

# SE2 Project Plan Document

Edoardo Giacomello      Mattia Fontana

January 25, 2016

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Revision History . . . . .	2
1.2	Purpose and Scope . . . . .	2
1.2.1	Purpose . . . . .	2
1.2.2	Scope . . . . .	2
1.3	List of Definitions and Abbreviations . . . . .	2
1.4	List of Reference Documents . . . . .	3
<b>2</b>	<b>Project Size and Cost Estimation</b>	<b>4</b>
2.1	Project Size: Function Points . . . . .	4
2.1.1	Complexity Weights . . . . .	4
2.1.2	ILF: Internal Logical Files . . . . .	5
2.1.3	EIFs (External Interface Files) . . . . .	5
2.1.4	EIs (External Inputs) . . . . .	5
2.1.5	EIQs (External Inquiries) . . . . .	6
2.1.6	EOs (External Outputs) . . . . .	6
2.1.7	UFPs (Un-adjusted Function-Points) . . . . .	6
2.2	Effort and Cost Estimation: COCOMO . . . . .	7
2.2.1	Source Lines of Code (SLOC) . . . . .	7
2.2.2	Scale Drivers . . . . .	8
2.2.3	Cost Drivers . . . . .	9
2.2.4	Effort Equation . . . . .	12
2.2.5	Schedule Estimation . . . . .	12
<b>3</b>	<b>Tasks and Schedule</b>	<b>14</b>
3.1	RASD : . . . . .	14
3.2	DD : . . . . .	14
3.3	Planning: . . . . .	14
3.4	Development . . . . .	14
3.5	Unit Testing . . . . .	15
3.6	Integration Testing: . . . . .	15
3.7	Inspection: . . . . .	15
<b>4</b>	<b>Resources Allocation and Schedule</b>	<b>16</b>
<b>5</b>	<b>Risks and Management</b>	<b>19</b>
<b>6</b>	<b>Work Hours</b>	<b>19</b>

# 1 Introduction

## 1.1 Revision History

## 1.2 Purpose and Scope

### 1.2.1 Purpose

The purpose of this document is to provide an estimation of the size, the costs and the possible risks in the development context for the MyTaxiServiceSystem.

### 1.2.2 Scope

The scope of this document is the planning and allocation of resources designated at the MyTaxiServicesoftware development. This document represent a starting point in cost and effort estimation.

## 1.3 List of Definitions and Abbreviations

**FP - Function Points** : Function points measure a software project by quantifying the information processing functionality associated with major external data or control input, output, or file types.

**RET - Record Element Type** : A RET is user recognizable sub group of data elements within an ILF or an EIF.

**DET - Data Element Type** A DET is a unique user recognizable, non-recursive (non-repetitive) field. A DET is information that is dynamic and not static. A dynamic field is read from a file or created from DETs contained in a FTR. Additionally, a DET can invoke transactions or can be additional information regarding transactions. If a DET is recursive then only the first occurrence of the DET is considered not every occurrence.

**EI - External Input** : Every unique user data or user control input type that enters the external boundary of the software system being measured

**EO - External Output** : Every unique user data or control output type that leaves the external boundary of the software system being measured.

**ILF - Internal Logical File** : Every major logical group of user data or control information that is generated, used or maintained by the software system

**EIF - External Interface File** : Files that are passed or shared between different software systems.

**EQ - External Inquiry** Every unique input-output combination, where input causes and generates an immediate output.

#### **1.4 List of Reference Documents**

- Assignment 1: Project Description
- MyTaxiServiceRequirement and Specification Analysis Document
- MyTaxiServiceDesign Document

## 2 Project Size and Cost Estimation

### 2.1 Project Size: Function Points

#### 2.1.1 Complexity Weights

In the evaluation of the complexity weights we referred to this table:

