POLITECNICO
MILANO 1863

Politecnico di Milano
A.A. 2016/2017
Software Engineering 2: "*PowerEnJoy*" v. 1.0

Project **P**lan

Matteo Bresich (mat. 774366)

January 22, 2017

Table of Content

Page

1 Introduction 1

1.1 Purpose and Scope . . . 1

1.2 List of Definitions and Abbreviations . . . 1

1.3 Overview . . . 1

2 Function Point analysis 2

2.1 Introduction . . . 2

2.2 Estimation . . . 2

3 COCOMO II analysis 4

3.1 Introduction . . . 4

3.2 Estimation . . . 5

3.2.1 Scale Factors choice . . . 5

3.2.2 Cost Drivers choice . . . 5

3.2.3 Effort and Duration calculation . . . 6

3.3 Online Tool . . . 6

4 Tasks identification 8

5 Task scheduling 9

5.1 Start and Deadlines . . . 9

5.1.1 Main Tasks . . . 9

5.1.2 Implementation Milestone Details . . . 9

5.2 Gantt diagrams . . . 10

6 Resources allocation 10

7 Risks associated with the project 11

7.1 Risks . . . 11

7.2 Solutions . . . 11

8 Effort Spent 12

8.1 Hours of work . . . 12

9 References 12

1 Introduction

1.1 Purpose and Scope

The purpose of this document is to manage risks and plan the time, cost and resources for the project. A good prediction will ensure greater likelihood of project success.

1.2 List of Definitions and Abbreviations

- FP: Function Points
- COCOMO: COnstructive COst MOdel
- ILF: Internal Logical File
- EIF: External Interface File
- SLOC: Source Lines of Code
- KSLOC: Thousands of SLOC
- DAG: Directed Acyclic Graph

1.3 Overview

- Section 1: Introduction, provides a short description of the document.
- Section 2: provides Function Point analysis description and estimation.
- Section 3: provides Comomo II description and estimation.
- Section 4: provides project tasks identification and highlights of task's dependencies.
- Section 5: provides Gantt diagram representing the guideline for the scheduling of the project and show how the available resources are allocated to the project.
- Section 6: provides the resource allocation for the project.
- Section 7: provides risks associated with the project and how to avoid or manage them.
- Section 8: provides the time spent to redact the document.
- Section 9: used software and other dependencies.

2 Function Point analysis

2.1 Introduction

A function point is a unit of measurement to express the amount of business functionality that an information system provides to a user. The Function point analysis uses function points to estimate the effort to develop a software project. FP are categorized in:

- Internal Logical File: homogeneous data used and managed by the application
- External Interface File: homogeneous data used by the application but generated and maintained by other applications
- External Input: elementary operation to elaborate data coming from the external environment
- External Output: elementary operation that generates data for the external environment
- External Inquiry: elementary operation that involves input/output without data elaboration

2.2 Estimation

For the estimation it will be used weights in the following table (Table 1). The complexity of each functionality is assigned considering the number of fields and interactions with other components required to implement it.

Function Type	**Complexity-Weight**		
	Low	**Avarage**	**High**
Internal Logical File	7	10	15
External Interface File	5	7	10
External Input	3	4	6
External Output	4	5	7
External Inquiry	3	4	6

Table 1: FP weights categorized by level and types.

ILF Functionalities	**Complexity**	**FP Count**
User	Simple	7
Car	Complex	15
Reservation	Complex	15
Payment receipts	Simple	7
Total:		**44**

EIF Functionalities	**Complexity**	**FP Count**
Payment Data	Medium	7
Google Maps	Medium	7
Push Notification	Simple	5
MOM	Complex	10
Total:		**29**

External Input Functionalities	**Complexity**	**FP Count**
Login	Simple	3
Register	Simple	3
Create Reservation	Complex	6
Delete Reservation	Complex	6
Share Reservation	Medium	4
Payment	Medium	4
Update Car Status	Simple	3
Total:		**29**

External Output Functionalities	**Complexity**	**FP Count**
Email	Simple	4
Push Notification	Simple	4
SMS	Simple	4
Handyman Car Distribution Monitor	Medium	5
Total:		**17**

