



Università
Ca' Foscari
Venezia

Dipartimento di Scienze Ambientali, Informatica e Statistica

Piano di progetto

Ingegneria del Software 2017/18

Docente del corso: A. Cortesi

Tutore del corso: A. Spanò

Data di consegna: 31/01/2018

V. 2.0

Corazza Sara 857682

Mutterle Ilaria 860720

Rigon Daniele 857319

Ruffato Denny 859171

Indice

1	Introduzione	4
1.1	Descrizione del progetto	4
1.2	Deriverables del progetto	4
1.3	Evoluzione del progetto	5
1.4	Materiale di riferimento	5
1.5	Glossario	6
2	Organizzazione del progetto	8
2.1	Modello del processo	8
2.2	Struttura organizzativa	8
2.3	Interfacce organizzative	9
2.4	Responsabilità di progetto	9
3	Processi gestionali	10
3.1	Obiettivi e priorità	10
3.2	Assunzioni, dipendenze, vincoli	10
3.3	Gestione dei rischi	11
3.4	Meccanismi di monitoraggio e di controllo	12
3.5	Pianificazione dello staff	12
4	Processi tecnici	14
4.1	Metodi, strumenti e tecniche	14
4.1.1	Sistemi di calcolo	14
4.1.2	Software utilizzati	14
4.2	Documentazione del software	15
4.3	Funzionalità di progetto	15
4.3.1	Pianificazione della qualità	15
4.3.2	Pianificazione della gestione delle configurazioni	15
5	Pianificazione del lavoro, delle risorse umane e del budget	17
5.1	WBS (Work breakdown structure)	17

<i>INDICE</i>	3
5.2 Dipendenze	18
5.3 Risorse necessarie	19
5.4 Allocazione del budget e delle risorse	19
5.4.1 STIMA COSTI CONSUNTIVI	20

Introduzione

1.1 Descrizione del progetto

Il progetto che abbiamo intenzione di presentare, la cui realizzazione prevista nel corso di Ingegneria del Software, svolto durante il Corso di Laurea in Informatica presso l'Università Ca' Foscari di Venezia e tenuto dal Professore Agostino Cortesi, nell'a.a. 2017/2018, consiste in un'applicazione Android volta a sfruttare gli open data a livello regionali e nazionale.

Nel nostro caso prendiamo in considerazione i dati riguardanti tutto il suolo nazionale.

La nostra applicazione darà la possibilità di scegliere una tipologia di carburante (GPL, benzina, gasolio e diesel) e visualizzare i distributori che lo forniscono.

Per aiutare l'utente abbiamo deciso di implementare una navigazione veloce che traccia il percorso ottimale tra la posizione attuale e il distributore prescelto.

In base a questo, l'idea potrebbe affiancarsi ad altri corsi della laurea riguardanti l'ottimizzazione di questo argomento.

Lo scopo dell'applicazione è di aiutare l'utente nella ricerca e/o nella pianificazione di un viaggio in base ai distributori che forniscono un determinato carburante.

1.2 Deriverables del progetto

Le scadenze previste per il progetto sono le seguenti:

- 25/09/2017 definizione gruppi di lavoro
- 23/10/2017 consegna del piano di progetto

- 02/11/2017 consegna dei documenti di analisi e specifica dei requisiti
- 15/11/2017 piano di testing
- 10/12/2017 consegna documento di progettazione
- 31/01/2018 completamento della realizzazione dell'applicazione e messa in linea

1.3 Evoluzione del progetto

Dal momento che ci troviamo ancora nella fase iniziale del progetto, probabilmente ulteriori funzionalità verranno stabilite in seguito e alcune di quelle già pensate potrebbero venire modificate o migliorate in seguito.

Noi, comunque, ci impegneremo in modo da sviluppare i concetti proposti nella prima parte di questa introduzione ed approfondiremo successivamente la loro descrizione.

L'implementazione è ancora da definire, ma sarà oggetto di *testing* e verrà sicuramente trattata in future versioni del piano di progetto e nel resto della documentazione che sarà presentata.

