PP

Project Plan

Version: 1.0.0

22th January 2017

Edited by:

Marzia Degiorgi - 878360
Giulia Leonardi - 877491
Valentina Ionata - 808678

Index

1. Introduction

1.1 Purpose and scope

The Project Plan document of the Power Enjoy project is a formal document used to guide both the project execution and the project control. It is aimed to estimate the resources needed to complete the project and schedule its development, to map people to their own tasks and to early assess the risks that may affect it. We carry out this analysis through two different methods: in the first part of the document, we employ the size estimation through the technique of Function Points; subsequently, in the second one, we evaluate costs and efforts by means of the constructive cost model COCOMO II.

1.2 Definitions, Acronyms, Abbreviations

1.2.1 Definitions

No specific definitions are used in the document.

1.2.2 Acronyms

- FP: Function Points
- ILF: Internal logic file
- ELF: External logic file
- EI: External Input
- EO: External Output
- EQ: External Inquiries
- DBMS: Database Management System
- API: Application Programming Interface
- GPS: Global Positioning System
- DMV: Department of Motor Vehicles

1.2.3 Abbreviations

No specific abbreviations are used in the document.

1.3 Reference documents

- *"MyTaxi Service, Project Plan Document"* Version 1.1, Casati, Castelli
- *"Power Enjoy, Requirements Analysis and Specification Document",* Degiorgi, Ionata, Leonardi
- *The Function Points complexity evaluation tables*
- *The COCOMO II Model Definition Manual*
- *"Project Management",*slide credits to: Hans van Vliet,Moreno Marzolla,Ian Sommerville, Damian Tamburri

2. Function Point Approach (FP)

The Function Point technique provides an objective, comparative measure that assists in the evaluation, planning, management and control of software production. The dimension of software is characterized based on the functionalities that it has to offer: to this purpose, each tasks of the project is mapped with a Function Type, and a weight is associated with each of them, according to the complexity. The evaluation about the effort required for each functionality has been estimated by means of the following table, which associate an amount of FPs to each Function Type, depending on the variable complexity.

Function type	**Complexity**		
	Simple	*Average*	*Complex*
Internal Logical File (ILF)	7	10	15
External Interface File (EIF)	5	7	10
External Input (EI)	3	4	6
External Output (EO)	4	5	7
External Inquiry (EQ)	3	4	6

We perform the evaluation step by step, pinpointing the functionalities belonging to each function type. Information about the project are gathered from RASD.

2.1 Internal Logic File

Power Enjoy system includes a number of ILFs that will be used to store essential informations.

First of all, the system needs to save data about the *users*. Name, surname, driving licence, credit card data, email and password have to be saved about them in a single table of the DBMS.

In a similar table, the *operators*' references have to be stored. These data are collected in a table composed by the following elements: name, surname, linked area and code assigned to the operator by the system when he/she is hired and, consequently, inserted into the DBMS. Moreover, both the tables with users' and operators' data have to contain a field in which the password that the system assigns to them can be stored. Address, date of birth and nationality are information less significant in term of the application, but useful for the identification: they are necessary in both the tables early mentioned.

The attributes regarding the entity "*Car*" are: identification number, battery level, availability and distance from the nearest safe areas. These last three entities, have to be constantly updated, depending on the information provided by the sensors embedded in each car.
Furthermore, the car position has to be stored as a pair <latitude, longitude> of coordinates, provided by the car GPS. Finally, the same table has to store the information about the charge: for a car which is parked on a safe area, the system needs to know if it is connected to a charger plug or not.

Concerning the *reservations*, the information have to be saved in a dedicated table, whose attributes are: the user who made the booking, the car rented, the number of passengers of the ride, the OTC (One Time Code) necessary to unlock the car, driver's and passengers' QR Codes, the time when the reservation has started (when the tuple is created), the time when the car is picked up, the one when it is released, and finally, the remaining battery level and the final cost of the ride, evaluated after the release of the car.

The system needs also to know the city division into *areas*, and where the safe areas are located. To this purpose, a table is set to store in each tuple the ID of the zone, the operator to which the zone was entrusted and its position. Since the territory is divided into squares, and the length of each square side is a known constant, the location is saved just through the coordinates of the centre of the area (still pinpointed as the <latitude, longitude> couple).