External Inquery Functionalities	**Complexity**	**FP Count**
Car Status	Simple	3
Active Reservation	Simple	3
User Profile	Simple	3
Total:		**9**

FP Amount:	128

The average conversion factor for J2EE language is 46[1]. To estimate the number of lines of code needed for the project will be used the following formula:

$$FPs * 46 = SLOC$$

$$122 * 46 = 5888\ SLOC$$

[1] For more info about the conversion factor see http://www.qsm.com/resources/function-point-languages-table

3 COCOMO II analysis

3.1 Introduction

The Constructive Cost Model (COCOMO) is a procedural software cost estimation model developed by Barry W. Boehm. The model parameters are derived from fitting a regression formula using data from historical projects. The result of this estimation is the number of Person-Months required to develop the project.
Two very important metrics are used for this evaluation:

- **Scale Factors**: the exponent used in the "effort equation"
  - **PREC**: Precedentedness
  - **FLEX**: Development flexibility
  - **RESL**: Risk resolution
  - **TEAM**: Team cohesion
  - **PMAT**: Process maturity
- **Cost Drivers**: the multiplicative factors that determine the effort required to complete the project
  - **RELY**: Required software reliability
  - **DATA**: Data base size
  - **CPLX**: Product Complexity
  - **RUSE**: Required Reusability
  - **DOCU**: Documentation match to life-cycle needs
  - **TIME**: Execution Time Constraint
  - **STOR**: Main Storage Constraint
  - **PVOL**: Platform Volatility
  - **ACAP**: Analyst Capability
  - **PCAP**: Programmer Capability
  - **PCON**: Personnel Continuity
  - **APEX**: Application Experience
  - **PLEX**: Platform Experience
  - **LTEX**: Language and Tool Experience
  - **TOOL**: Usage of Software Tools
  - **SITE**: Multisite Development
  - **SCED**: Required Development Schedule

Follow the formulas used for the estimation:

$$PM = A \cdot Size^{E} \prod_{i=1}^{17} EM_i \tag{1}$$

$$E = B + 0.01 \sum_{j=1}^{5} SF_j \tag{2}$$

$$TDEV = C \cdot (PM)^{F} \cdot (\frac{SCED\%}{100}) \tag{3}$$

$$F = D + 0.2 \cdot (E - B) \tag{4}$$

A = 2.94
B = 0.91
C = 3.67
D = 0.28

3.2 Estimation

3.2.1 Scale Factors choice

Code	Name	Factor	Value
PREC	Precedentedness	Nominal	3.72
FLEX	Development flexibility	Low	4.05
RESL	Risk resolution	High	2.83
TEAM	Team cohesion	High	2.19
PMAT	Process maturity	High	3.12
$E = 0.91 + 0.01 \times \sum_j SF_j$			1.0691

3.2.2 Cost Drivers choice

EAF: Effort Adjustment Factor

Code	Name	Factor	Value
RELY	Required Software Reliability	Nominal	1.00
DATA	Data base size	Nominal	1.00
CPLX	Product Complexity	Nominal	1.00
RUSE	Required Reusability	High	1.07
DOCU	Documentation match to life-cycle needs	Nominal	1.00
TIME	Execution Time Constraint	Nominal	1.00
STOR	Main Storage Constraint	Nominal	1.00
PVOL	Platform Volatility	Nominal	1.00
ACAP	Analyst Capability	High	1.10
PCAP	Programmer Capability	High	0.88
PCON	Personnel Continuity	Very High	0.81
APEX	Application Experience	Very Low	1.22
PLEX	Platform Experience	Low	1.09
LTEX	Language and Tool Experience	Low	1.09
TOOL	Usage of Software Tools	Nominal	1.00
SITE	Multisite Development	High	0.93
SCED	Required Development Schedule	High	1.00
Total	$EAF = \prod_i EM_i$		1.131

For more information see Cocomo II Model Definition Manual

3.2.3 Effort and Duration calculation

Using 1,2,3,4 formulas and A,B,C,D values:

$$PM = A \cdot Size^{E} \cdot EAF$$

$$PM = 2.94 \cdot 5.888^{1.0691} \cdot 1.131 = 22.130 \; [Person\text{-}Month]$$

$$TDEV = 3.67 \cdot (22.130)^{(0.28+0.2\cdot(1.0691-0.91))} \cdot 130\%/100 = 12.531 \; [Months]$$