1.4 Materiale di riferimento

Documenti a cui faremo riferimento nel piano di progetto:

- Slides del corso di Ingegneria del Software del Prof. Cortesi
- Codice messo a disposizione dal Tutor
- Sito web per sviluppatori Android
<https://developer.android.com/guide/index.html>
- Manuale guida per LaTeX 1
<http://www.sharelatex.com/learn>
- Manuale guida per LaTeX 2
<http://www.guit.sssup.it/downloads/LaTeX-facile.pdf>
- Guida per l'utilizzo di SQLite all'interno dell'applicazione Android
https://www.mrwebmaster.it/android/generazione-utilizzo-database-sql-lite_12428.html

- Anagrafica degli impianti di distribuzione di carburanti <http://www.sviluppoeconomico.gov.it/index.php/it/open-data/elenco-dataset/2036944-carburanti-archivio-prezzi>

1.5 Glossario

Android : sistema operativo per dispositivi mobili sviluppato da Google Inc. e basato sul kernel Linux. Si tratta di un sistema embedded progettato principalmente per smartphone e tablet, con interfacce utente specializzate per televisori (Android TV), automobili (Android Auto), orologi da polso (Android Wear), occhiali (Google Glass);

AndroidStudio : Android Studio è un ambiente di sviluppo integrato (IDE) per lo sviluppo per la piattaforma Android;

Api : con Application Programming Interface (in acronimo API, in italiano interfaccia di programmazione di un'applicazione), in informatica, si indica ogni insieme di procedure disponibili per il programmatore, di solito raggruppate per formare un set di strumenti specifici per l'espletamento di un determinato compito all'interno di un certo programma. Spesso con tale termine si intendono le librerie software disponibili in un certo linguaggio di programmazione.

App : applicazione software dedicata ai dispositivi di tipo mobile, quali smartphone o tablet;

Bug : errore nella scrittura del codice sorgente che comporta comportamenti anomali del software;

Diagramma di Gantt : è uno strumento che permette di modellizzare la pianificazione dei compiti necessari alla realizzazione di un progetto; In un diagramma di Gantt ogni compito è rappresentato con una linea, mentre le colonne rappresentano i giorni, le settimane o i mesi del calendario secondo la durata del progetto. Il tempo stimato per un'azione è modellizzato su una barra orizzontale la cui estremità a sinistra è posizionata sulla data d'inizio prevista e l'estremità a destra sulla data prevista per la fine della realizzazione. I compiti possono susseguirsi in sequenza oppure essere eseguiti in parallelo;

IDE : In informatica un IDE è un software che, in fase di programmazione, aiuta i programmatori nello sviluppo del codice sorgente di un programma.

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;

Playstore : è un negozio online di applicazioni, brani musicali, pellicole cinematografiche, libri e riviste sviluppato da Google Inc. principalmente per

offrire servizi ai dispositivi mobili Android.

SQLite database : sistema per gestire una quantità di dati attraverso l'uso di un database accessibile solo dall'applicazione che lo genera e non dalle altre;

Testing : indica l'attività di verifica e collaudo del software;

Software : l'informazione o le informazioni utilizzate da uno o più sistemi informatici e memorizzate su uno o più supporti informatici. Tali informazioni possono essere quindi rappresentate da uno o più programmi, oppure da uno o più dati, oppure da una combinazione delle due;

UML (Unified modeling language) : "linguaggio di modellizzazione unificato". In ingegneria del software è un linguaggio di modellizzazione e specifica basato sul paradigma orientato agli oggetti.

XML (eXtensible Markup Language) : metalinguaggio per la definizione di linguaggi di markup, ovvero un linguaggio marcatore basato su un meccanismo sintattico che consente di definire e controllare il significato degli elementi contenuti in un documento o in un test.

Organizzazione del progetto

2.1 Modello del processo

Premesse:

- I componenti del team non hanno lavorato assieme precedentemente;
- Il team ha come unico luogo di ritrovo fisico l'università poichè nessuno è residente a Mestre o vive nella stessa città degli altri componenti del gruppo;
- Nessun componente del gruppo ha mai lavorato con AndroidStudio;
- Il team non ha mai sviluppato un'applicazione mobile;

Abbiamo scelto di seguire il modello di tipo evolutivo poichè siamo tutti inesperti e abbiamo la possibilità in questo modo di confrontarci man mano che la prototipizzazione avanza. In questo modo sarà possibile effettuare modifiche in corso d'opera fino a raggiungere lo scopo di questo progetto: l'applicazione mobile.

2.2 Struttura organizzativa

Per i motivi citati nel punto 2.1 abbiamo deciso, di comune accordo, di non creare una struttura gerarchica all'interno del team di lavoro e adottare quindi un tipo di team democratico decentralizzato. Le scelte vengono prese in comune, ascoltando le richieste e opinioni di tutti; in caso di non unanimità metteremo al voto le scelte e sarà la maggioranza, dopo un'approfondita analisi dei pro e dei contro, a prendere la decisione finale.

Non ci saranno ruoli specifici, in base alle difficoltà e necessità che si dovranno affrontare si assegneranno i diversi compiti.

2.3 Interfacce organizzative

Sicuramente nello sviluppo del progetto sarà necessario comunicare con persone esterne al nostro team per informazioni relative al codice e al feedback del nostro lavoro.

In queste relazioni saranno sicuramente coinvolti:

- Forum di supporto degli sviluppatori di Android e AndroidStudio;
- Prof. A. Cortesi e tutor A. Spanò per feedback sullo sviluppo dell'applicazione;
- Possibili utenti che esprimano le loro esigenze aggiuntive per la versione beta dell'app (prima di pubblicarla nello store);

2.4 Responsabilità di progetto

Principalmente seguiremo le direttive del Professor A. Cortesi in quanto a tempistiche per l'ultimazione dei vari task del progetto:

- piano di progetto (23/10/2017)
- consegna dei documenti di analisi e specifica dei requisiti (06/11/2017)
- piano di testing (15/11/2017)
- consegna documento di progettazione (10/12/2017)
- completamento della realizzazione dell'applicazione e messa in linea (31/01/2018).

Inoltre, avendo scelto la tipologia di team democratico decentralizzato divideremo le responsabilità e i compiti in maniera equa e consona alla disponibilità e competenza di ogni componente del gruppo di progetto.

Processi gestionali

3.1 Obiettivi e priorità

I nostri obiettivi, suddivisi per priorità, sono:

Priorità alta:

- progettare l'applicazione di ricerca degli impianti di distribuzione di carburanti;
- acquisire praticità con il software di sviluppo Android Studio;
- rendere l'applicazione funzionante e rispettare i requisiti preimpostati;
- rispettare i tempi di consegna che sono stati stabiliti;

Priorità media:

- acquisire capacità di lavoro in team riguardanti la suddivisione dei tempi e dei compiti;
- apprendere i metodi di sviluppo di un'applicazione in team.

3.2 Assunzioni, dipendenze, vincoli

Assunzioni:

- Nessuno dei membri abbandonerà il progetto;
- I componenti del gruppo hanno competenze necessarie e in fase di sviluppo per la realizzazione dell'intero progetto;
- Il team è in possesso di tutte le licenze software necessarie per lo sviluppo;

- L'utilizzo dell'app sarà progettata per utenti non portatori di handicap motori e visivi

Dipendenze:

- Altri corsi universitari da seguire;
- Altri esami da sostenere;
- Imparare a usare il software di sviluppo Android Studio
- Imparare le best practices del nuovo linguaggio di programmazione (Android).

Vincoli:

- Rispettare i tempi di consegna;
- Rispettare altre disposizioni date dal docente riguardo alle consegne.

3.3 Gestione dei rischi

Risk #5			Risk #2	
Risk #1	Risk #6	Risk #4		
	Risk #3			

1. Abbandono : come soluzione è stato deciso di comune accordo che non sarà possibile abbandonare il progetto in corso d'opera ed ognuno si è impegnerà a seguire tale vincolo;

2. Non rispettare le scadenze : come soluzione è stato deciso di ultimare ogni fase con almeno un giorno di anticipo per evitare che le scadenze effettivamente imposte non vengano rispettate;
3. Non imparare a programmare con Android nel tempo stabilito : questo rischio viene gestito nel momento in cui si determina il periodo di apprendimento ed ogni componente del gruppo si impegna ad imparare questo linguaggio di programmazione;
4. Perdere i file del progetto : attraverso il sistema di condivisione GitHub e altri sistemi di backup si cercherà di non perdere i file confidando nella buona operatività di questi sistemi;
5. Modifica dei requisiti durante lo sviluppo del progetto : si cerca di definire preventivamente i requisiti pensando ai possibili scenari.

3.4 Meccanismi di monitoraggio e di controllo

Sebbene la suddivisione dei compiti durante lo sviluppo del progetto, a task completato saremo tutti tenuti a controllare se ci sono errori o mancanze, anche nelle parti sviluppate dai nostri compagni di gruppo.

In questo modo ognuno avrà la possibilità di contribuire al progetto e alla realizzazione del task. Con la revisione finale, prima della consegna del task, avremo modo di comprendere a fondo anche le parti sviluppate dai nostri colleghi.

In questo modo le competenze acquisite da ogni persona saranno pressochè uguali. In caso di errori e mancanze di altro genere, le soluzioni, i cambiamenti o altre eventuali aggiunte verranno discusse in gruppo.

Useremo altre applicazioni e strumenti come **Telegram** (applicazione di messaggistica), per tenerci in contatto, e piattaforme di condivisione per gli scambi di file come **GitHub**(per condividere in tempo reale le modifiche apportate al progetto) e **GoogleDrive**.

3.5 Pianificazione dello staff

Sono necessarie conoscenze quali:

- conoscenza dell'IDE di sviluppo Android Studio;
- conoscenza di utilizzo di Git e GitHub.

I componenti del gruppo possiedono (sebbene in maniera non perfetta o poco approfondita) le conoscenze di cui sopra.

Verranno comunque integrate con conoscenze facilmente reperibili a lezione, in siti specializzati e video-lezioni o tutorial.

Processi tecnici

4.1 Metodi, strumenti e tecniche

Ogni elemento del gruppo svolge una parte del lavoro scelta autonomamente (ma con successiva approvazione del resto del gruppo) in modo da poter dare il meglio di sé sulla parte che ritiene essere adatta a lui/lei a patto che sia proporzionale (sia come difficoltà che come mole di lavoro) alle parti degli altri colleghi. La divisione dei lavori viene fatta per il solo scopo di velocizzare i tempi di completamento e lascia comunque spazio a collaborazioni in caso di problemi o necessità e la comunicazione resta un punto fondamentale del processo di sviluppo del progetto.

4.1.1 Sistemi di calcolo

I calcolatori usati per la progettazione dell'applicazione dovranno essere dotati di un ambiente di sviluppo Android (Android Studio) che sono disponibili per tutte le piattaforme.

Utilizzando Android Studio, possiamo servirci dell'emulatore dello smartphone Android, munito nell'IDE stesso, per testare il programma.

4.1.2 Software utilizzati

Il team di sviluppo dispone delle seguenti attrezzature:

- I sistemi operativi preinstallati sui calcolatori (fra i quali Windows 10 e Ubuntu 16.04);
- Android Studio;
- Microsoft Project Document.
- Latex Texmaker

- SQLite database
- SQLite Studio

4.2 Documentazione del software

Il documento consegnato è:

- Piano di Progetto (23/10/2017).

I documenti che verranno scritti in seguito sono:

- consegna del documento di analisi e specifica dei requisiti (06/11/2017);
- piano di testing (15/11/2017);
- consegna documento di progettazione (10/12/2017);
- completamento della realizzazione dell'applicazione e messa in linea (31/01/2018).

4.3 Funzionalità di progetto

4.3.1 Pianificazione della qualità

- seguire le linee guida preimpostate e i requisiti del progetto;
- seguire il metodo prescelto dal team;
- aggiornare i documenti se e quando necessario;
- rispettare i tempi imposti per la consegna;
- scrivere in modo indentato il codice del programma e testare successivamente il suo funzionamento.

4.3.2 Pianificazione della gestione delle configurazioni

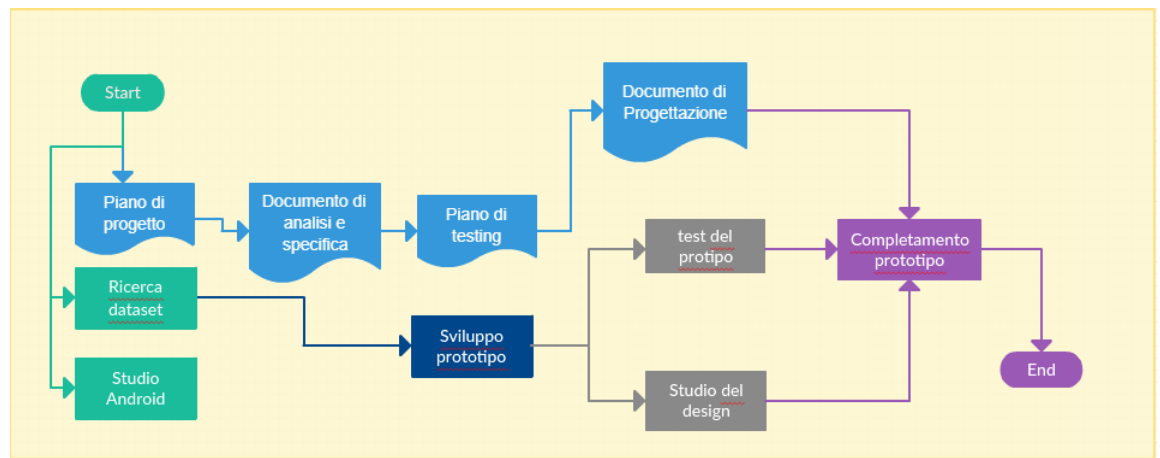
- Rispettare lo standard di programmazione, utilizzare nomi significativi e chiari, commentare adeguatamente il codice;
- Tenere traccia di ogni modifica del programma e documentare in modo dettagliato;

- Utilizzare la "*condivisione centralizzata online*" (Creare una repository GitHub) per facilitare l'aggiornamento del codice da parte di ogni membro del gruppo.

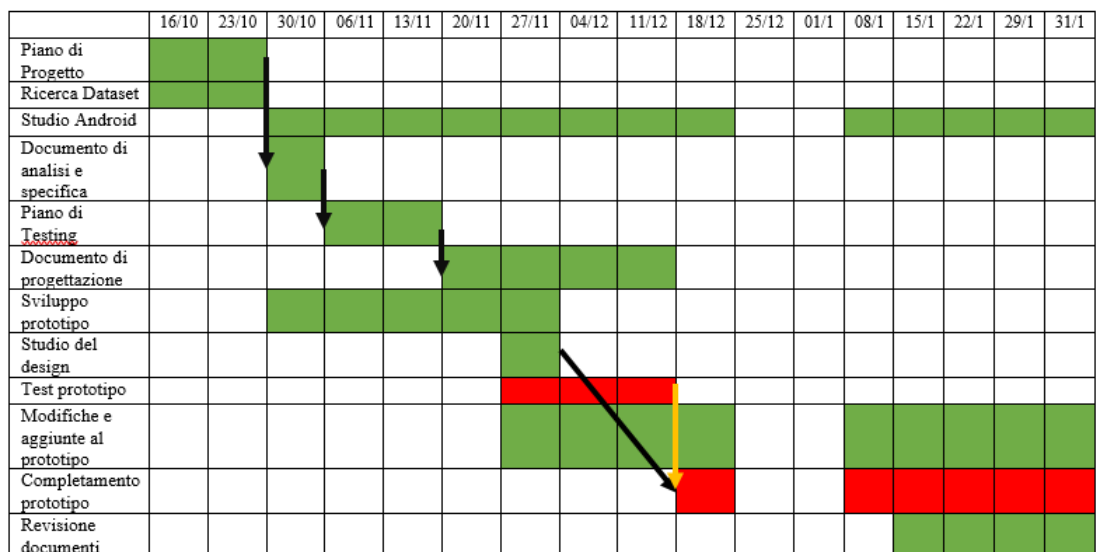
Pianificazione del lavoro, delle risorse umane e del budget

5.1 WBS (Work breakdown structure)

- Piano di progetto
- Ricerca dei dataset necessari
- Studio del linguaggio per programmare in Android
- Documento di analisi e specifica
- Piano di testing
- Documento di Progettazione
- Sviluppo di un semplice prototipo
- Studio del design dell'app
- Test del prototipo
- Completamento prototipo



5.2 Dipendenze



Attività



Attività critiche



Attività collegate



Attività critiche collegate

5.3 Risorse necessarie

- Gruppo di lavoro formato da 4 studenti di informatica
- IDE Android Studio
- IDE SQLite Studio
- Computer
- Smartphone con sistema operativo Android o simulatore

5.4 Allocazione del budget e delle risorse

Non è necessario allocare un budget per lo sviluppo della nostra applicazione in quanto i software che useremo (Texmaker, Android Studio, SQLite Studio, Telegram ed eventuali simulatori Android) sono gratuiti, come lo sono anche i materiali che useremo per imparare a programmare in Android (pdf forniti dal prof. Cortesi) e i dati(dataset) facilmente reperibili sul web in maniera gratuita.

Le risorse, verranno così suddivise:

- Ad ogni studente saranno affidati uno o più task di progetto
- Android Studio e Telegram saranno utilizzati da ogni studente
- Lo smartphone o il simulatore sarà utilizzato da ogni studente per l'importante fase di testing

Attività	Durata	Inizio	Fine
Piano di progetto	7 gg	lun 16/10/17	lun 23/10/17
Ricerca dataset	7 gg	lun 16/10/17	lun 23/10/17
<i>(*)Studio di Android</i>	<i>73 g</i>	<i>lun 30/10/17</i>	<i>mer 31/01/18</i>
Doc. analisi	5 gg	lun 30/10/17	ven 03/11/17
Piano di testing	11 gg	lun 6/11/17	ven 17/11/17
Doc. Progettazione	26 gg	lun 20/11/17	ven 15/12/17
Sviluppo prototipo	32 gg	lun 30/10/17	ven 01/12/17
Studio design	1 gg	lun 27/11/17	lun 27/11/17
Modifiche e aggiunte al prototipo	25 gg	lun 27/11/17	ven 22/12/17
Completamento prototipo	44 g	lun 18/12/17	mer 31/01/18
Revisione documenti	16 gg	lun 15/01/18	ven 31/01/18

NOTE:

(1) La dicitura studio di Android è stata inserita in quanto il progetto viene svolto all'interno del contesto universitario quindi sarà da tenere in considerazione che per progetti futuri questa parte non sarà da tenere in considerazione in quanto assolvibile come conoscenza pregressa. Una volta appreso il linguaggio, infatti, non sarà necessario tenere in conto le ore passate per lo studio del suddetto.

(2) In questa tabella i giorni vengono considerati con giorni festivi compresi.

5.4.1 STIMA COSTI CONSUNTIVI

Nella stima dei costi consuntivi prendiamo come riferimento i seguenti parametri:

- retribuzione media prevista 64 Euro / gg (8 Euro/ora)
- sono considerati solamente i **giorni lavorativi** (festività non comprese)

COSTO COMPLESSIVO PREVISTO :

16-10 / 31-10 = 14 giorni

01-11 / 30-11 = 26 giorni

01-12 / 31-12 = 20 giorni

01-01 / 31-01 = 26 giorni

Totale dei giorni : 86 giorni

Costo consuntivo a persona : 64 Euro/giorno * 86 giorni = 5504 Euro

Costo consuntivo totale : 5504 Euro * 4 Unità = 22016 Euro