Another distinct table is necessary to store data about the *safe areas*. Each tuple contains the attribute ID and the coordinates of the car lots (usually three or four per safe area). This information are aimed to distinguish if a car is left within a safe lot or not, so as to evaluate if a discount is relevant.

By resuming this description, we can gather the following evaluation.

ILF	***Complexity***	***FP***
User data	Simple	7
Operator data	Simple	7
Car data	Simple	7
Reservation data	Average	10
Area data	Simple	7
Safe Area data	Simple	7
Total		***45***

2.2 External Interface File

The Power Enjoy system includes also the payment system and the verification of the driving licence validity. To this purpose, the application needs to consult data generated and maintained externally. Furthermore, it relies on Google Maps' for the research of the path and the indications to reach the rented car.

The EIF for the payment system consists of two main action:

- Given user's credentials and data about the payment method, verify that they are valid and get an affirmative or negative answer depending on them integrity;
- Given the data of the chosen payment method and the owed amount, carries out the charge and send a confirmation of the occurred payment back to the PowerEnjoy application.

The interface between the Power Enjoy application and the DMV databases is essential to prove the validity of the driving licence, given by the user while signing up, and that it properly belonged to him/her.
In this case, there is just one kind of interaction:

- Given the data provided by the user, get an affirmative or negative answer, depending on them integrity.

The interaction with Google Maps is necessary to allow the application to show the location of the available cars on the territory, the correct path to the car chosen, safe areas and interesting attractions during the way. It consists of these actions:

- Given an address, get the correspondent coordinates;
- Given the coordinates (or the addresses) of two location, evaluates the needed time to reach the second from the first and gives back road indications;

- Given the coordinates (or an address), pinpoints the safe areas and tourist attractions in proximity.

Finally, the estimate in term of values.

EIF	***Complexity***	***FP***
Payment Interface	Average	7
DMV Interface	Simple	5
Google Maps Interface	Average	7
Total		***19***

2.3 External Input

The Power Enjoy application allows to its users to carry out a wide range of operations, starting from the research of a car in a precise zone, to the booking or to the update of their personal information, in addition to provide the necessary actions for the administrator of the system and for the operators.

All users:

- *Login/Logout:* simple operations that involve just the session manager.

Users:

- *Create a new account:* since this operation needs a certain number of verifications and steps, an average complexity is linked to it;
- *Delete an account:* since it is a straightforward operation, it is assigned a low complexity to it;
- *Password retrieval:* this operation is quite complex, since it includes a huge amount of steps, in order to guarantee the safety of the personal information;
- *Change settings:* action with an high complexity, due to the high number of attributes that could be changed;
- *Search for a car:* this is a complex operation, which involves several controls and the interaction with another interface;

- *Make a reservation:* also this operation includes a large number of components and it is linked to an high complexity;
- *Delete a reservation:* since it involves just one entity, it is showed as a simple operation;
- *Find the path to a car:* operation with high complexity, that need to involve also external interfaces;
- *Unlock the car:* this is a simple operation, which just verify that the OTC inserted by the User matches with the one expected;
- *Authenticate him/herself in the car:* this is a simple operation too, since it involves just the check the correctness of the QR Code.

Operators:

- *Set a car as arranged:* simple operation which just involve one entity;
- *Find the path to a car:* operation with high complexity, that need to involve also external interfaces.

Administrator:

- *Add/Remove an operator:* highly complex operation which involves a great number of components;
- *Manage areas and safe areas:* as for the operators, a highly complex operation;
- *Request service statistics:* as this operation involves some fairly complicated aggregate queries on the database, it can be considered complex.

The resulting data are stored in the following table.