<b>Table 2. FP Counting Weights</b>			
<b>For Internal Logical Files and External Interface Files</b>			
<b>Data Elements</b>			
<b><u>Record Elements</u></b>	<b><u>1 - 19</u></b>	<b><u>20 - 50</u></b>	<b><u>51+</u></b>
1	Low	Low	Avg.
2 - 5	Low	Avg.	High
6+	Avg.	High	High
<b>For External Output and External Inquiry</b>			
<b>Data Elements</b>			
<b><u>File Types</u></b>	<b><u>1 - 5</u></b>	<b><u>6 - 19</u></b>	<b><u>20+</u></b>
0 or 1	Low	Low	Avg.
2 - 3	Low	Avg.	High
4+	Avg.	High	High
<b>For External Input</b>			
<b>Data Elements</b>			
<b><u>File Types</u></b>	<b><u>1 - 4</u></b>	<b><u>5 - 15</u></b>	<b><u>16+</u></b>
0 or 1	Low	Low	Avg.
2 - 3	Low	Avg.	High
3+	Avg.	High	High

### 2.1.2 ILF: Internal Logical Files

The data model resides in the database, which is typically counted as a single ILF which is structured as follow:

Name	RET	DET	Complexity
Zone	1	6	Low
Location	1	3	Low
Taxi	1	5	Low
Requests and Reservations	1	8	Low
Account	1	8	Low
Logs	1	$\leq 4$	Low

### 2.1.3 EIFs (External Interface Files)

The system does not rely on files that resides on external systems

### 2.1.4 EIs (External Inputs)

Name	RET	DET	Complexity
Login / Logout / Registration	1: Account	$\leq 8$	Low
TaxiProbe	1: Location	3	Low
TaxiReservation input and processing	2 Account, Reservation, Zones, Taxi	16	High
DriverResponse	1	3 (ReqId + TaxiId + Accepted)	Low
DriverNotification	1	3 (ReqId + TaxiId + Completed)	Low
LocationUpdate	1: Location	3	Low
BackupDatabase	n.a.		Avg
RestoreDatabase	n.a		Avg

### 2.1.5 EIQs (External Inquiries)

Name	RET	DET	Complexity
ShowProfile	1	8	Low
ShowAccounts	1	8	Low
ShowTaxiList	2	16	Avg
ShowLogs	n.a.		Avg

### 2.1.6 EOs (External Outputs)

Name	RET	DET	Complexity
TaxiProbeResponse	1	5(Taxi + Location)	Low
Confirmation	1Reservation	8	Low
Notification	1Reservation	8	Low
DriverRequest	1Reservation	8	Low

### 2.1.7 UFPs (Un-adjusted Function-Points)

Name	Low	Avg	High	Total
ILF	6*7	0	0	42
EIFs	0	0	0	0
EIs	5*3	2 *4	1*6	29
EIQs	2*3	2*4	0	14
EOs	4*4	0	0	16
UFP	79	16	6	<b>101</b>

## 2.2 Effort and Cost Estimation: COCOMO

All the tables used in this analysis have been taken from COCOMO II, Model Definition Manual at:

[http://csse.usc.edu/csse/research/COCOMOII/cocomo2000.0/CH\\_modelman2000.0.pdf](http://csse.usc.edu/csse/research/COCOMOII/cocomo2000.0/CH_modelman2000.0.pdf)

Elements of the COCOMO II model:

- Source Lines of Code (SLOC)
- Scale Drivers
- Cost Drivers
- The Effort Equation
- The Effort Adjustment Factor
- The Schedule Equation
- The SCED (Schedule Constraints) Cost Driver

### 2.2.1 Source Lines of Code (SLOC)

101 FPs \* 53 = 5353 SLOC

Where 53 is found from this table :