The duration calculated previously results in a total of **9.639 Months**.
This would result in 2 developers needed for this project in fact

$$N_{people} = \lceil 22.130/12.531 \rceil = \lceil 1.77 \rceil = 2 \; People$$

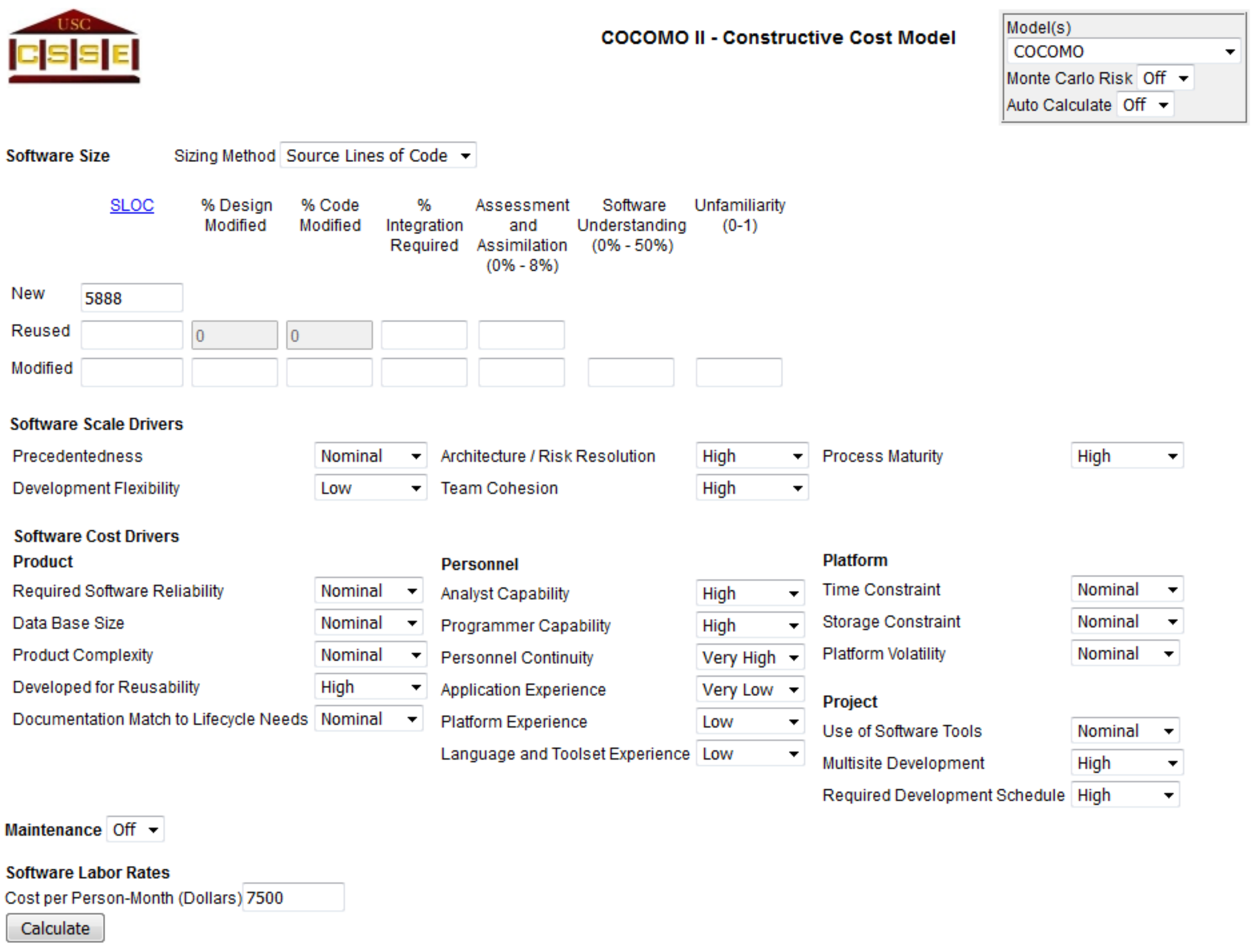
2 developers are required for the entire project. A reasonable development time would be:

$$22.130 \;/\; 2 = 11.065 \; [Month]$$

In order to be cautious with the task scheduling, we will approximate it to **12 Months**.