EI	***Complexity***	***FP***
Login/Logout	Simple	2x3
New Account	Average	4
Delete Account	Simple	3
Password retrieval	Complex	6
Change settings	Complex	6
Search for a car	Complex	6
Make a reservation	Complex	6
Delete a reservation	Simple	3
Find the path to a car	Complex	6
Unlock the car	Simple	3

Authenticate	Simple	3
Set a car as arranged	Simple	3
Add/Remove an operator	Complex/Simple	6+3
Add/Remove car	Complex/Simple	6+3
Manage areas and safe areas	Complex	2x6
Request service statistics	Complex	6
Total		***69***

2.4 External Output

As part of its normal behavior, Power Enjoy occasionally needs to communicate with the user outside the context of an inquiry. These occasions are:

- Notify the user who made the booking that it successfully concluded or not;
- Notify the user who ended a ride that the charge had a positive ending or not;
- Notify the operator that a car has been parked in his/her zone and if needs assistance or not.

All these operations can be though as fairly simple. The resulting table is the following:

EO	***Complexity***	***FP***
Booking notification	Simple	4
Charge notification	Simple	4
Assistance notification	Simple	4
Total		***12***

2.5 External Inquiries

An EQ is an operation which involves input and output, without significant elaboration of data from logic files.
Power Enjoy takes advantage of a few operations which don't require a complex computation, as following described:

- A user can retrieve the history of his/her reservations;
- An administrator can retrieve the list of the car owned by the company;
- An administrator can retrieve the list of the operators hired by the company;
- An administrator can retrieve the list of the stored areas and their coordinates;
- An administrator can retrieve the list of the user registered;
- An operator can retrieve the coordinates of the area assigned to him/her.

The evaluation about the EQ in through values in the following table.

EQ	***Complexity***	***FP***
Reservation history	Simple	3
List of cars	Simple	3
List of operators	Simple	3
List of areas	Average	4
List of users	Simple	3
Coordinates	Average	4
Total		***20***

2.6 Recap

The following table summarizes the results of our estimation activity:

Function Type	***Value***
Internal Logical File (ILF)	45
External Interface File (EIF)	19
External Input (EI)	69
External Output (EO)	12
External Inquiry (EQ)	20
Total	***165***

3. COCOMO Approach

3.1 Introduction

Effort estimation is the process of predicting the most realistic amount of effort, expressed in terms of person-hours, required to develop our software. To compute this number we need to define some concepts which will be used later in the actual calculation. The equation to compute the effort is the following:

$$\text{Effort} = A \cdot \text{SLOC}^{E} \cdot \prod_{i=1}^{n} EM_i$$

Where A = 2.94 for COCOMO II.2000, SLOC is the amount of lines of code estimated to be written for implementing the software (evaluated in KSLOC), E is an aggregation of dive scalar factors and EM is called effort multiplier which is defined for each Cost Driver. Finally, in COCOMO II.2000 the number of Cost Driver are 17, thus n = 17.

3.2 Lines of code

In order to estimate the number of lines of code needed, we use an average conversion of factor 46, as described in the "Function Point Languages Table" at http://www.qsm.com/resources/function-point-languages-table.

165 FPs x 46 = 7590 SLOC

This first estimation based on FPs approach converts FPs to SLOC.

3.3 Scale drivers

For the following evaluations, we will refer to this table, taken from the "COCOMO II Model Definition Manual":

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

Precisely, we will consider:

1. *Precedentedness:* **nominal**, since this is one of first times for us in developing a large scale projects;
2. *Development flexibility*: **low**, since the requirements are well-defined and quite strict;
3. *Risk resolution:* **high**, thanks to the risk analysis achieved;
4. *Team cohesion:* **very high**, since all the members come from the same training and they already worked successfully together in previous projects;
5. *Process maturity*: **nominal**.

The outcoming values are:

Scale driver	***Factor***	***Value***
PREC	Low	3.72
FLEX	Low	4.05
RESL	High	2.83

TEAM	Very High	1.10
PMAT	Nominal	4.68
Total		***16,38***

3.4 Cost drivers

- *Required software reliability:* the correct operation of the system is the only way to make use of the car sharing. For this reason, the value of this cost driver is set to **high**.

RELY Descriptors:	slight inconven-ience	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

- *Database size*: the estimated size of our database, evaluated taking into account the number of data to store early described, is 4GB. Since the expected lines of code are between 5000 and 10000, the D/P ratio is contained between 3.24 and and 6.48. Therefore, the Database size cost driver is set to **low**.

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*: set to **very high** according to the COCOMO II rating scale.

Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	0.73	0.87	1.00	1.17	1.34	1.74

- *Required reusability*: in our project, a great effort is consumed with creating more generic design of software, more elaborate documentation, and more extensive testing to ensure components are ready for use in other application. For this reason, the cost driver is set to **very high**.

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

- *Documentation match to life-cycle needs*: the suitability of our project's documentation to its life-cycle needs has been evaluated as **high**.

DOCU Descriptors:	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	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	0.81	0.91	1.00	1.11	1.23	n/a

- *Execution time constraint*: the percentage of available execution time expected to be used by our system consuming the execution time resource is under 50%: therefore, the value of the driver cost is set to **nominal.**

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

- *Storage constraint*: the usage of a dedicated system doesn't impose to us any particular constraint, so we can set this effort multiplier to **very low**.

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

- *Platform volatility*: the platform is not supposed to be subjected to great changes over time, thus **low** is the appropriate value for this cost driver.

PVOL Descriptors:		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	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	n/a	0.87	1.00	1.15	1.30	n/a

- *Analyst capability*: the value attributed to this cost driver is **high**, due to the accurate analysis made and the effort spent in writing the RASD and DD documents.

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

- *Programmer capability*: evaluation should be based on the capability of the programmers as a team rather than as individuals. Major factors, which should be considered in the rating, are efficiency, thoroughness, the ability to communicate and to cooperate. In our situation we chose the value of **high**.

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

- *Application experience*: since this is one of our first experiences in developing such a complex system , the cost driver is set to **low**.

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

- *Platform experience*: because of our competences about database, user interfaces and server side development, this cost driver is estimated as **nominal**.

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

- *Language and tool experience*: since this parameter reflects the competences taken into account in the previous cost driver, it is set to **nominal** too.

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

- *Personnel continuity*: the rating scale is in terms of project annual personnel turnovers. In our case it is set to **very low**.

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

- *Usage of software tools*: since no actual implementation is done, we can assume the use of a Java IDE such as Eclipse and a VCS of choice such as Git which can be easily integrated. Therefore, this value is set to **nominal**.

TOOL Descriptors	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	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	1.17	1.09	1.00	0.90	0.78	n/a

- *Multisite development*: since the developers were sporadically located in different cities, the value of this cost driver is set to **nominal**.

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

- *Required development schedule*: our efforts were well distributed over the available development time, but regardless of this fact, the implementation required high efforts at the later phases. Mainly this is due to the fact that we expanded the initial problem description in the more complex and profitable way for a real world application. For these reason this parameter is set to **high**.

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

3.5 Cost drivers recap

Our chosen values are recapped in the following table:

Cost driver	***Factor***	***Value***
RELY	High	1.10
DATA	Low	0.90
CPLX	Very High	1.34

RUSE	Very High	1.15
DOCU	High	1.11
TIME	Nominal	1.00
STOR	Very Low	n/a
PVOL	Low	0.87
ACAP	High	0.85
PCAP	High	0.88
APEX	Low	1.10
PLEX	Nominal	1.00
LTEX	Nominal	1.00
PCON	Very Low	1.29
TOOL	Nominal	1.00
SITE	Nominal	1.00
SCED	High	1.00
Product		***1.56373825***

3.6 Effort computation

Using the values computed in the previous sections we can now calculate the effort. Given that

- ➔ A=2.94 for COCOMO II
- ➔ SLOC = **7.590** from the **Lines of Code**
- ➔ EM = **1.56** from the **Cost Drivers**
- ➔ E= 0.91 + 0.01 x **16.38** = 1.0738 from the **Scale Drivers**

the effort is

$$\text{Effort} = 2.94 \times 7.590^{1.0738} \times 1.56 = 40.42 \text{ PM}$$

We can also calculate the duration of a project in terms of months required to complete it with this equation:

$$\text{Duration} = 3.67 \times (\text{Effort})^{SE} = 3.67 \times 40.42^{0.31276} = 11.7 \text{ Months}$$

Where SE = D + 0.2 x (E − B), where D = 0.28 and B = 0.91. Therefore, SE = 0.31276.

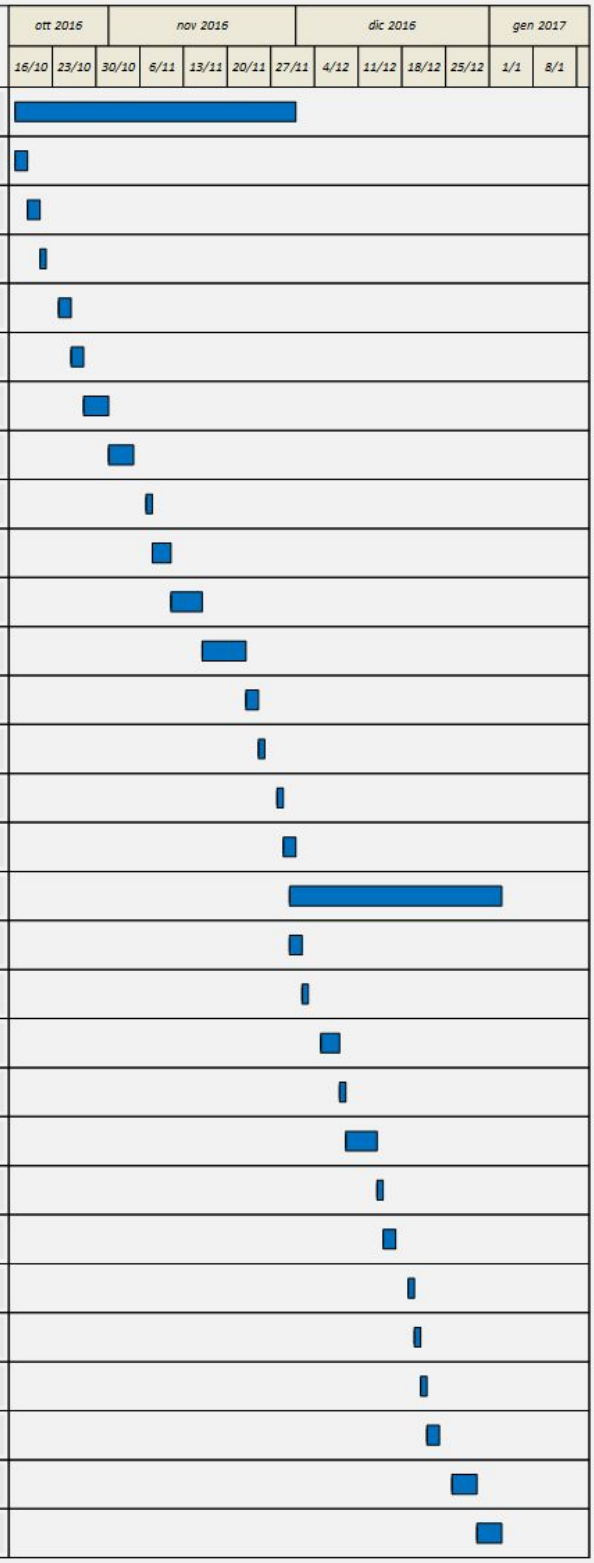
Given the values of Effort and Duration, for the project we can compute the number of required people “N”:

$$N_{people} = \lceil Effort/Duration \rceil = \lceil 1.92 \rceil = 4 \text{ People}$$

4. Schedule

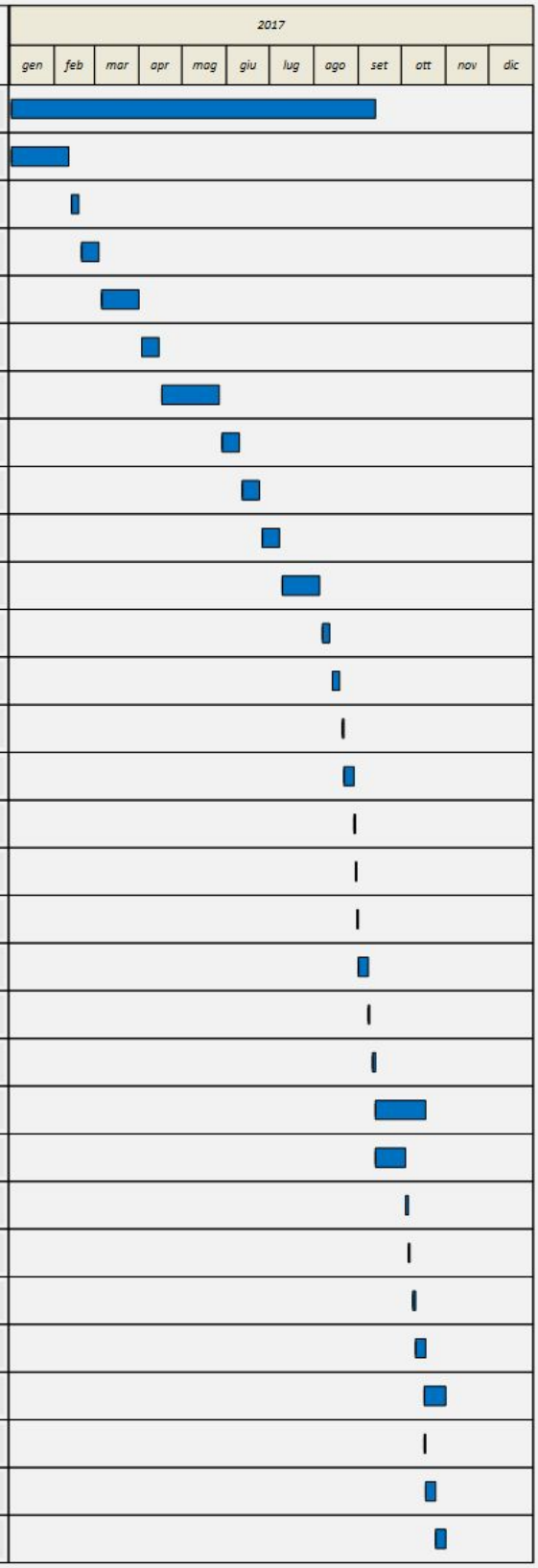
In this section we provide the schedule of our software from the first phase of its design to its commissioning. We were slightly wide on the calculation of the expected time for the implementation, compared to those calculated with COCOMO, foreseeing any problems that may occur in the implementation of the software. In addition to the schedule concerning the design phase personally made by us (shown in the first table) we also added an hypothetical schedule for the remaining phases of realization (represented in the second table). The tables illustrate first the total time for each stage of the project and then, following, the specific ones for each activity.

ID	Nome attività	Inizio	Fine	Durata
1	RASD	17/10/2016	30/11/2016	33g
2	Definition of main functionalities	17/10/2016	18/10/2016	2g
3	Identification of requirements	19/10/2016	20/10/2016	2g
4	Identification of stakeholders	21/10/2016	21/10/2016	1g
5	Drawing of temporary mock-up	24/10/2016	25/10/2016	2g
6	Writing Activity Diagrams	26/10/2016	27/10/2016	2g
7	Definition of all functionalities	28/10/2016	31/10/2016	2g
8	Definition of final requirements	01/11/2016	04/11/2016	4g
9	Meeting with stakeholders	07/11/2016	07/11/2016	1g
10	Drawing of final Mock-up	08/11/2016	10/11/2016	3g
11	Drawing of Use Cases	11/11/2016	15/11/2016	3g
12	Writing of all sequence diagrams	16/11/2016	22/11/2016	5g
13	Design of Class Diagram	23/11/2016	24/11/2016	2g
14	Initial cost, schedule and effort estimation	25/11/2016	25/11/2016	1g
15	Meeting with clients	28/11/2016	28/11/2016	1g
16	Final Refinements	29/11/2016	30/11/2016	2g
17	DD	30/11/2016	02/01/2017	24g
18	Draft of high-level architectures	30/11/2016	01/12/2016	2g
19	Identification of main components	02/12/2016	02/12/2016	1g
20	Design of Component view	05/12/2016	07/12/2016	3g
21	Design of Deployment view	08/12/2016	08/12/2016	1g
22	Design of Runtime view	09/12/2016	13/12/2016	3g
23	Definition of all components	14/12/2016	14/12/2016	1g
24	Identification of some critical algorithms	15/12/2016	16/12/2016	2g
25	Meeting with clients	19/12/2016	19/12/2016	1g
26	Eventual change in the architecture	20/12/2016	20/12/2016	1g
27	Eventual change in the components	21/12/2016	21/12/2016	1g
28	Identification of definitive interfaces	22/12/2016	23/12/2016	2g
29	Drafting of UX and BCE diagrams	26/12/2016	29/12/2016	4g
30	Final refinements	30/12/2016	02/01/2017	2g



Schedule of RASD and DD

ID	Nome attività	Inizio	Fine	Durata
1	DEVELOPEMENT	02/01/2017	12/09/2017	182g
2	Subcomponent development	02/01/2017	10/02/2017	30g
3	External component analysis	13/02/2017	17/02/2017	5g
4	Code inspection	20/02/2017	03/03/2017	10g
5	Unit testing	06/03/2017	31/03/2017	20g
6	Subcomponent integration	03/04/2017	14/04/2017	10g
7	Component development	17/04/2017	26/05/2017	30g
8	Component integration	29/05/2017	09/06/2017	10g
9	Integration testing	12/06/2017	23/06/2017	10g
10	System testing	26/06/2017	07/07/2017	10g
11	Apps Development	10/07/2017	04/08/2017	20g
12	Apps integration	07/08/2017	11/08/2017	5g
13	Apps testing	14/08/2017	18/08/2017	5g
14	Internal demonstration	21/08/2017	21/08/2017	1g
15	Refinements	22/08/2017	28/08/2017	5g
16	Second internal demonstration	29/08/2017	29/08/2017	1g
17	Demonstration to clients	30/08/2017	30/08/2017	1g
18	Demonstration to stakeholders	31/08/2017	31/08/2017	1g
19	Final fixes	01/09/2017	07/09/2017	5g
20	Document revision	08/09/2017	08/09/2017	1g
21	User Manual	11/09/2017	12/09/2017	2g
22	DEPLOYMENT	13/09/2017	17/10/2017	25g
23	System Deployment	13/09/2017	03/10/2017	15g
24	Teaching meeting to amministrators	04/10/2017	05/10/2017	2g
25	Teaching meeting to assistants	06/10/2017	06/10/2017	1g
26	Final meetings	09/10/2017	10/10/2017	2g
27	Final preparations	11/10/2017	17/10/2017	5g
28	START-UP	17/10/2017	31/10/2017	11g
29	System first start	17/10/2017	17/10/2017	1g
30	Real-life simulation	18/10/2017	24/10/2017	5g
31	Main updates and checks	25/10/2017	31/10/2017	5g



Schedule of Development, Deployment and Start-Up

5. Risk Management

In this section are described all the possible hazards, real or potential conditions that can cause injuries for the “Power Enjoy” system, the equipment and the resources.

There are many risks to valuate during the development of a project. First, the main risk to consider is the Customer characteristics, in fact the priority is to understand and communicate as much as possible with the Clients and stakeholders in order to understand the needs, and make a good job in the Requirements analysis.
Another factor to not undervalued are the technological issues that can arise during the building process of our system. This factor can cause extra efforts and consequently extra costs that can impact on the roadmap previously programmed.
Also guarantee the performance and security of the sensitive data stored in the Database is a continuous risk for the entire management system, but even the dynamical changes inside a company has to be keep on track and informed about it.

In the following table we present a summary of the potential risks explained above and other ones.

Risk	***Probability***	***Effect***
Misunderstanding of the stakeholders needs	High	Catastrophic
The organization is restructured so that different management are responsible for the project	High	High Serious
The Database cannot match the expected performance	Moderate	Serious
Technological Issues	Moderate	Moderate
Fault in the software	Moderate	Serious

For all the potential problems and even more of the already specified one, is good attitude to allocate extra time and resources.

4. Effort Spent

To redact this document, we spent 15 hours per person. We also report here the overall amount of hours per person required by the project.

Document	***Hours of work per Person***
RASD	45
DD	30
ITPD	19
PP	15
Overall Document Revision	3
Total	**112**