**Table 4. UFP to SLOC Conversion Ratios**

Language	Default SLOC / UFP	Language	Default SLOC / UFP
Access	38	Jovial	107
Ada 83	71	Lisp	64
Ada 95	49	Machine Code	640
AI Shell	49	Modula 2	80
APL	32	Pascal	91
Assembly - Basic	320	PERL	27
Assembly - Macro	213	PowerBuilder	16
Basic - ANSI	64	Prolog	64
Basic - Compiled	91	Query – Default	13
Basic - Visual	32	Report Generator	80
C	128	Second Generation Language	107
C++	55	Simulation – Default	46
Cobol (ANSI 85)	91	Spreadsheet	6
Database – Default	40	Third Generation Language	80
Fifth Generation Language	4	Unix Shell Scripts	107
First Generation Language	320	USR_1	1
Forth	64	USR_2	1
Fortran 77	107	USR_3	1
Fortran 95	71	USR_4	1
Fourth Generation Language	20	USR_5	1
High Level Language	64	Visual Basic 5.0	29
HTML 3.0	15	Visual C++	34
Java	53		



## 2.2.2 Scale Drivers

We use this table :

**Table 10. Scale Factor Values,  $SF_j$ , for COCOMO II Models**

Scale Factors	Very Low	Low	Nominal	High	Very High	Extra High
<b>PREC</b> $SF_j$ :	thoroughly unprecedented 6.20	largely unprecedented 4.96	somewhat unprecedented 3.72	generally familiar 2.48	largely familiar 1.24	thoroughly familiar 0.00
<b>FLEX</b> $SF_j$ :	rigorous 5.07	occasional relaxation 4.05	some relaxation 3.04	general conformity 2.03	some conformity 1.01	general goals 0.00
<b>RESL</b> $SF_j$ :	little (20%) 7.07	some (40%) 5.65	often (60%) 4.24	generally (75%) 2.83	mostly (90%) 1.41	full (100%) 0.00
<b>TEAM</b> $SF_j$ :	very difficult interactions 5.48	some difficult interactions 4.38	basically cooperative interactions 3.29	largely cooperative 2.19	highly cooperative 1.10	seamless interactions 0.00
<b>PMAT</b> $SF_j$ :	The estimated Equivalent Process Maturity Level (EPML) or					
	SW-CMM Level 1 Lower 7.80	SW-CMM Level 1 Upper 6.24	SW-CMM Level 2 4.68	SW-CMM Level 3 3.12	SW-CMM Level 4 1.56	SW-CMM Level 5 0.00

for defining the following parameters:

Scale Driver	Factor	Value
Precedentedness	Low	4.96
Development Flexibility	High	2.03
Architecture / Risk Resolution	Very High	1.41
Team Cohesion	Nominal	3.29
Process Maturity	High	3.12
Total :	/	14.81

Analysis of the results :

- Precedentedness : Low, the team was not expert with business scale projects.
- Development Flexibility: High, the project has been structured in a way that facilitates further changes.
- Architecture / Risk Resolution: Reflects the extent of risk analysis carried out, most of risk was deleted.

- Team Cohesion : Reflects how well the development team know each other and work together, this is Nominal because this is the first project that we do together.
- Process Maturity : This was evaluated around the 18 Key Process Area (KPA's) in the SEI Capability Model.

### 2.2.3 Cost Drivers

Cost Driver	Factor	Value
Required Software Reliability	Very Low	0.82
Data Base Size	Low	0.90
Product Complexity	High	1.17
Required Reusability	High	1.07
Documentation match to life-cycle needs	Nominal	1.00
Execution Time Constraint	High	1.11
Main Storage Constraint	Nominal	1.00
Platform Volatility	Very Low	n/a
Analyst Capability	High	0.85
Programmer Capability	High	0.88
Application Experience	Low	1.10
Personnel continuity	Very Low	1.29
Platform Experience	Low	1.09
Language and Tool Experience	Low	1.09
Usage of Software Tools	Nominal	1.00
Multisite development	Extra High	0.80
Required development schedule	High	1.00
Total :	/	16.17/17=0.95

For define this table we use the information contained in these tables :  
**Required Software Reliability** : Very Low, some slight inconvenience is allowed if recovered in a short time.