3.3 Online Tool

Follows a second analysis which has been done with the help of an online tool at this link:
http://csse.usc.edu/tools/COCOMOII.php.



COCOMO II - Constructive Cost Model

Model(s): COCOMO
Monte Carlo Risk: Off
Auto Calculate: Off

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	5888						
Reused		0	0				
Modified							

Software Scale Drivers

Precedentedness	Nominal	Architecture / Risk Resolution	High	Process Maturity	High
Development Flexibility	Low	Team Cohesion	High		

Software Cost Drivers

Product

Required Software Reliability	Nominal
Data Base Size	Nominal
Product Complexity	Nominal
Developed for Reusability	High
Documentation Match to Lifecycle Needs	Nominal

Personnel

Analyst Capability	High
Programmer Capability	High
Personnel Continuity	Very High
Application Experience	Very Low
Platform Experience	Low
Language and Toolset Experience	Low

Platform

Time Constraint	Nominal
Storage Constraint	Nominal
Platform Volatility	Nominal

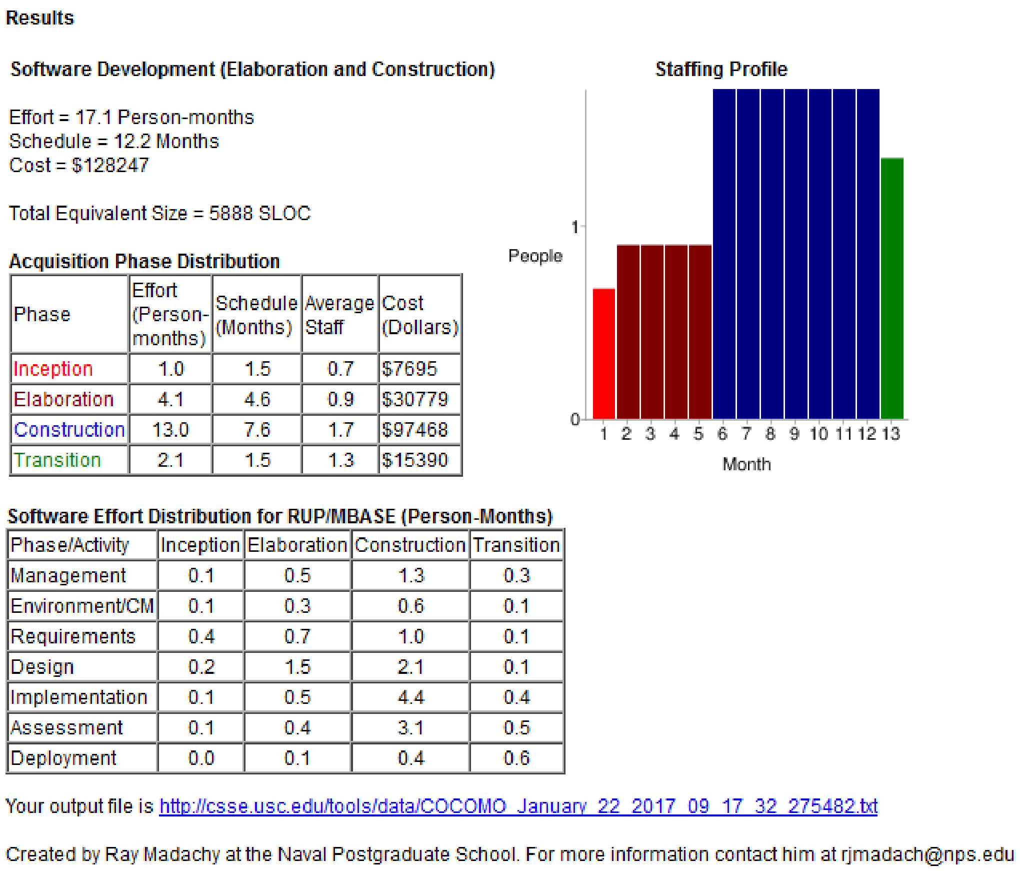
Project

Use of Software Tools	Nominal
Multisite Development	High
Required Development Schedule	High

