



# VilleInVeneto

Preparato per: Università Ca' Foscari, Venezia

Preparato da: Nikita Baldan (857172), Roberto Perissa (859143), Daniele Scocco (859542)

7 gennaio 2018

Numero versione: 2.0

---

# Indice

<b>Introduzione</b>	<b>3</b>
OVERVIEW DEL PROGETTO	3
STRUTTURA DEL PROGETTO	3
EVOLUZIONE DEL PROGETTO	3
DEFINIZIONI	3
<b>Organizzazione del progetto</b>	<b>5</b>
MODELLO DEL PROCESSO	5
STRUTTURA ORGANIZZATIVA E RESPONSABILITÀ DI PROGETTO	5
<b>Processi gestionali</b>	<b>5</b>
OBIETTIVI E PRIORITÀ	5
ASSUNZIONI, DIPENDENZE E VINCOLI	5
RAPPRESENTAZIONE E GESTIONE DEI RISCHI	6
RAPPRESENTAZIONE E GESTIONE DEI RISCHI	7
<b>Processi tecnici</b>	<b>8</b>
METODI STRUMENTI E TECNICHE	8
DOCUMENTAZIONE DEL SOFTWARE	8
FUNZIONALITÀ DI SUPPORTO AL PROGETTO	8
<b>Pianificazione del lavoro e risorse umane</b>	<b>8</b>
WBS	9
DIAGRAMMA PERT	10
DIAGRAMMA GANTT	11
RISORSE NECESSARIE	111
RIEPILOGO ATTIVITÀ E DURATA	122

---

# INTRODUZIONE

## Overview del progetto

VilleInVeneto è un app che permetterà alle Ville Venete di avere una maggiore visibilità, oltre al nostro territorio. Lo scopo di questa app è quello di definire il percorso migliore che tocchi le più belle Ville nel raggio di km definito dall'utente. L'app ottiene i dati, utili alla identificazione delle varie Ville presenti sul suolo della regione Veneto, tramite l'utilizzo di open data. Per la semplificazione dell'utilizzo delle API di Google, che utilizziamo per usufruire di Google Maps, utilizzeremo la libreria fornitaci dal Dott. Alvise Spanò.

## Struttura del progetto

Per quanto riguarda le scadenze del progetto, ci atterremo a quelle indicate dal professore Cortesi e sono strutturate in questo modo

- 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)

## Evoluzione del progetto

L'evoluzione del progetto avverrà principalmente in tre fasi:

- Utilizzo delle librerie forniteci da dott. Spanò per interfacciarsi a Google Maps in maniera più semplice ed efficace e specializzandoci nell'utilizzo e nella capacità di filtrare gli Open Data forniti dal sito <http://www.datiopen.it/it> per identificare le posizioni delle più importanti Ville Venete.
- Creazione dell'algoritmo che permetterà, dato un punto o una Villa Veneta e un chilometraggio definito dall'utente, l'ottenimento del miglior percorso che coinvolga le più belle Ville della zona.
- Creazione dell'interfaccia utente, che cercherà di rendere l'utilizzo "user friendly"

## Definizioni

Faremo riferimento al sito <http://www.datiopen.it/it> per l'accesso agli Open Data, a <https://it.wikipedia.org/> per reperire le informazioni sulle ville e per le definizioni. Adopereremo Google Maps per mappare i dati acquisiti. Durante la fase di programmazione utilizzeremo le librerie del dott. Spanò e consulteremo forum di programmazione come, ad esempio, <http://stackoverflow.com>.

## Android

Android è un sistema operativo per dispositivi mobili (mobile OS) sviluppato da Google Inc. basato su kernel Linux, erroneamente considerato a tutti gli effetti al pari di una distribuzione GNU/Linux per sistemi embedded (sistemi hardware che contengono anche il software di gestione). È stato 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), e altri. È per la quasi totalità Free and Open Source

---

Software (ad esclusione per esempio dei driver non-liberi inclusi per i produttori di dispositivi e di alcune Google apps incluse come il Google Play store), ed è distribuito sotto i termini della licenza libera.