**Table 17. RELY Cost Driver**

RELY Descriptors:	slight inconvenience	low, easily recoverable losses	moderate, easily recoverable losses	high financial loss	risk to human life	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	0.82	0.92	1.00	1.10	1.26	n/a

**Data Base Size** : P=5353 SLOC D= 640 KB D/P=8.364

**Table 18. DATA Cost Driver**

DATA* Descriptors		Testing DB bytes/Pgm SLOC < 10	10 ≤ D/P < 100	100 ≤ D/P < 1000	D/P ≥ 1000	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	n/a	0.90	1.00	1.14	1.28	n/a

**Product Complexity:** high according to the new COCOMO II CPLEX rating scale.

**Table 20. CPLX Cost Driver**

<b>Rating Levels</b>	Very Low	Low	Nominal	High	Very High	Extra High
<b>Effort Multipliers</b>	0.73	0.87	1.00	1.17	1.34	1.74

**Required Reusability:**

**Table 21. RUSE Cost Driver**

<b>RUSE Descriptors:</b>		none	across project	across program	across product line	across multiple product lines
<b>Rating Levels</b>	Very Low	Low	Nominal	High	Very High	Extra High
<b>Effort Multipliers</b>	n/a	0.95	1.00	1.07	1.15	1.24

**Documentation match to life-cycle needs:**

**Table 22. DOCU Cost Driver**

<b>DOCU Descriptors:</b>	Many life-cycle needs uncovered	Some life-cycle needs uncovered.	Right-sized to life-cycle needs	Excessive for life-cycle needs	Very excessive for life-cycle needs	
<b>Rating Levels</b>	Very Low	Low	Nominal	High	Very High	Extra High
<b>Effort Multipliers</b>	0.81	0.91	1.00	1.11	1.23	n/a

**Execution Time Constraint:**

**Table 23. TIME Cost Driver**

<b>TIME Descriptors:</b>			≤ 50% use of available execution time	70% use of available execution time	85% use of available execution time	95% use of available execution time
<b>Rating Levels</b>	Very Low	Low	Nominal	High	Very High	Extra High
<b>Effort Multipliers</b>	n/a	n/a	1.00	1.11	1.29	1.63

**Main Storage Constraint:**

**Table 24. STOR Cost Driver**

<b>STOR Descriptors:</b>			≤ 50% use of available storage	70% use of available storage	85% use of available storage	95% use of available storage
<b>Rating Levels</b>	Very Low	Low	Nominal	High	Very High	Extra High
<b>Effort Multipliers</b>	n/a	n/a	1.00	1.05	1.17	1.46

**Platform Volatility:**

**Table 25. PVOL Cost Driver**

<b>PVOL Descriptors:</b>		Major change every 12 mo.; Minor change every 1 mo.	Major: 6 mo.; Minor: 2 wk.	Major: 2 mo.; Minor: 1 wk.	Major: 2 wk.; Minor: 2 days	
<b>Rating Levels</b>	Very Low	Low	Nominal	High	Very High	Extra High
<b>Effort Multipliers</b>	n/a	0.87	1.00	1.15	1.30	n/a

## Analyst Capability:

**Table 26. ACAP Cost Driver**

<b>ACAP Descriptors:</b>	15th percentile	35th percentile	55th percentile	75th percentile	90th percentile	
<b>Rating Levels</b>	Very Low	Low	Nominal	High	Very High	Extra High
<b>Effort Multipliers</b>	1.42	1.19	1.00	0.85	0.71	n/a

## Programmer Capability:

**Table 27. PCAP Cost Driver**

<b>PCAP Descriptors</b>	15th percentile	35th percentile	55th percentile	75th percentile	90th percentile	
<b>Rating Levels</b>	Very Low	Low	Nominal	High	Very High	Extra High
<b>Effort Multipliers</b>	1.34	1.15	1.00	0.88	0.76	n/a