Maintenance Off

Software Labor Rates

Cost per Person-Month (Dollars) 7500



Results

Software Development (Elaboration and Construction)

Effort = 17.1 Person-months
Schedule = 12.2 Months
Cost = $128247

Total Equivalent Size = 5888 SLOC

Acquisition Phase Distribution

Phase	Effort (Person-months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	1.0	1.5	0.7	$7695
Elaboration	4.1	4.6	0.9	$30779
Construction	13.0	7.6	1.7	$97468
Transition	2.1	1.5	1.3	$15390

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

Phase/Activity	Inception	Elaboration	Construction	Transition
Management	0.1	0.5	1.3	0.3
Environment/CM	0.1	0.3	0.6	0.1
Requirements	0.4	0.7	1.0	0.1
Design	0.2	1.5	2.1	0.1
Implementation	0.1	0.5	4.4	0.4
Assessment	0.1	0.4	3.1	0.5
Deployment	0.0	0.1	0.4	0.6

Your output file is http://csse.usc.edu/tools/data/COCOMO_January_22_2017_09_17_32_275482.txt

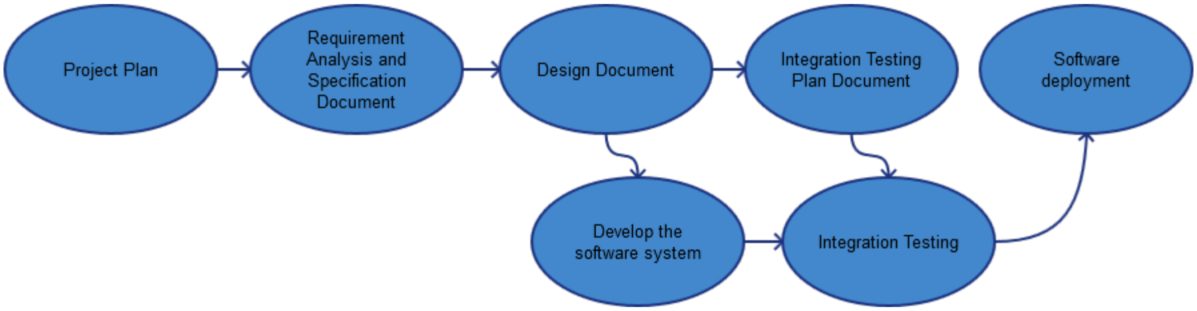
Created by Ray Madachy at the Naval Postgraduate School. For more information contact him at rjmadach@nps.edu

4 Tasks identification

The main tasks involving this project are shown in the table below.

Tasks ID	Task	**Effort** [Person-Days]	**Duration** [Days]	Dependencies
T1	Draft a Project Plan	20	10	-
T2	Draft a Requirement Analysis and Specification Document	40	20	T1
T3	Draft a Design Document	40	20	T2
T4	Develop the software system and write unit tests	258	128	T3
T5	Draft a Integration Testing Plan Document	22	12	T3
T6	Perform integration testing	32	16	T4, T5
T7	Software deployment	8	4	T6

The effort person-days can be calculated using a factor that represent the number of working day in a month that is 19. For *PowerEnJoy* the amount of person-days is 420.
Same Tasks shown before are now presented in a Directed Acyclic Graph that help to highlight dependencies among them.


Project Plan
Requirement Analysis and Specification Document
Design Document
Integration Testing Plan Document
Software deployment
Develop the software system
Integration Testing

Observation: The task T1, contrary to as shown in the table above and in the dag, was started and completed after RASD, DD and ITPD Documents contrary to what normally happens in a real project.