*Fonte:* <https://it.wikipedia.org/wiki/Android>.

## **App**

In informatica con il neologismo app s'intende una variante delle applicazioni informatiche dedicate ai dispositivi di tipo mobile, quali smartphone e tablet. Il termine deriva proprio dall'abbreviazione di "applicazione".

*Fonte:* [https://it.wikipedia.org/wiki/App\\_%28smartphone%29](https://it.wikipedia.org/wiki/App_%28smartphone%29).

## **Testing**

È il vero e proprio collaudo del software.

*Fonte:* [https://it.wikipedia.org/wiki/Collaudo\\_del\\_software](https://it.wikipedia.org/wiki/Collaudo_del_software)

## **Open Data**

I dati aperti, comunemente chiamati con il termine inglese open data anche nel contesto italiano, 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.

*Fonte:* [https://it.wikipedia.org/wiki/Dati\\_aperti](https://it.wikipedia.org/wiki/Dati_aperti)

## **GPS**

Il sistema di posizionamento globale (GPS) è un sistema di posizionamento e navigazione satellitare civile che, attraverso una rete dedicata di satelliti artificiali in orbita, fornisce ad un terminale mobile o ricevitore GPS informazioni sulle sue coordinate geografiche ed orario, in ogni condizione meteorologica, ovunque sulla Terra o nelle sue immediate vicinanze ove vi sia un contatto privo di ostacoli con almeno quattro satelliti del sistema.

*Fonte:* [https://it.wikipedia.org/wiki/Sistema\\_di\\_posizionamento\\_globale](https://it.wikipedia.org/wiki/Sistema_di_posizionamento_globale)

## **IDE**

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

*Fonte:* [https://it.wikipedia.org/wiki/Integrated\\_development\\_environment](https://it.wikipedia.org/wiki/Integrated_development_environment)

## **Debugging**

Il debugging (o semplicemente debug), in informatica, indica l'attività che consiste nell'individuazione e correzione da parte del programmatore di uno o più errori (bug) rilevati nel software, direttamente in fase di programmazione oppure a seguito della fase di *testing* o dell'utilizzo finale del programma stesso.

*Fonte:* <https://it.wikipedia.org/wiki/Debugging>

---

# Organizzazione del progetto

## Modello del processo

Lo sviluppo di questo progetto seguirà il modello di processo a cascata. Inizialmente ci focalizzeremo su raccolta e filtraggio degli open data. Successivamente scriveremo l'algoritmo alla base dell'applicazione. Una volta scritto l'algoritmo, ci occuperemo dell'interfaccia che sfrutta Google maps per visualizzare i dati e offrire il servizio richiesto dall'utilizzatore.

## Struttura organizzativa e responsabilità di progetto

Il team è composto da tre membri. La ricerca dei dati e la grafica della documentazione è sotto la responsabilità di Nikita Baldan, i contenuti della documentazione sono a carico di Daniele Scocco e Roberto Perissa. La stesura del codice dell'algoritmo sarà frutto della collaborazione di tutti e tre i membri in modo da poter trovare la soluzione ottimale.

# PROCESSI GESTIONALI

## Obiettivi e priorità

L'obiettivo è quello di fornire all'utilizzatore un servizio ben funzionante, che implementi tutte le funzionalità proposte e descritte dal piano del progetto rispettando, inoltre, tutti i requisiti. La priorità è che venga consegnata l'applicazione entro la data stabilita.

## Assunzioni, dipendenze e vincoli

### Assunzioni:

Assumo che lo smartphone abbia a disposizione memoria sufficiente, connessione dati e dispositivo GPS; il sistema operativo Android abbia tutte le API necessarie.

Assumo che l'utente abbia capacità e conoscenze sufficienti per poter utilizzare l'applicazione maps in quanto comporta saper utilizzare una serie di strumenti implementati anche in questa applicazione.