## Application Experience:

**Table 29. APEX Cost Driver**

<b>APEX Descriptors:</b>	≤ 2 months	6 months	1 year	3 years	6 years	
<b>Rating Levels</b>	Very Low	Low	Nominal	High	Very High	Extra High
<b>Effort Multipliers</b>	1.22	1.10	1.00	0.88	0.81	n/a

## Platform Experience:

**Table 30. PLEX Cost Driver**

<b>PLEX Descriptors:</b>	≤ 2 months	6 months	1 year	3 years	6 year	
<b>Rating Levels</b>	Very Low	Low	Nominal	High	Very High	Extra High
<b>Effort Multipliers</b>	1.19	1.09	1.00	0.91	0.85	n/a

## Language and Tool Experience:

**Table 31. LTEX Cost Driver**

<b>LTEX Descriptors:</b>	≤ 2 months	6 months	1 year	3 years	6 year	
<b>Rating Levels</b>	Very Low	Low	Nominal	High	Very High	Extra High
<b>Effort Multipliers</b>	1.20	1.09	1.00	0.91	0.84	

## Personnel continuity:

**Table 28. PCON Cost Driver**

<b>PCON Descriptors:</b>	48% / year	24% / year	12% / year	6% / year	3% / year	
<b>Rating Levels</b>	Very Low	Low	Nominal	High	Very High	Extra High
<b>Effort Multipliers</b>	1.29	1.12	1.00	0.90	0.81	

## Usage of Software Tools:

**Table 32. TOOL Cost Driver**

<b>TOOL Descriptors</b>	edit, code, debug	simple, frontend, backend CASE, little integration	basic life-cycle tools, moderately integrated	strong, mature life-cycle tools, moderately integrated	strong, mature, proactive life-cycle tools, well integrated with processes, methods, reuse	
<b>Rating Levels</b>	Very Low	Low	Nominal	High	Very High	Extra High
<b>Effort Multipliers</b>	1.17	1.09	1.00	0.90	0.78	n/a

Multisite development:

**Table 33. SITE Cost Driver**

<b>SITE: Collocation Descriptors:</b>	Inter- national	Multi-city and Multi- company	Multi-city or Multi- company	Same city or metro. area	Same building or complex	Fully collocated
<b>SITE: Communications Descriptors:</b>	Some phone, mail	Individual phone, FAX	Narrow band email	Wideband electronic communication.	Wideband elect. comm., occasional video conf.	Interactive multimedia
<b>Rating Levels</b>	Very Low	Low	Nominal	High	Very High	Extra High
<b>Effort Multipliers</b>	1.22	1.09	1.00	0.93	0.86	0.80

Required development schedule:

**Table 34. SCED Cost Driver**

<b>SCED Descriptors</b>	75% of nominal	85% of nominal	100% of nominal	130% of nominal	160% of nominal	
<b>Rating Level</b>	Very Low	Low	Nominal	High	Very High	Extra High
<b>Effort Multiplier</b>	1.43	1.14	1.00	1.00	1.00	n/a

## 2.2.4 Effort Equation

$$Effort = A * EAF * (KSLOC)^E$$

Where:

- $A = 2.94$
- $EAF =$  product of all the cost drivers, equal to : 0.95 ;
- $E =$  exponent derived from Scale Drivers. Is calculated as:  
 $B + 0.01 * \sum\{i\} SF[i] := B + 0.01 * 14.81 = 0.91 + 0.1481 = 1.0581$ ;  
in which B is equal to: 0.91 for COCOMO.2000 .
- $KSLOC =$  estimated lines of code using the FP analysis: 5,353

$$Effort = 2.94 * 0.95 * 5,353^{1.0581} = 16.4816 Person - months$$

## 2.2.5 Schedule Estimation

$$Duration := 3.67 * Effort^F$$

Where:

- $F := 0.28 + 0.2 * (E - B) = 0.28 + 0.2 * (1.0581 - 0.91) = 0.30962$

The estimated project duration is:

$$Duration = 3.67 * 16.4816^{0.30962} = 8.739 = 9 Months$$

The estimation of the team size for this project is:

$$P = Effort / Duration = 16.4816 / 9 = 1.83 = 2 People$$

We want to give a more precise estimation adjusting some Scale Driver. To evaluate the COCOMO II and determine the effort required to complete the software project we also used an online tool (<http://csse.usc.edu/tools/COCOMOII.php>). This is the result of the online tool:



## COCOMO II - Constructive Cost Model

Software Size Sizing Method Source Lines of Code

[SLOC](#) % Design Modified % Code Modified % Integration Required Assessment and Assimilation (0% - 8%) Software Understanding (0% - 50%) Unfamiliarity (0-1)

New

Reused

Modified

**Software Scale Drivers**

Precedentedness Low Architecture / Risk Resolution Very High Process Maturity High

Development Flexibility High Team Cohesion Nominal

**Software Cost Drivers**

**Product**

Required Software Reliability Very Low

Data Base Size Low

Product Complexity High

Developed for Reusability High

Documentation Match to Lifecycle Needs Nominal

**Personnel**

Analyst Capability High

Programmer Capability High

Personnel Continuity Very Low

Application Experience Low

Platform Experience Low

Language and Toolset Experience Low

**Platform**

Time Constraint High

Storage Constraint Nominal

Platform Volatility Low

**Project**

Use of Software Tools Nominal

Multisite Development Extra High

Required Development Schedule High

Maintenance Off

**Software Labor Rates**

Cost per Person-Month (Dollars)

### Results

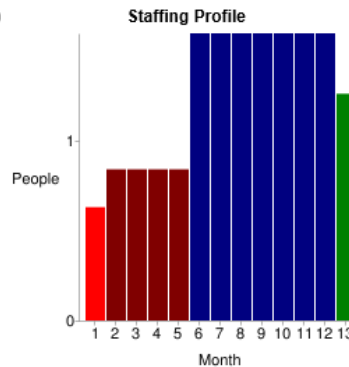
#### Software Development (Elaboration and Construction)

Effort = 15.4 Person-months  
Schedule = 11.8 Months  
Cost = \$30851

Total Equivalent Size = 5353 SLOC

#### Acquisition Phase Distribution

Phase	Effort (Person-months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	0.9	1.5	0.6	\$1851
Elaboration	3.7	4.4	0.8	\$7404
Construction	11.7	7.4	1.6	\$23447
Transition	1.9	1.5	1.3	\$3702



#### Software Effort Distribution for RUP/MBASE (Person-Months)

Phase/Activity	Inception	Elaboration	Construction	Transition
Management	0.1	0.4	1.2	0.3
Environment/CM	0.1	0.3	0.6	0.1
Requirements	0.4	0.7	0.9	0.1
Design	0.2	1.3	1.9	0.1
Implementation	0.1	0.5	4.0	0.4
Assessment	0.1	0.4	2.8	0.4
Deployment	0.0	0.1	0.4	0.6

Your output file is [http://csse.usc.edu/tools/data/COCOMO\\_January\\_23\\_2016\\_06\\_20\\_49\\_490696.txt](http://csse.usc.edu/tools/data/COCOMO_January_23_2016_06_20_49_490696.txt)

Created by Ray Madachy at the Naval Postgraduate School. For more information contact him at [rjmadach@nps.edu](mailto:rjmadach@nps.edu)

## **3 Tasks and Schedule**

### **3.1 RASD :**

- Identify the project objectives.
- Describe the structure and functions of the project.
- Define the specific requirements.
- Identify the use cases and scenarios.
- Representing UML diagrams.
- Representing the model in Alloy.
- Latex document.

### **3.2 DD :**

- Describe the architecture of the system.
- Describe component view and deployment view.
- Describe runtime view and component interfaces.
- Describe algorithm design.
- Define user interface design.
- Latex document.