5 Task scheduling

5.1 Start and Deadlines

5.1.1 Main Tasks

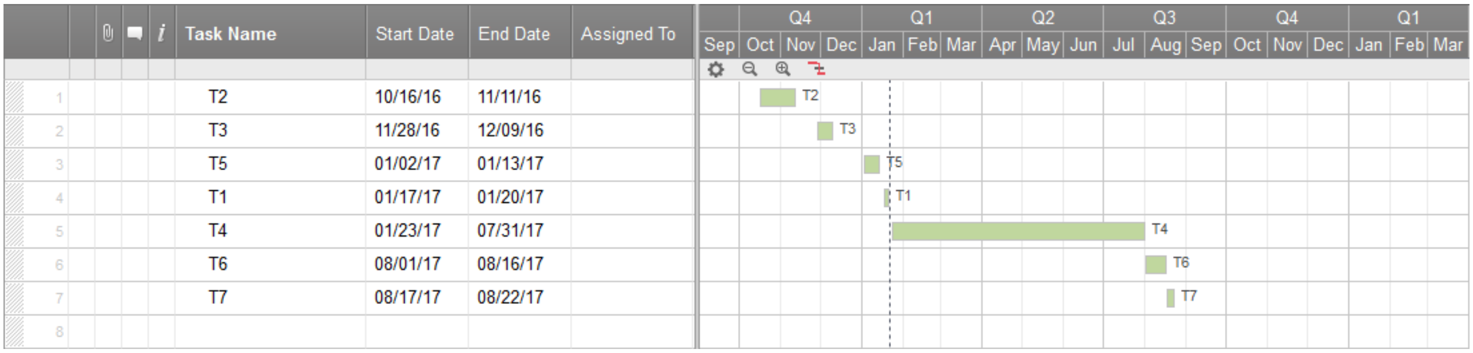
Task ID	Activity	Start Date	Deadline
T2	RASD	2016-10-16	2016-11-13
T3	DD	2016-11-28	2016-12-11
T5	ITPD	2017-01-02	2017-01-15
T1	PP	2017-01-17	2017-01-22
T4	Implementation	2017-01-23	2017-07-31
T6	Integration Testing	2017-08-01	2017-08-16
T7	Deployment	2017-08-17	2017-08-22

5.1.2 Implementation Milestone Details

Task ID	Activity	Start Date	Deadline
T4.1	User Manager development	2017-01-23	2017-02-10
T4.2	Reservation Manager development	2017-02-11	2017-03-20
T4.3	Car Manager development	2017-03-21	2017-04-13
T4.4	Notification development	2017-04-14	2017-05-09
T4.5	Mobile Interface development	2017-05-10	2017-06-05
T4.6	Payment Manager development	2017-06-06	2017-06-29
T4.7	Web Interface development	2017-06-30	2017-07-31

5.2 Gantt diagrams

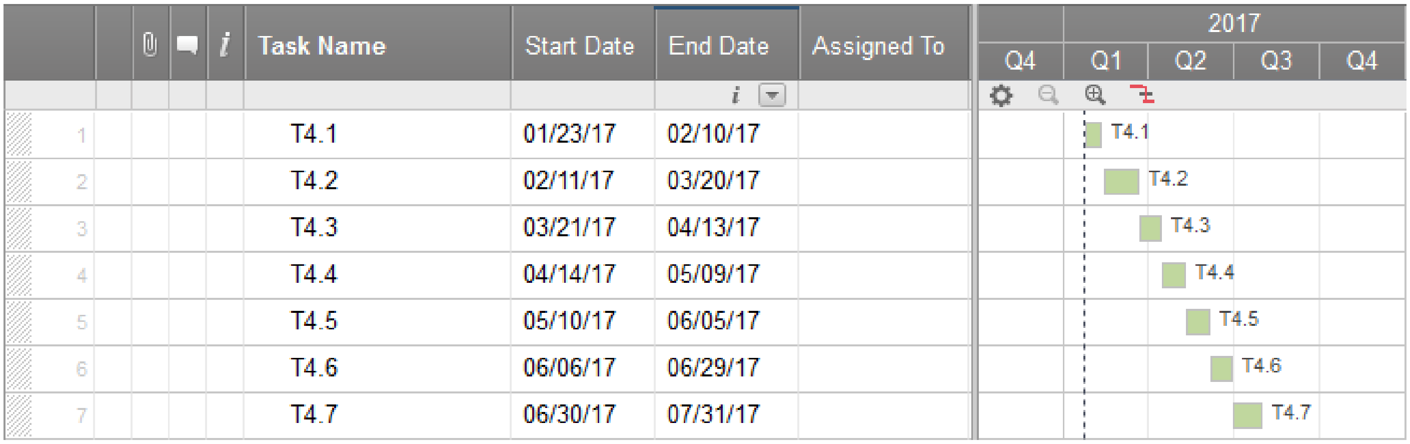
Below is shown the Gantt diagram representing the guideline for the scheduling of the project which helps respecting the deadlines and deliverables.



	Task Name	Start Date	End Date	Assigned To
1	T2	10/16/16	11/11/16	
2	T3	11/28/16	12/09/16	
3	T5	01/02/17	01/13/17	
4	T1	01/17/17	01/20/17	
5	T4	01/23/17	07/31/17	
6	T6	08/01/17	08/16/17	
7	T7	08/17/17	08/22/17	
8				

Q4
Q1
Q2
Q3
Q4
Q1
Sep Oct Nov Dec Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec Jan Feb Mar
T2
T3
T5
T1
T4
T6
T7

Below is shown in details the Gantt diagram of the Implementation Task.



	Task Name	Start Date	End Date	Assigned To
1	T4.1	01/23/17	02/10/17	
2	T4.2	02/11/17	03/20/17	
3	T4.3	03/21/17	04/13/17	
4	T4.4	04/14/17	05/09/17	
5	T4.5	05/10/17	06/05/17	
6	T4.6	06/06/17	06/29/17	
7	T4.7	06/30/17	07/31/17	

6 Resources allocation

As shown in Section 3, thanks to cocomo model, the project will be managed by a team of 2 people. The two developers will work in parallel on each task. This decision is motivated by the risk of lack of staff availability.

During the the software development part the team could be organized in such a way that permit one person to develop the software component while the other one could program the component's unit tests for a better result. In this way in fact the person that create the test does not know how the component is programmed.

7 Risks associated with the project

7.1 Risks

The main risk factors related to the project with a probability-of-occurrence are:

1. Delays and Unrealistic schedule
2. Wrong functionalities
3. Availability of staff
4. Bad external components
5. Integration testing failure

7.2 Solutions

1. **Delays and Unrealistic schedule**: the project could require more time than expected. To avoid the problem, the work plan is oversized compared to the estimation with more time for every deadline. If it is not enough it can be release a first, incomplete but working version of the project and build the less essential features later (e.g. starting from the mobile version and develop the web part later).

2. **Wrong functionalities**: to avoid misunderstanding about project's functionalities is important to organize, as soon as the project is assigned to the team, many interviews with stakeholders and managers of the project. These interviews should be organized not only during the design part, but also in the development phase too. In this way there is a constant check on the objectives and functionalities, and in case there is the possibility to timely intervention.

3. **Availability of staff**: the project could require more time than expected due to illness of key staff at critical times. To avoid the problem, the work plan is oversized compared to the estimation with more time for every deadline, and the highest priority tasks and complex ones are assigned to at least two persons.

4. **Bad external components**: the wrong choice about external components (for *PowerEnJoy* project are SMS, payment and email services) could cause dependencies with low quality services which could be unreliable or shut-down in the near future. To avoid this, first it is important to choose high quality ones offered by well known companies that have been in the business for years. Second it is important to choose easily replaceable external components.

5. **Integration testing failure**: after the implementation of some components, they may not pass the integration testing phase. To mitigate the problem of rewrite the software, it is necessary to define precisely the interfaces between components and subsystems, and by doing integration tests early using stubs and drivers.

8 Effort Spent

8.1 Hours of work

The time spent to redact this document:

- Bresich Matteo: 30 hours.

Days	Hours of work
17/01/17	4h
18/01/17	4h
19/01/17	4h
20/01/17	4h
21/01/17	6h
22/01/17	8h

9 References

- TeXstudio v2.11.2 (http://www.texstudio.org/) to produce this document.
- Evolus Pencil v2.0.5 (http://pencil.evolus.vn/) to generate diagrams.