### Dipendenze:

Lo sviluppo applicativo dell'algoritmo dipende dall'aver indentificato, uniformato e strutturato i dati relativi agli edifici storici. La progettazione dell'interfaccia

utente dipende dallo sviluppo dell'algoritmo computazionale.

### Vincoli:

Vincoli dell'applicazione:

- L'algoritmo deve interfacciarsi con Google e gli Open Data
- Il percorso individuato dall'applicazione dovrà rispettare le specifiche date dall'utente
- Consegna progetto e documentazione entro i limiti temporali imposti

## Rappresentazione dei rischi

	Rischi	Probabilità occorrenza	Impatto	Descrizione
1	Malattia	Medio	Medio / basso	Uno o più componenti del gruppo si ammalano
2	Guasto pc	Basso	Basso	Problema software o hardware
3	Perdita dati	Basso	Basso	Cancellazione di uno o più file
4	Sovrascrittura	Medio	Medio / basso	Cambiamento errato di uno o più file
5	Problemi struttura codice	Molto Alto	Alto	Progettazione del codice completamente o parzialmente errata
6	Bug	Molto Alto	Alto	Errori in fase di esecuzione del programma

maggiore						
↑						5, 6
Impatto						
↓			1, 4			
minore		2, 3				
	minore	←	Probabilità occorrenza	→	maggiore	

---

## Gestione dei rischi

	Rischi	Gestione rischio
1	Malattia	Prendo vitamine e mi vesto adeguatamente alla temperatura [PRE] Aspetto di guarire [POST]
2	Guasto pc	Evito siti potenzialmente dannosi e evito che il pc prenda colpi [PRE] Tentativo di ripararlo e ammortizzamento con l'utilizzo di altri pc [POST]
3	Perdita dati	Salvo continuamente le modifiche mantenendo backup frequenti [PRE] Sfrutto i backup per ripristinare il prima possibile [POST]
4	Sovrascrittura	Utilizzo di Git per gestire i cambiamenti [DURANTE]
5	Problemi struttura codice	Uso IDE, librerie standard e modelli ad alto livello consolidato [PRE] Porre maggiore attenzione in fase di progettazione [PRE]
6	Bug	Testare a breakpoint il codice [PRE] Utilizzo del debugger [POST]

---

# PROCESSI TECNICI

## Metodi strumenti e tecniche

Per la scrittura dell'applicazione utilizzeremo l'IDE di programmazione fornito da Google per la realizzazione di app Android: Android Studio. Per la condivisione del testo fra membri del team, per avere backup e cronologia delle modifiche utilizzeremo il servizio di web hosting Bitbucket. Per la fase di testing useremo un emulatore eseguito sul pc e degli smartphone.

## Documentazione del software

La documentazione riguardante il software verrà stesa durante lo sviluppo del progetto. I documenti che verranno redatti saranno i seguenti:

- Piano di progetto
- Documento di specifica dei requisiti
- Documento di progettazione

## Funzionalità di supporto al progetto

Per sviluppare questo progetto vi sarà bisogno dei seguenti programmi di supporto:

- Android Studio: IDE che permette di sviluppare per la piattaforma Android
- BitBucket: servizio di web hosting che permette di tenere una cronologia delle modifiche apportate al codice sorgente dell'applicazione
- Dropbox: spazio di archiviazione in cloud che permette di avere un backup della documentazione e una cronologia delle modifiche effettuate su di essa
- GitKraken: client grafico per repository git
- Debug USB smartphones Android: con il debug usb è molto più efficiente il test delle applicazioni rispetto ad un emulatore.

# PIANIFICAZIONE DEL LAVORO E RISORSE UMANE

Ho omesso la dicitura "budget" in quanto non verrà istanziato alcun budget.

Nella pagina successiva verrà presentata la WBS del nostro progetto.



---

## WBS

### 1. Presentazione del progetto

#### 1.1. Definizione dell'ambito del progetto

#### 1.2. Allocazione delle risorse e del calendario

#### 1.3. Analisi dei requisiti

##### 1.3.1. Definizione funzionalità / specifiche del prodotto

##### 1.3.2. Definizione strumenti utilizzati per lo sviluppo

#### 1.4. Realizzazione Documento di presentazione

### 2. Piano di progetto

#### 2.1. Analisi dei rischi

#### 2.2. Definizione delle fasi di sviluppo

#### 2.3. Definizione delle attività di sviluppo

#### 2.4. Costruzione del WBS, Gantt e Pert

#### 2.5. Realizzazione del documento del piano di progetto

### 3. Sviluppo Applicazione

#### 3.1. Logica applicativa

##### 3.1.1. Analisi

##### 3.1.2. Progettazione

##### 3.1.3. Sviluppo

##### 3.1.4. Test

#### 3.2. Sviluppo Interfaccia Utente

##### 3.2.1. Analisi

##### 3.2.2. Progettazione

##### 3.2.3. Sviluppo

##### 3.2.4. Test

### 4. Documentazione

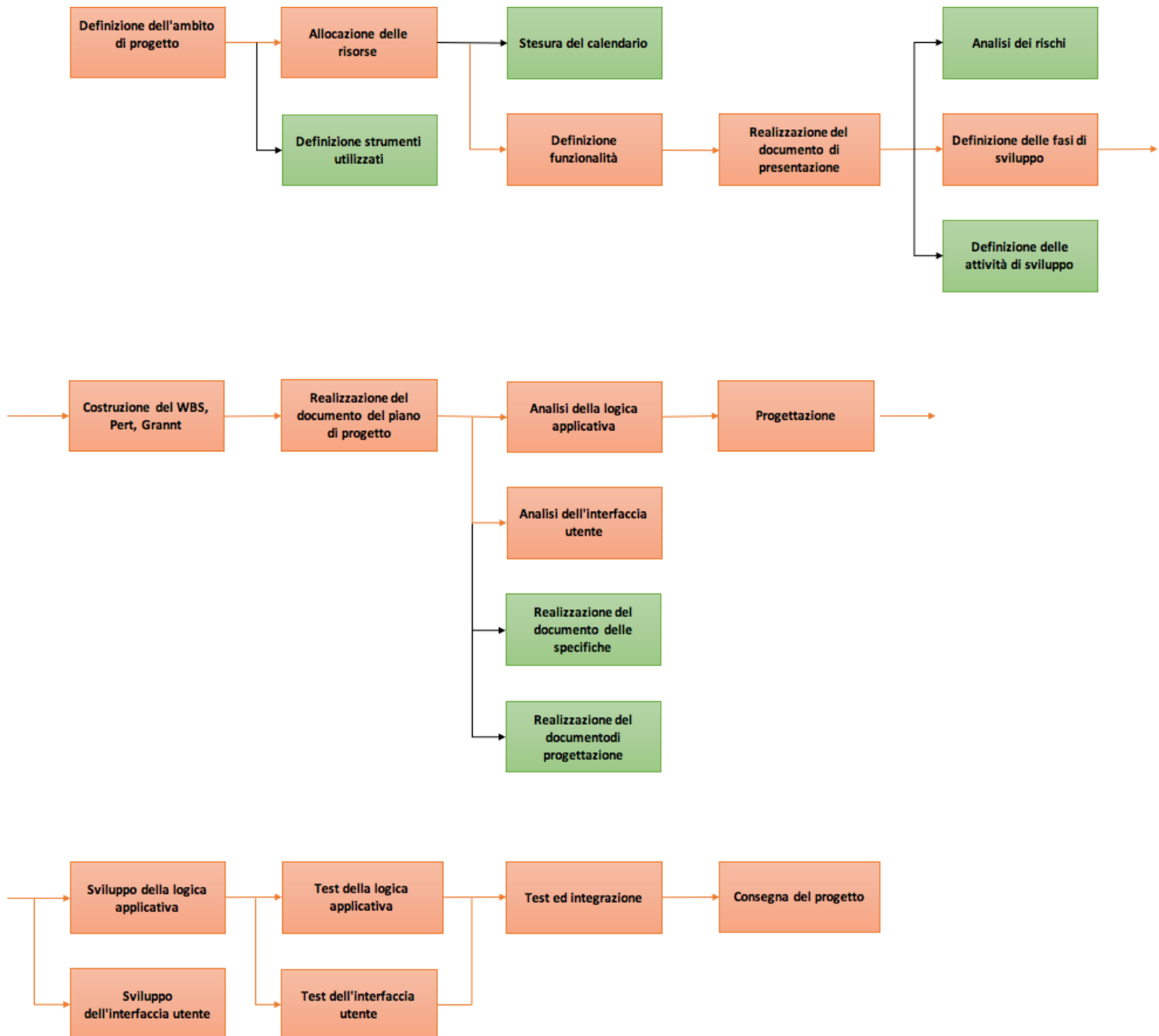
#### 4.1. Realizzazione del documento delle specifiche

#### 4.2. Realizzazione della documentazione relativa allo sviluppo

### 5. Rilascio

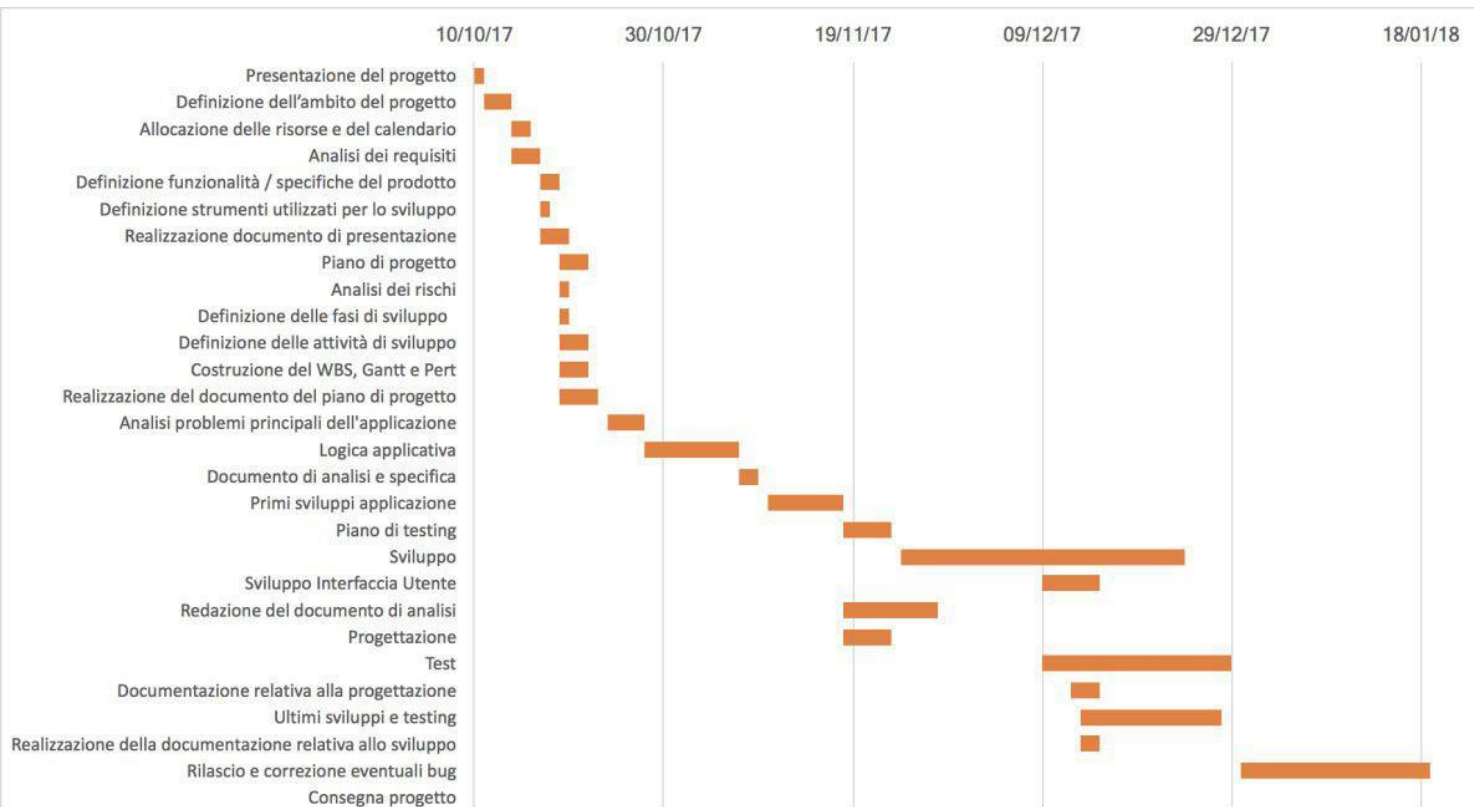
#### 5.1. Consegna progetto

## Diagramma PERT



Le attività critiche sono state identificate con l'arancione, quelle non critiche con il verde.

## Diagramma GANTT



## Risorse necessarie

Per lo sviluppo dell'applicazione abbiamo a disposizione 3 pc, dei quali: due con Windows 10 e uno con OSX.

Per testare l'applicazione utilizzeremo degli emulatori su pc che simulano varie versioni di Android e due smartphone Android con diverse caratteristiche hardware e software: un One Plus 5 (con Android 7.1.1) e un Samsung Galaxy Note 3 neo (con Android 5.1.1).

## Riepilogo attività e durata

Task	Data inizio	Giorni necessari	Data fine
Presentazione del progetto	10/10/17	1	11/10/17
Definizione dell'ambito del progetto	11/10/17	3	14/10/17
Allocazione delle risorse e del calendario	14/10/17	2	16/10/17
Analisi dei requisiti	14/10/17	3	17/10/17
Definizione funzionalità / specifiche del prodotto	17/10/17	2	19/10/17
Definizione strumenti utilizzati per lo sviluppo	17/10/17	1	18/10/17
Realizzazione documento di presentazione	17/10/17	3	20/10/17
Piano di progetto	19/10/17	3	22/10/17
Analisi dei rischi	19/10/17	1	20/10/17
Definizione delle fasi di sviluppo	19/10/17	1	20/10/17
Definizione delle attività di sviluppo	19/10/17	3	22/10/17
Costruzione del WBS, Gantt e Pert	19/10/17	3	22/10/17
Realizzazione del documento del piano di progetto	19/10/17	4	23/10/17
Analisi problemi principali dell'applicazione	24/10/17	4	28/10/17
Logica applicativa	28/10/17	10	07/11/17
Documento di analisi e specifica	07/11/17	2	09/11/17
Primi sviluppi applicazione	10/11/17	8	18/11/17
Piano di testing	18/11/17	5	23/11/17
Sviluppo	24/11/17	30	24/12/17
Sviluppo Interfaccia Utente	09/12/17	6	15/12/17
Redazione del documento di analisi	18/11/17	10	28/11/17
Progettazione	18/11/17	5	23/11/17
Test	09/12/17	20	29/12/17
Documentazione relativa alla progettazione	12/12/17	3	15/12/17
Ultimi sviluppi e testing	13/12/17	15	28/12/17
Realizzazione della documentazione relativa allo sviluppo	13/12/17	2	15/12/17
Rilascio e correzione eventuali bug	30/12/17	20	19/01/18
Consegna progetto	30/01/12	1	31/01/18
<b>Totale</b>		57 giorni ognuno	