 - Risoluzione delle problematiche impreviste che si erano precedentemente presentate
- Rischio 10
 - Meeting e attivita' di team building per risolvere questa tipologia di problematiche

- Rischio 11
 - Scheduling delle attività del personale non sempre disponibile in accordo con gli orari desiderati
- Rischio 12
 - Analisi delle task che i membri del gruppo mancanti dovevano eseguire
 - Ricerca e assunzione di personale in grado di risolvere i task che i membri del gruppo mancanti hanno lasciato incompleti
 - Consegna del progetto entro i tempi stabiliti

3.5 Meccanismi di monitoraggio e di controllo

I controlli sulla qualità del lavoro svolto dal team saranno di due tipi:

- **Controlli pianificati:** Almeno un controllo generale ogni settimana. La frequenza può aumentare a seconda dei problemi rilevati durante la settimana precedente.
- **Controlli a discrezione del singolo membro:** Ogni componente del team potrà a propria discrezione effettuare controlli di vario genere sul progetto.

Ogni membro del team al termine di ogni modifica o implementazione di nuove funzionalità è tenuto a commentare e/o segnalare eventuali TODO nel proprio codice in modo tale da rendere più comprensibile anche agli altri membri il lavoro svolto.

I canali di comunicazione preferenziali saranno: Telegram, WhatsApp, Discord. Per evitare la perdita dei dati verranno utilizzati strumenti di versioning e archiviazione online come Google Drive e GitHub.

3.6 Pianificazione dello staff

È gradita ma non strettamente necessaria una buona familiarità con gli strumenti.

- Google Sheets, Google Docs, Draw.io
- Android Studio

Le abilità mancanti verranno colmate attraverso i seguenti metodi

- Lettura di testi tecnici specifici per la progettazione e l'implementazione di applicazioni per la piattaforma Android
 - Ricerca online di corsi specifici per la programmazione di applicativi Android
 - Consultazione con il tutor di laboratorio di Ingegneria del Software

4 Metodi - Strumenti - Tecniche

4.1 Sistemi di calcolo

Il team sviluppera' il progetto attraverso elaboratori su piattaforma Windows 10 o Ubuntu. In tali elaboratori dovra' essere installato l'ambiente di sviluppo necessario per la realizzazione dell'applicativo.

4.1.1 Android Studio - PRO

- Ambiente di sviluppo completo e pronto all'uso con software di virtualizzazione del dispositivo integrato.
- Rende possibile sviluppare applicati per ogni Smartphone/Tablet con sistema operativo Android indipendentemente da casa produttrice e hardware.
- E' disponibile gratuitamente per Windows, Linux e Mac

4.1.2 Android Studio - Contro

- Non e' possibile la portabilita' per altre piattaforme come iOS, Windows Phone, ecc.

L'applicativo potra' essere testato sui dispositivi fisici dei developers e su quelli dei tester scelti, oltre che su dispositivi emulati direttamente da Android Studio.

4.2 Metodi di Sviluppo

Lo sviluppo dell'applicativo sara' completando attraversando molteplici fasi di prototipazione, ciascuna delle quali necessaria per aggiungere una particolare funzionalita'. Ogni ciclo di prototipazione consiste principalmente in quattro fasi:

- Analisi della funzionalita' da implementare
- Progettazione della funzionalita'
- Implementazione della funzionalita'
- Testing della funzionalita'

Quando tutte le funzionalita' minime previste dal documento di specifica dell'applicativo saranno implementate esso sara' pronto per essere pubblicato.

4.3 Documentazione del Software

I documenti che saranno realizzati dal team possono essere raggruppati nella categoria dei documenti di progetto:

- Piano di progetto
- Piano di testin
- Documento dei requisiti
- Documento di progettazione
- Commenti presenti all'interno del codice sorgente dell'applicativo stesso

4.4 Funzionalità di supporto al progetto

4.4.1 Pianificazione della qualità

Punti di qualità che devono essere rispettati per assicurare una buona riuscita del progetto:

- Mantenere aggiornata la documentazione perseguendo uno standard qualitativo di alto livello.
- Rispettare quanto definito come metodologia di sviluppo.
- Analizzare i problemi con metodo e rispettare gli standard di buona programmazione.
- Rispettare i requisiti funzionali e non che verranno definiti in corso d'opera.
- Attuare una politica di testing critico in modo tale da individuare possibili problemi ed errori prima della pubblicazione.

4.4.2 Pianificazione della gestione delle configurazioni

- Tenere traccia di tutte le versioni dell'applicativo prodotte, in modo tale da poter risalire ai cambiamenti effettuati a fronte di determinati problemi e poter eseguire rollback di versione in caso di successivo malfunzionamento.
- Documentare dettagliatamente ogni cambiamento delle versioni nella documentazione in modo chiaro e dettagliato rendendo più semplice il lavoro di gruppo.

5 Pianificazione del Lavoro

5.1 WBS

- A = Proposte di idee per l'applicativo.
- B = Definizione struttura del progetto.
- C = Ricerca materiale statistico (Open Data).
- D = Definizione struttura del documento.
- E = Apprendimento programmazione Android.
- F = Definizione schema UML dell'applicativo.
- G = Definizione pattern.
- H = Definizione GUI dell'applicativo.
- I = Decisione ruoli interni.
- J = Stesura codice sorgente dell'applicativo.
- L = Stesura documento di analisi e specifica dei requisiti.
- M = Progettazione dei test di usabilita'.
- N = Svolgimento dei test di usabilita'.
- O = Raccolta statistiche usabilita'.
- P = Consegna piano di testing.
- Q = Stesura documento finale di progettazione.
- R = Consegna del progetto.

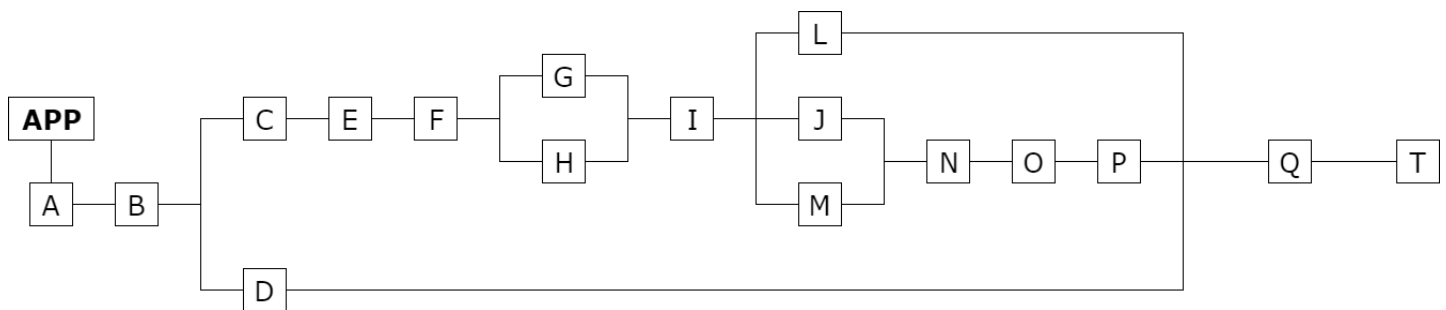


Diagramma di Gantt
Diagramma di Pert

5.2 Dipendenze

Mansioni	Durata in Giorni	Dipendenze
A	07	-
B	07	A
C	06	B
D	02	B
E	10	C
F	02	E
G	07	F
H	02	F
I	01	G - H
J	42	I
L	04	I
M	07	I
N	12	J - M
O	07	N
P	01	O
Q	04	P - D
R	01	Q

5.3 Risorse necessarie

Le risorse necessarie per la realizzazione del progetto possono essere raggruppate nelle seguenti categorie:

- **Risorse Umane:** Sviluppatori (Membri del team). Non e' prevista l'introduzione di ulteriori componenti all'interno del team, una volta raggiunta la fase di testing dell'applicativo potrebbe essere necessario reclutare un campione di utenti.
- **Risorse Hardware:** Ogni membro del team necessitera' di un calcolatore con connessione Internet.
- **Risorse Software:** Ogni calcolatore dovra' essere munito di una versione stabile e funzionante di Android Studio.

5.4 Allocazione del budget e delle risorse

Lo sviluppo di un applicativo Android da parte di un team di studenti non richiede nessuna tipologia di risorsa economica in quanto i software impiegati e i dati utilizzati sono entrambi totalmente gratuiti. Saranno inoltre necessari dei costi temporali indicati in giorni. (Vedi tabella delle Dipendenze)

5.5 Pianificazione

La pianificazione del progetto seguirà la stesura dei deliverables da consegnare nelle scadenze stabilite con il professor A. Cortesi durante lo svolgimento del corso di Ingegneria del Software.

- 23/10/2017 - Piano di Progetto
- 09/11/2017 - Documento di Analisi Specifica
- 23/11/2017 - Piano di Testing
- 15/12/2017 - Documento di Progettazione
- 31/01/2018 - Realizzazione e Messa in Linea

Durante il periodo di sviluppo dell'applicativo i documenti redatti verranno rivisti, corretti e aggiornati.