### **3.3 Planning:**

- Define Function points.
- Define effort and cost estimation : Cocomo.
- Describe tasks for the project and the schedule.
- Define the risks.
- Latex document.

### **3.4 Development**

- Web Service
- Mobile Application
- Back-end
- Database
- Documentation

### **3.5 Unit Testing**

- Web Service
- Mobile Application
- Back-end
- Database
- Documentation

### **3.6 Integration Testing:**

- Describe integration strategy.
- Define individual steps and test description.
- Define program stubs.
- Latex document.

### **3.7 Inspection:**

- Define functional rules.
- Describe assignment checklist.
- Latex document.



## 4 Resources Allocation and Schedule

Here is presented the resource (team members) allocation with the relative deadline for each task. If a deadline is not present it means that that task is not assigned to the team member.

Task ID	Description	Giacomello	Fontana
1.1	Identify the project objectives.	15/10/2015	15/10/2015
1.2	Describe the structure and functions of the project.	15/10/2015	
1.3	Define the specific requirements.	15/10/2015	
1.4	Identify the use cases and scenarios.		15/10/2015
1.5	Representing UML diagrams.	30/10/2015	
1.6	Representing the model in Alloy.		30/10/2015
1.7	Documentation	30/10/2015	30/10/2015
DESIGN DOCUMENT			
2.1	Describe the architecture of the system	30/11/2015	30/11/2015
2.2	Describe component view and deployment view	30/11/2015	
2.3	Describe runtime view and component interfaces		30/12/2015
2.4	Describe algorithm design	15/01/2016	
2.5	Define user interface design		15/02/2016
2.6	Documentation	15/02/2016	15/02/2016
PLANNING			
3.1	Define Function points	30/02/2016	
3.2	Define Cocomo		30/02/2016
3.3	Describe task for the project and the schedule	30/02/2016	30/02/2016
3.4	Define the risks	30/02/2016	30/02/2016
3.5	Documentation	30/02/2016	30/02/2016

DEVELOPMENT			
4.1	Web Service		15/04/2016
4.2	Mobile Application		30/04/2016
4.3	Backend	30/04/2016	
4.4	Database	15/04/2016	
4.5	Documentation	30/04/2016	30/04/2016
UNIT TESTING			
5.1	Web Service	30/04/2016	
5.2	Mobile Application	30/04/2016	
5.3	Backend		30/04/2016
5.4	Database		30/04/2016
5.5	Documentation	30/04/2016	30/04/2016
INTEGRATION TESTING			
6.1	Component Stubs	30/06/2016	30/06/2016
6.2	Phase 1	30/06/2016	
6.3	Phase 2		15/07/2016
6.4	Phase 3	30/07/2016	
6.5	Phase 4		15/08/2016
6.6	Documentation	15/08/2016	15/08/2016
INSPECTION DOCUMENT			
7.1	Define functional rules	15/09/2016	
7.2	Describe assignment check list		15/09/2016
7.3	Documentation	15/09/2016	15/09/2016

	Oct	Oct	No	No	De	De	Jan	Jan	Fe	Fe	Ma	Ma	Apr	Apr	Ma	Ma	Ju	Ju	Jul	Jul	Au	Au	Se	Se	Oct	Oct
1.1																										
1.2																										
1.3																										
1.4																										
1.5																										
1.6																										
1.7																										
2.1																										
2.2																										
2.3																										
2.4																										
2.5																										
2.6																										
3.1																										
3.2																										
3.3																										
3.4																										
3.5																										
4.1																										
4.2																										
4.3																										
4.4																										
4.5																										
5.1																										
5.2																										
5.3																										
5.4																										
5.5																										
6.1																										
6.2																										
6.3																										
6.4																										
6.5																										
6.6																										
7.1																										
7.2																										
7.3																										
Deployment																										

## 5 Risks and Management

Risk	Severity	Possible Resolution
------	----------	---------------------

## 6 Work Hours

- Edoardo Giacomello:
- Mattia Fontana: