



Università Ca' Foscari Venezia

Ingegneria Del Software

Piano di Progetto

Versione 1.1



Team HTN :

- Fabio Dainese 857661
- Giovanni Libralesso 858151
- Matteo Carraro 854143
- Nicolò Veronese 860568

Data : 4/1/2018

1 Introduzione	3
1.1 Overview del Progetto	3
1.2 Deliverables del progetto	3
1.3 Evoluzione del progetto	3
1.4 Materiali di riferimento	4
1.5 Definizioni e abbreviazioni	4
2 Organizzazione del progetto	6
2.1 Modello del processo	6
2.2 Struttura organizzativa	6
2.3 Interfacce organizzative	6
2.4 Responsabilità di progetto	7
3 Processi gestionali	8
3.1 Obiettivi e priorità	8
3.2 Assunzioni, Dipendenze, Vincoli	8
3.3 Gestione dei rischi	9
3.4 Meccanismi di monitoraggio e di controllo	10
4 Processi Tecnici	11
4.1 Metodi, Strumenti e tecniche	11
4.1.1 Sistemi di calcolo	11
4.1.2 Software utilizzati	11
4.2 Documentazione del software	11
4.3 Funzionalità di supporto al progetto	12
4.3.1 Pianificazione della qualità	12
4.3.2 Pianificazione della gestione delle configurazioni	12
5 Pianificazione del lavoro, delle risorse umane e del budget	13
5.1 WBS (Work Breakdown Structure)	13
5.1.1 Piano di progetto	13
5.1.2 Apprendimento delle competenze	13
5.1.3 Stesura dei documenti	13
5.1.4 Progettazione	13
5.1.5 Implementazione	14
5.1.6 Test	14
5.1.7 Commercializzazione	14
5.2 Dipendenze	14
5.3 Risorse necessarie	17
5.4 Allocazione del budget e delle risorse	17
5.5 Pianificazione	17

1 Introduzione

1.1 Overview del Progetto

La proposta di progetto del nostro gruppo prevede il monitoraggio di dati rilevati da una rete di stazioni fisse di monitoraggio che forniscono informazioni acquisite nelle ultime 48 ore riguardanti l'ozono e l'andamento degli ultimi 7 giorni per il PM10, le quali saranno selezionabili dall'utente attraverso una mappa dedicata con informazioni di tipo grafico/visivo.

I dati a cui facciamo riferimento sono stati presi dal sito di open data dell'Arpav (ARPA Veneto), ma quando avremo dati anche da altre regioni, sarà possibile unirli ai dataset esistenti, condividendo le stesse funzionalità. Nel caso venissero aggiunte altri tipi di fattori inquinanti sarà possibile aggiornare l'applicazione con sezioni aggiuntive.

1.2 Deliverables del progetto

Le scadenze previste per il progetto sono le seguenti :

- 24 Settembre 2017, definizione gruppi;
- 23 Ottobre 2017, consegna del piano di progetto;
- 09 Novembre 2017, consegna dei documenti di analisi e specifica dei requisiti;
- 23 Novembre 2017, piano di Testing (questo task verrà assolto grazie alla partecipazione da parte di tutti i membri del gruppo all'hackathon del 7 e 8 ottobre);
- 15 Dicembre 2017, consegna documento di progettazione;
- 31 Gennaio 2018, completamento della realizzazione dell'applicazione e messa in linea;

1.3 Evoluzione del progetto

Il progetto è, al tempo della stesura/revisione di questo documento, nelle fasi finali. Le funzionalità principali sono state ben definite e sono state soggette a cambiamenti e revisioni in base al feedback e considerazioni dei membri del gruppo.

1.4 Materiali di riferimento

- Slides del corso di Ingegneria del software:
<http://static.unive.it/isa/index/docente/persona/5591776/elementId/16023520>
- Manuale Android Studio e API Android:
<https://developer.android.com/guide/index.html>
- Sito Arpav Veneto:
<http://www.arpa.veneto.it/>
- Informazioni sull'elenco delle stazioni:
<http://89.96.234.233/aria-json/exported/aria/stats.json>
- Informazioni sui dati rilevati nelle ultime 48 ore per l'ozono e l'andamento degli ultimi 7 giorni per il PM10:
<http://89.96.234.233/aria-json/exported/aria/data.json>
- Coordinate geografiche delle stazioni:
<http://89.96.234.233/aria-json/exported/aria/coords.json>

1.5 Definizioni e abbreviazioni

- Android: È un sistema operativo per dispositivi mobili sviluppato da Google Inc. e basato sul kernel Linux. È 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);
- 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 al programmatore, di solito raggruppate a 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: È un'applicazione software dedicata ai dispositivi di tipo mobile, quali smartphone o tablet.
- 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 sinistra è posizionata sulla data d'inizio prevista e l'estremità destra sulla data prevista per la fine della realizzazione. I compiti possono susseguirsi in sequenza oppure essere eseguiti in parallelo;
- Diagramma di Pert: È una tecnica che permette di gestire la pianificazione in un progetto. Il metodo PERT consiste nel rappresentare in un graco una rete di azioni il cui svolgimento permette di arrivare al raggiungimento degli obiettivi di un progetto. Preventivamente, il metodo PERT implica: Una precisa scomposizione del progetto in azioni La stima della durata di ogni azione

- Java: un linguaggio di programmazione orientato agli oggetti a tipizzazione statica, specificatamente progettato per essere il più possibile indipendente dalla piattaforma di esecuzione;
- XML (eXtensible Markup Language): È un 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 testo;
- Software: è 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.
- 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;
- Bug: errore nella scrittura del codice sorgente che comporta comportamenti anomali del software;
- IDE: ambiente di sviluppo integrato che aiuta i programmatori nello sviluppo del codice sorgente dei programmi;
- Testing: indica l'attività di verifica e collaudo del software;
- Android Studio: Android Studio è un ambiente di sviluppo integrato (IDE) per lo sviluppo per la piattaforma Android.

2 Organizzazione del progetto

2.1 Modello del processo

Il team, dopo un'attenta riflessione, ha deciso di adottare per questo progetto il modello a cascata per i seguenti motivi:

- definire in modo chiaro e preciso il tempo di ciascuna attività e quindi anche il tempo totale necessario per il completamento del progetto;
- lavorare tutti insieme sulla stessa parte di progetto e nello stesso arco temporale;
- il processo di sviluppo viene diviso in fasi sequenziali;
- una fase produce un risultato che viene usato per quella successiva;
- l'implementazione viene rimandata fino a quando non sono perfettamente chiari gli obiettivi;
- semplificazione del controllo dell'andamento del progetto tramite la suddivisione del ciclo di vita in fasi successive ben definite.

Nel modello a cascata si riconoscono le seguenti fasi:

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

2.2 Struttura organizzativa

La struttura che il gruppo intende adottare è quella della cosiddetta Cooperazione (Democrazia) Decentralizzata che presenta le caratteristiche di:

- decisioni e consenso generale di gruppo per le soluzioni e per quanto riguarda l'organizzazione del lavoro;
- non è presente un leader vero e proprio in modo permanente;
- vi è comunicazione orizzontale, ovvero tutti parlano e ascoltano (tutti partecipano in modo attivo).

2.3 Interfacce organizzative

Il nostro gruppo cercherà di avere un riscontro con:

- il professore di Ingegneria del Software Agostino Cortesi, in modo da poter avere feedback sull'andamento del progetto e, in caso, di attuare delle rispettive modifiche;
- possibili altri gruppi che cercheranno di sviluppare un'applicazione simile alla nostra in modo da vedere se ci potrebbero essere funzioni da aggiungere o per consultarsi su alcune funzioni in comune di entrambi.

2.4 Responsabilità di progetto

Per quanto riguarda la responsabilità generale del progetto, poiché abbiamo stabilito che non vi è un vero e proprio leader permanente nel gruppo, ogni membro si occuperà sia della stesura dei documenti (o di una parte di essi) sia, successivamente, della parte riguardante la progettazione e il testing dell'applicazione, in modo da poter avere un carico di peso distribuito in modo equo per tutti i componenti del team.

3 Processi gestionali

3.1 Obiettivi e priorità

I nostri obiettivi sono:

- progettare l'applicazione CleAir (priorità alta);
- l'applicazione deve essere funzionante e rispettare i requisiti pre-impostati (priorità alta);
- imparare a utilizzare il software di sviluppo Android Studio (priorità alta);
- rispettare i tempi di consegna che sono stati stabiliti (priorità alta);
- imparare e migliorare le attività di lavoro in team (priorità media);
- apprendere i metodi di sviluppo di un'applicazione in team (priorità media);

3.2 Assunzioni, Dipendenze, Vincoli

Assunzioni:

- Nessuno dei membri abbandoni il progetto;
- I componenti del gruppo hanno competenze necessarie per la realizzazione dell'intero progetto;
- Siamo in possesso di tutte le licenze software necessarie per lo sviluppo;
- Gli end-user devono avere una connessione ad internet.

Dipendenze:

- Altri esami da sostenere;
- Velocità di imparare un nuovo linguaggio di programmazione (Android)

Vincoli:

- Rispettare i tempi di consegna;
- Altri corsi universitari da seguire.

3.3 Gestione dei rischi

			Risk #2	
	Risk #1			
Risk #6	Risk #5	Risk #4		
		Risk #3		
	Risk #8		Risk #7	

Immagine 0 - Tabella dei rischi

- Risk #1: Abbandono da parte di un membro del team;
- Risk #2: Sottostimare il tempo per completare il lavoro;
- Risk #3: Perdita di tempo dovuta a problemi di salute;
- Risk #4: Problemi nella codifica dell'applicazione-Generazione codice data un'idea;
- Risk #5: Perdita dei file e codice dell'applicazione;
- Risk #6: Modifica dei requisiti in corso d'opera;
- Risk #7: Mancanza di collaborazione tra i membri del team;
- Risk #8: Malfunzionamento dei sistemi informatici.

L'immagine 0 rappresenta la tabella dei rischi dove è possibile visualizzare il grado di probabilità e l'impatto che essi possono assumere nella realizzazione del progetto in questione. In modo particolare il colore verde rappresenta un incrocio tra una bassa probabilità che l'evento accada, e un basso impatto ai fini della realizzazione del progetto. Conseguentemente il colore giallo e rosso rappresentano un rischio maggiore.

3.4 Meccanismi di monitoraggio e di controllo

Ogniqualevolta si arrivi al completamento del task del progetto, ogni persona del gruppo provvederà a controllare se ci sono eventuali errori o se manca qualcosa, visto che collaboriamo tutti insieme per l'avanzamento del progetto, le competenze acquisite da ogni persona saranno pressoché uguali, così da permettere a ogni membro del gruppo di contribuire attivamente alla realizzazione del task, e così da ottenere una eventuale correzione degli errori in maniera più immediata. Inoltre si discuterà insieme la soluzione o eventuali cambiamenti.

Si useranno strumenti quali Google Drive (o, in caso di necessità, Messenger) per la condivisione dei file e WhatsApp e Discord per metterci in contatto qualora il gruppo non si riesca a trovare fisicamente. Per la condivisione del codice si utilizzerà la piattaforma GitHub.

4 Processi Tecnici

4.1 Metodi, Strumenti e tecniche

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 oppure Eclipse) che sono disponibili per tutte le piattaforme. Utilizzando IDE Android Studio, possiamo servirci dell'emulatore dello smartphone Android, munito nell'IDE stesso, per testare il programma. In alternativa, avremmo bisogno di uno smartphone con Android OS di versione minima Lollipop (5.0.1).

4.1.2 Software utilizzati

Il team di sviluppo dispone delle seguenti attrezzature:

- I sistemi operativi preinstallati sui calcolatori (Windows 10 , MacOS 10.12 , Ubuntu 17.04);
- Android Studio;
- Software per la realizzazione di diagrammi UML;
- Gantt Project per la realizzazione del diagramma di Gantt;
- Microsoft Visio, per la realizzazione del diagramma di Pert.

4.2 Documentazione del software

Il documento consegnato è :

- Piano di progetto 1.0, consegnato in data 23 Ottobre 2017

I documenti che verranno scritti in seguito sono :

- 09 Novembre 2017, consegna dei documenti di analisi e specifica dei requisiti;
- 22 Dicembre 2017, consegna documento di progettazione;
- 31 Gennaio 2018 completamento della realizzazione dell'applicazione e messa in linea .

4.3 Funzionalità di supporto al progetto

4.3.1 Pianificazione della qualità

- seguire le linee guida pre-impostate 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 i nomi significativi e commentare adeguatamente il codice;
- Tenere traccia di ogni modifica del programma e documentare in modo dettagliato;
- Utilizzare GitHub per facilitare l'aggiornamento del codice da parte di ogni membro del gruppo.

5 Pianificazione del lavoro, delle risorse umane e del budget

5.1 WBS (Work Breakdown Structure)

5.1.1 Piano di progetto

1. Stabilire gli obiettivi primari;
2. Analisi delle specifiche;
3. Analisi dei rischi;
 - a. Probabilità che accada;
 - b. Impatto sul progetto;
4. Definizione del modello;
5. Analisi delle risorse;
 - a. Definizioni delle risorse materiali;
 - b. Definizioni del tempo necessario per le attività.

5.1.2 Apprendimento delle competenze

1. Approfondimento Java;
2. XML e JSON;
3. Ambiente di sviluppo (Android Studio);
4. API (Android, Google).

5.1.3 Stesura dei documenti

1. Proposta di progetto;
2. Piano di progetto;
3. Documento di analisi dei requisiti;
4. Documento di progettazione.

5.1.4 Progettazione

1. Definire scheletro applicazione;
2. Definire sezioni principali;
3. Definire un'interfaccia utente di facile utilizzo e comprensione;
4. Progettazione lettura open data.

5.1.5 Implementazione

1. Sviluppo in parallelo delle varie sezioni (diverse tipologie di inquinanti dell'aria);
2. Collegamento open data;
3. Sviluppo interfaccia utente e integrazione con le funzioni.

5.1.6 Test

1. Controllo del corretto funzionamento dell'applicazione.

5.1.7 Commercializzazione

1. Registrazione account per il marketplace;
2. Pubblicazione applicazione sul marketplace.

5.2 Dipendenze

Qui di seguito è riportato il diagramma di Gantt (Immagine 1) del progetto. Successivamente è possibile vedere anche il diagramma di Pert (Immagine 2).

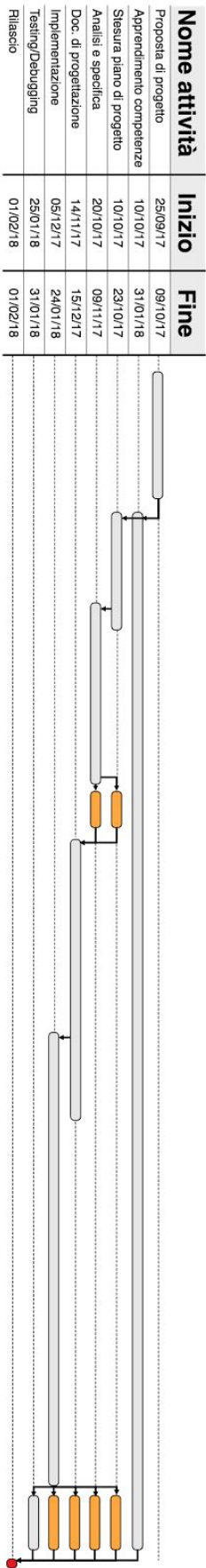


Immagine 1 - Diagramma di Gantt (le attività contrassegnate con l'arancione indicano la revisione dei relativi documenti)

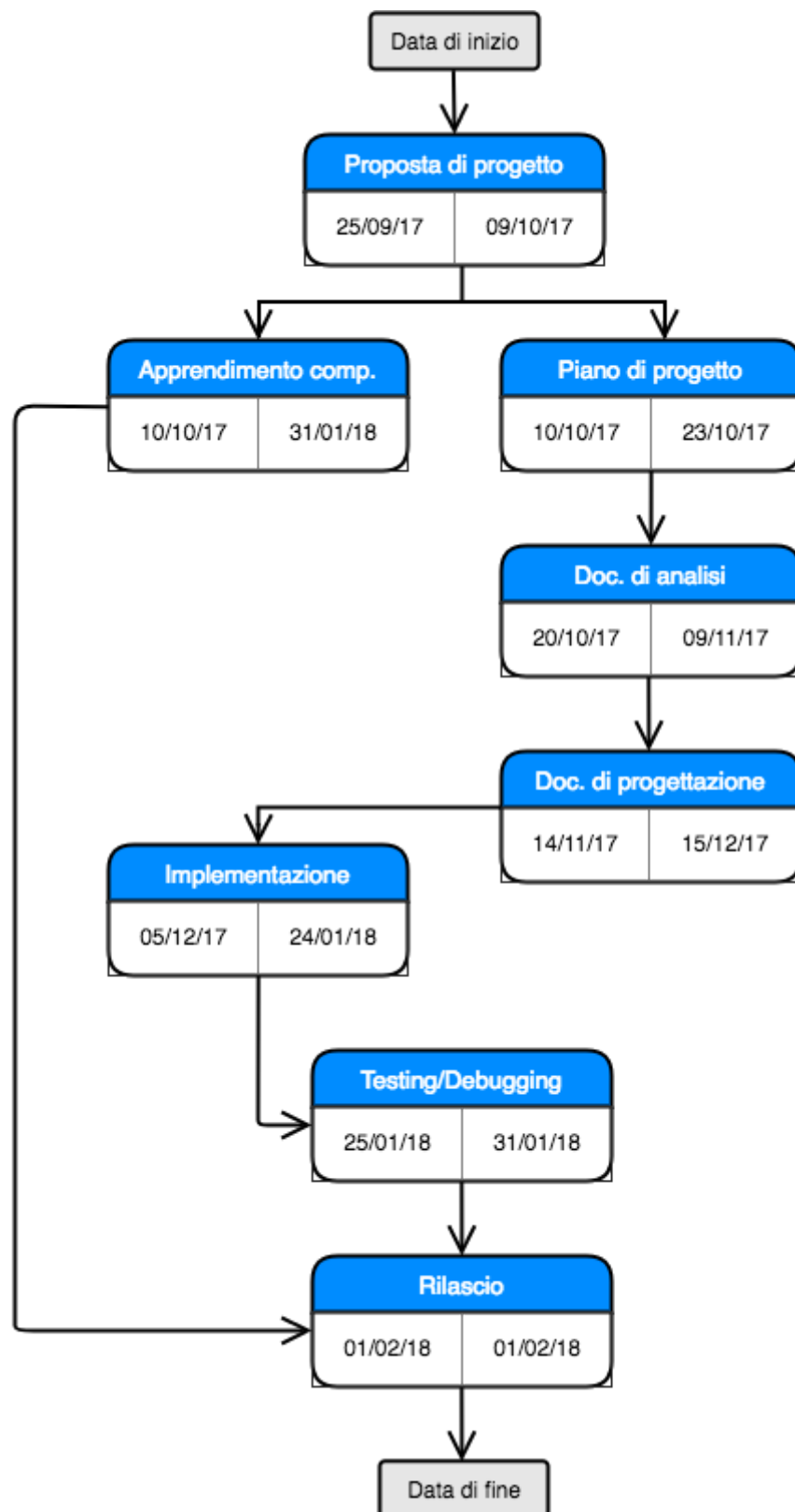


Immagine 2 - Diagramma di Pert

5.3 Risorse necessarie

Software:

- ambiente di sviluppo : Android Studio(Windows, Mac e Linux);
- ambiente per testing: emulatore di Android Studio e smartphone di prova;
- Editor immagini : Photoshop e Gimp;
- Editor di testo : Notepad++ (Windows) , Atom(Mac e Linux);
- Strumenti per lavoro di gruppo quali : Google Drive, GitHub, Gantt Project e Microsoft Visio;
- Sistemi operativi : Windows 10, MacOS 10.13 (High Sierra) e Ubuntu 17.04.

Hardware:

- Device per testare l'applicazione : Samsung S Advance (Android 5.0.1 Lollipop), Xiaomi mi 5 (Android 7.0 con MIUI 9 beta);
- MacBook Pro 15" Retina con emulatore di android studio.

5.4 Allocazione del budget e delle risorse

Per quanto riguarda lo sviluppo dell'applicazione non è stata prevista alcuna allocazione di budget, infatti il team ha già sia gli strumenti Hardware (come cellulari per testare il funzionamento dell'applicazione) e sia gli strumenti Software (ambienti di sviluppo) a completa disposizione.

Per lo sviluppo dell'applicazione e per la scrittura e/o modifica dei documenti abbiamo valutato come durata complessiva circa 90 gg/u.

5.5 Pianificazione

- 9 Novembre 2017, consegna dei documenti di analisi e specifica dei requisiti;
- 15 Dicembre 2017, consegna documento di progettazione;
- 31 Gennaio 2018, completamento della realizzazione dell'applicazione e messa in linea.