Project Plan Document

January 22, 2017

Giacomo Bossi

Marco Nanni

Content

1. INTRODUCTION

1.1 Revision history

Version	Date	Authors	Summary
1.0	22/01/2017	Giacomo Bossi, Marco Nanni	First release

1.2 Purpose and scope

This document is the Project Plan document for PowerEnJoy, which is a digital management system for a car sharing service that exclusively employs electric cars. The Project Plan aims at defining a plan for our project, in order to have a size and cost estimation of the system to be developed. By means of this document, project managers can have a clear view of the system, in order to split the project into tasks and assign them to people, estimate time and resources needed to complete each tasks, assess and monitor risks that can affect our project, establish realistic deadlines, manage the team work.

This document is structured in 5 main sections:

- in section 2 we are going to estimate the size of our project by an algorithmic cost modelling approach: in this way we are going to evaluate the expected range of Kilo-Source Lines Of Code by means of a formula based on Function Points. These points represent our system characteristics such as Internal and External Logic Files, External Inputs, External Inquiries and External Outputs

- in section 3 we are going to evaluate our scale and cost drivers values in order to estimate the project effort in Person Months, the Time to Develop, the Number of Persons needed in the team to accomplish the goal and the total cost for the project

- in section 4 we are going to represent the time needed to accomplish each task by means of Gantt diagrams

- in section 5 we are going to represent the number of hours and tasks assigned to each member by means of a Gantt diagram

- in section 6 we are going to define the possible risks of our project, their relevance and our intended actions to avoid and resolve them

1.3 Definitions, acronyms and abbreviations

- RASD: Requirements Analysis and Specification Document
- DD: Design document
- ITPD: Integration Test Plan Document
- ILF: Internal Logic File
- ELF: External Logic File
- EIs: External Inputs
- EQs: External Inquiries
- EOs: External Outputs
- PM: Person Months
- KSLOC: Kilo Source Line Of Code
- FP: Function Point
- UFP: Unadjusted Function Points
- SF: Scale Factor
- EM: Effort Multiplier
- COCOMO II: Constructive Cost Model, version 2
- App: Application

1.4 Reference Documents

- Assignment 4 of the specification document
- http://sunset.usc.edu/research/COCOMOII/expert_cocomo/drivers.html
- http://csse.usc.edu/csse/research/COCOMOII/cocomo2000.0/CII_modelman2000.0.pdf

2. PROJECT SIZE ESTIMATION

In this section we are estimating the PowerEnJoy System size, measured in kilo-source lines of code (KSLOC), by applying an algorithmic cost modelling technique. According to this approach, we evaluate some attributes and characteristics of our project in order to obtain a range of values that represents the minimum and maximum expected lines of code that have to be written by our programmers. The technique consists of assigning Function Points to several characteristics of the project: these points depends on the function type and its weight.

For Internal Logic Files (ILFs) and External Logic Files (ELFs) we have:

	Data Elements		
Record Elements	1-9	10-39	40+
1	Low	Low	Average
2-7	Low	Average	High
8+	Average	High	High

For External Outputs (EOs) and External Inquiries (EQs) we have:

	Data Elements		
File Types	1-7	8-24	25+
0-1	Low	Low	Average
2-4	Low	Average	High
5+	Average	High	High

For External Inputs (EIs) we have:

	Data Elements		
File Types	1-4	5-14	15+
0-1	Low	Low	Average
2-4	Low	Average	High
5+	Average	High	High

Hence, the Unadjusted Function Point (UFP) Complexity Weight is:

	Complexity Weight		
Function Type	Low	Average	High
Internal Logic Files (ILFs)	8	10	15
External Logic Files (ELFs)	6	8	10
External Inputs (EIs)	3	5	7
External Outputs (EOs)	4	6	8
External Inquiries (EQs)	3	5	7

2.1 Internal Logic Files (ILFs)

Internal Logic Files are homogeneous sets of data used and managed by the application. Hence, we have identified five ILFs for our system:

- Car: consists of car status, car position, battery level and other details, so it has an average complexity
- User: formed by user status, user position and user information provided during Sign Up phase; it has a low complexity
- Employments: consists of the records used to keep track of all the employments between users and cars, hence it has a high complexity
- Zone: refers to safe and unsafe areas where the user can park his/her car or not. It also includes zones with power grid stations, so high complexity
- Map: consists of the map of the city of Milan, hence average complexity

Internal Logic Files (ILFs)	Complexity	Function Points
Car	Average	10
User	Low	8
Employment	High	15
Zone	High	15
Map	Average	10
Total		58

2.2 External Logic Files (ELFs)

External Logic Files are homogeneous sets of data used by the application but generated and maintained by other applications. Hence, we have identified two ELFs for our system:

- Payment Service: the Payment Gateway of our system communicates with an external Payment Service in order to check the validity of the user's credit card and allow users to pay for their rides. Since the operations are critical and very frequent, it has a high complexity
- Driving License Service: the Driving License Gateway communicates with an external Driving License Service in order to check the validity of the user's driving license. Since it is performed only during the Sign Up phase and when the license expires, it has average complexity

External Logic Files (ELFs)	Complexity	Function Points
Payment Service	High	10
Driver License	Average	8
Total		18

2.3 External Inputs (EIs)

External Inputs are elementary operations that elaborate data coming from the external environment. We have identified several EIs for our system:

- Sign Up: has a high complexity since during this phase the system has to check all the fields filled by the user, the provided credit card validity and the provided driving license validity
- Login: has a low complexity given that it requires only a username and password check
- Reserve a Car: average complexity. This operation has to check if the car is already available, change the car and user statuses
- Update Car/User Position: low complexity since it only requires the position from the GPS installed inside the car or in the user's smartphone
- Open a Car: average complexity. It has to check if the right user is near the car and change user and car statuses
- Enter PIN Code: low complexity since it is a simple check
- Occupied Seats: low complexity since it only retrieves the information provided by the camera sensors installed inside the car
- Recharge the Car: average complexity. It has to verify that the power grid is plugged into the car and apply the discount
- End Ride: average complexity since it has to check that the car is parked in a safe area
- New, Update, End Employment: high complexity given that they consist of all insertions and updates of employments between users and cars
- Update Car/User Status: average complexity. It has to verify and change the status
- Check Credit Card/Driving License: average. It has to check the credit card/driving license validity by communicating with an external service (Payment Service/Driving License Service respectively)

External Inputs (EIs)	Complexity	Function Points
Sign Up	High	7
Login	Low	3
Reserve a Car	Average	5
Update Car/User Position	Low	2·3
Open a Car	Average	5
Enter PIN Code	Low	3
Occupied Seats	Average	5
Recharge Car	Average	5
End Ride	Average	5
New, Update, End Employment	High	3·7
Update Car/User Status	Average	2·5
Check Credit Card/Driver License	High	2·7
Total		92

2.4 External Inquiries (EQs)

External Inquiries are elementary operations that involve input and output. We have identified three EQs for our system:

- Select a Car: low complexity since it only shows information about a selected car
- Search a Car: average complexity given that it has to find all the available cars located in that specific area
- Show User Information: low complexity. It only shows the user personal information

External Inquiries (EQs)	Complexity	Function Points
Select a Car	Low	4
Search a Car	Average	6
Show User Information	Low	4
Total		14

2.5 External Outputs (EOs)

External Outputs are elementary operations that generate data for the external environment. We have identified four EOs for our system:

- Show Current Charge: low complexity. It shows the current charge of the user during his/her ride
- Show Area: low complexity. It informs the user if he/she is in a safe area
- Show nearest Charging Point: average complexity. It recognizes the car's position and shows the nearest charging point
- Send Payment: high complexity since it has to communicate with an external bank service

External Outputs (EOs)	Complexity	Function Points
Show Current Charge	Low	3
Show Area	Low	3
Show nearest Charging Point	Average	5
Send Payment	High	7
Total		18

2.6 Overall estimation

Function Type	Value
ILFs	58
ELFs	18
EIs	92
EQs	14
EOs	18
Total	200

Hence, we can estimate the range of the total number of lines of code required to satisfy the requirements as:

$$SLOC_{min} = 200 \cdot 46 = 9200$$

$$SLOC_{max} = 200 \cdot 67 = 13400$$

3. COST AND EFFORT ESTIMATION

3.1 Scale Drivers

Scale Factors	Very Low	Low	Nominal	High	Very High	Extra High
PREC	Thoroughly unprecedented 6.20	Largely unprecedented 4.96	Somewhat unprecedented 3.72	Generally familiar 2.48	Largely familiar 1.24	Thoroughly familiar 0.00
FLEX	Rigorous 5.07	Occasional relaxation 4.05	Some relaxation 3.04	General conformity 2.03	Some Conformity 1.01	General Goals 0.00
RESL	Little (20%) 7.07	Some (40%) 5.65	Often (60%) 4.24	Generally (75%) 2.83	Mostly (90%) 1.41	Full (100%) 0.00
TEAM	Very difficult interactions 5.48	Some difficult interactions 4.38	Basically cooperative Interactions 3.29	Largely cooperative interactions 2.19	Highly cooperative interactions 1.10	Seamless interactions 0.00
PMAT	Level 1 lower 7.80	Level 1 upper 6.24	Level 2 4.68	Level 3 3.12	Level 4 1.56	Level 5 0.00

In this part of the document we are going to explain how we have evaluated and chosen the rating of each scale driver:

- *Precedentedness (PREC): Low*
  This scale driver quantifies the similarity with precedent projects that has been already developed. We set this driver to low because this product is not like several previously developed projects

- *Development Flexibility (FLEX): Nominal*
  In this case we set the value to nominal because there is a considerable need of conformance with pre-established requirements and external interfaces, but there are no specifications about the technology to be used

- *Architecture / Risk Resolution (RESL): Very High*
  This value comes from the risks analysis explained in chapter 5 of this document

- *Team Cohesion (TEAM): Very High*
  The Team Cohesion scale measures the sources of project turbulence caused by difficulties in synchronizing the project stakeholders. By analysis of the TEAM rating component it's emerged a very high value, determined by the high competences of the stakeholders.

- *Process Maturity (PMAT): Level 2 - Nominal*
  Since this is a value based on the capabilities of the software developers, we evaluate this parameter as Nominal.

Scale Driver	Factor	Value
Precedentedness	Low	4.96
Development Flexibility	Nominal	3.04
Architecture/Risk Resolution	Very High	1.41
Team Cohesion	Very High	1.10
Process Maturity	Level 2	4.68
Total		15.19

3.2 Cost Drivers

3.2.1 Product Factors

- **Required Software Reliability (RELY)**
  PowerEnJoy System represents the only way to rent the electric car, so a system failure would implicate a total loss of the profit. In this way, we evaluate the RELY parameter as High, due to a high financial loss.

RELY Descriptors	Slight Inconvenience	Low, Easily recoverable losses	Moderate	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	1.10	1.26	n/a

- **Database Size (DATA)**
  This cost driver attempts to capture the effect that large test data requirements have on product development. We estimate the dimension of our main Database approximately at 4 GB. Since we calculated the SLOC between 9000 and 14000, the ratio Database Bytes / program SLOC is greater than a thousand, so we have set the DP ratio as Very High.

DATA Descriptors		DP Ratio < 10	10 < DP Ratio < 100	100 < DP Ratio < 1000	DP Ratio > 1000	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	n/a	0.90	1	1.14	1.28	n/a

- **Product Complexity (CPLX)**
  Since we are evaluating only the computational complexity of the application server, we estimate this cost driver as an average of the following values:
  - *User Interface Management Operation = Very Low*
    Since the user interacts with the PowerEnJoy System through the application, the computational effort caused by the user interface is spent by the user device
  - *Data Management Operation = High*
    The Database is not distributed, but it has some complex data structures to handle
  - *Device Dependent Operation = Very High*
    Each operation requires a high communication with the I/O devices
  - *Control Operation = Nominal*
    The costs of control operation are related to some inter module control
  - *Computational Operation = Nominal*
    There are no recursive operations made on the data sets; we suppose the employment of standard math and basic matrix/vector operations

  So, the average of the component of the product complexity is High.

CPLX Driver

Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	n/a	0.90	1	1.14	1.28	1,74

- **Required Reusability (RUSE)**
  This cost driver accounts for the additional effort needed to construct components intended for reuse on current or future projects. In the case of the PowerEnJoy System, the reusability of the code is limited to the project itself, so we set this cost value as Nominal.

RUSE Descriptors		**None**	**Across Project**	**Across Program**	**Across Product Line**	**Across Multiple Product Line**
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	n/a	0.95	1	1.07	1.15	1.24

- **Documentation match to life-cycle needs (DOCU)**
  The rating scale for the DOCU cost driver is evaluated in terms of the suitability of the project documentation to its life-cycle needs. We foresee to produce a right-sized documentation for the life-cycle needs, so we set this parameter to Nominal.

DOCU Descriptors	Many Life-Cycle needs uncovered	Some life-cycle needs uncovered	Right-sized for 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	1.11	1.23	N/A

3.2.2 Platform Factors

- **Execution Time Constraint (TIME)**
  This is a measure of the execution time constraint imposed upon a software system. The rating is expressed in terms of the percentage of involved CPU time during the execution of the application. The PowerEnJoy System can be considered as a complex software, so we expect a great usage of the CPU and we set the rating level as Very High.

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	1.11	1.29	1.63

- **Main Storage Constraint (STOR)**
  This rating represents the degree of main storage constraint imposed on a software system. We expect to have more storage space available than the one required by our database, so we have set this parameter to Nominal.

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	1.05	1.17	1.46

- **Platform Volatility (PVOL)**
  The PowerEnJoy System is not supposed to be changed very often even if it could be possibly extended to other cities. So, we estimate this possibility as a minor change and we evaluate the Platform Volatility rating level as Low.

PVOL Descriptors		**Major change every 12 months Minor change every 1 month**	**Major: 6 mo. Minor: 1 mo.**	**Major: 2 mo. Minor: 2 wk.**	**Major: 2 wk. Minor: 2days**	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	n/a	0.90	1	1.14	1.28	n/a

3.2.3 Personnel Factors

- **Analyst Capability (ACAP)**
  Since Analysts are personnel who work on requirements, high-level design and detailed design, and given that our team has spent many hours in doing this first and important part, we think that the rating level is High.

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 (PCAP)**
  The ability of the programmers as a team is High, given that the team is very cohesive.

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

- **Applications Experience (AEXP)**
  Since we have already used Java but never Java Enterprise Edition, the applications experience of the project team developing the software system is Low.

AEXP 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 (PLEX)**
  Since we do not have a high level of platform experience by recognizing the importance of the use of more powerful platforms like graphic user interface, networking and distributed middleware capabilities, our PLEX factor is Low.

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

- **Language and Tool Experience (LTEX)**
  Our team's programming language and software tool experience is Nominal.

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

- **Personnel Continuity (PCON)**

  Since our team was made exclusively for this project, the project's personnel turnover is very low, and so the continuity is High.

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	n/a

3.2.4 Project Factors

- **Use of Software Tools (TOOL)**
  Our application environment is strong and well integrated, so this factor is set to High.

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 (SITE)**
  To accomplish this document, the team has worked in the same city with a wideband electronic communication, so the SITE parameter is set to High.

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: Communication Descriptors	Some phone, mail	Individual phone, FAX	Narrow band email	Wideband electronic communication	Wideband electronic communication, occasional video conferences	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 (SCED)**

  This rating measures the schedule constraint imposed on the project team developing the software. The ratings are defined in terms of the percentage of schedule stretch-out or acceleration with respect to a nominal schedule for a similar project. In this way, our project was not stretch-out even if we had several deadlines, so the factor is set as High.

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

Finally, our Cost Drivers values are:

Cost Driver	Rating Level	Effort Multiplier
Product Factors		
Required Software Reliability (RELY)	High	1.10
Database Size (DATA)	Very High	1.28
Product Complexity (CPLX)	High	1.14
Required Reusability (RUSE)	Nominal	1.00
Documentation match to life-cycle needs (DOCU)	Nominal	1.00
Platform Factors		
Execution Time Constraint (TIME)	Very High	1.29
Main Storage Constraint (STOR)	Nominal	1.00
Platform Volatility (VOL)	Low	0.90
Personnel Factors		
Analyst Capability (ACAP)	High	0.85
Programmer Capability (PCAP)	High	0.88
Application Experience (AEXP)	Low	1.10
Platform Experience (PLEX)	Low	1.09
Language and Tool Experience (LTEX)	Nominal	1.00
Personnel Continuity (PCON)	High	0.90
Project Factors		
Usage of Software Tools (TOOL)	High	0.90
Multisite Development (SITE)	High	0.93
Required Development Schedule (SCED)	High	1.00
Total		1.259

3.3 Effort equation

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

where

$$A = 2.94\ (for\ COCOMO\ II.2000)$$

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

$$B = 0.91\ (for\ COCOMO\ II.2000)$$

Hence we have:

$$E = 0.91\ + 0.01\ \cdot (4.96 + 3.04 + 1.41 + 1.10 + 4.68) = 1.0619$$

$$PM_{min} = 2.94\ \cdot\ 9.200^{1.0619}\ \cdot 1.259\ \cong 39$$

$$PM_{max} = 2.94\ \cdot\ 13.400^{1.0619}\ \cdot 1.259\ \cong 58$$

3.4 Schedule estimation

Time to develop equation:

$$TDEV = [C \cdot (PM_{NS})^F] \cdot \frac{SCED\%}{100}$$

where

$$F = (D + 0.2 \cdot [E - B])$$

B = 0.91

C = 3.67

D = 0.28

Hence we have:

$$F = 0.31038$$

$$TDEV_{min} = 11.44\ months$$

$$TDEV_{max} = 12.94\ months$$

$$Number\ of\ persons_{min} = \frac{PM_{min}}{TDEV_{min}} = 3.4 \sim 4$$

$$Number\ of\ persons_{max} = \frac{PM_{max}}{TDEV_{max}} = 4.48 \sim 5$$

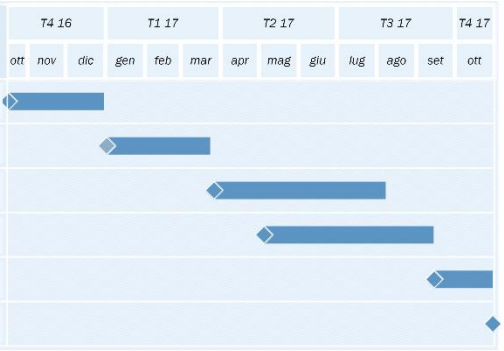
We expect an average employee salary of 5.000€/month. Hence, the range of the total cost of the project can be estimated as:

$$total\ cost = salary \cdot number\ of\ persons \cdot TDEV \sim €240.000 - €325.000$$

4. SCHEDULE

In this part of the document we are going to describe the complete time process needed to achieve the objective set by the stakeholders. The tasks time has been deducted by the COCOMO computation and analysis, so we estimate 12 months between the definition of the requirements and the final product, divided in five phases. We will use a Waterfall Model to develop the project, even if there will be an overlapping between the development and the testing phase.

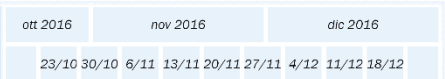
ID	Activity Name	Begin	End	Duration
1	Requirement Analysis and Specification Document	17/10/2016	30/12/2016	440h
2	Design Document	02/01/2017	23/03/2017	472h
3	Development	27/03/2017	07/08/2017	680h
4	Integration Test	05/05/2017	14/09/2017	320h
5	Deployment	15/09/2017	30/10/2017	256h
6	Release	31/10/2017	31/10/2017	8h



4.0 Overall description of the production phases

In the following Gantt Diagram, we are focusing on the Requirement Analysis. In this phase, there will be several meetings with our stakeholders in order to identify their wills.

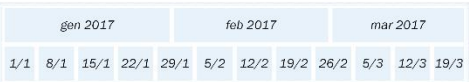
ID	Activity Name	Begin	End	Duration
1	**RASD**	**17/10/2016**	**30/12/2016**	**55g**
2	Requirements Identification	17/10/2016	21/10/2016	5g
3	Problem Analisys	21/10/2016	25/10/2016	3g
4	Definition of main functionalities	26/10/2016	03/11/2016	7g
5	Definition of possible use case	03/11/2016	07/11/2016	3g
6	Mockup	07/11/2016	10/11/2016	4g
7	First document release	10/11/2016	17/11/2016	6g
8	First meeting with stakeholder	17/11/2016	17/11/2016	1g
9	Revision of the Document	18/11/2016	30/11/2016	9g
10	Revision Cost Estimation	01/12/2016	05/12/2016	3g
11	Final Requirements Document	06/12/2016	16/12/2016	9g
12	Second Meeting with Stakeholder	19/12/2016	19/12/2016	1g
13	Second Revision of the document	19/12/2016	23/12/2016	5g
14	Final Refinements	27/12/2016	30/12/2016	4g



4.1 Detailed description of the RASD phase. Oct 2016 – Dec 2016.

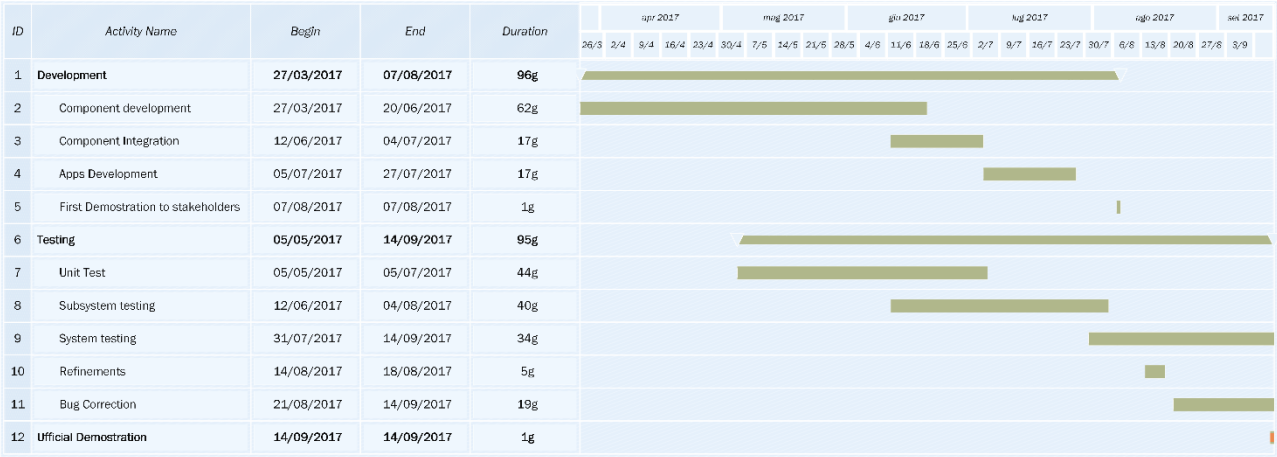
The Design Document is composed by many sub activities; indeed, the architectural process begins from a high level and it ends with the description of subcomponents, their interfaces and the most important algorithm. During this phase, there are two meetings with the stakeholders, and after them it begins the development phase.

ID	Activity Name	Begin	End	Duration
1	**DD**	**02/01/2017**	**24/03/2017**	**60g**
2	Identification of main subsystems	02/01/2017	13/01/2017	10g
3	Identification of main components	13/01/2017	18/01/2017	4g
4	UML Diagrams	18/01/2017	20/01/2017	3g
5	First DD Document	20/01/2017	27/01/2017	6g
6	Third Meeting with stakeholders	30/01/2017	30/01/2017	1g
7	Revision of the document	31/01/2017	10/02/2017	9g
8	Algorithm design and component Interfaces	10/02/2017	20/02/2017	7g
9	Refinement of the Design Document	20/02/2017	28/02/2017	7g
10	Fourth Meeting with stakeholder	01/03/2017	01/03/2017	1g
11	Final Design Decision	01/03/2017	10/03/2017	8g
12	Final Design Document	10/03/2017	24/03/2017	11g



4.2 Detailed description of the DD phase. Gen 2017 – Mar 2017

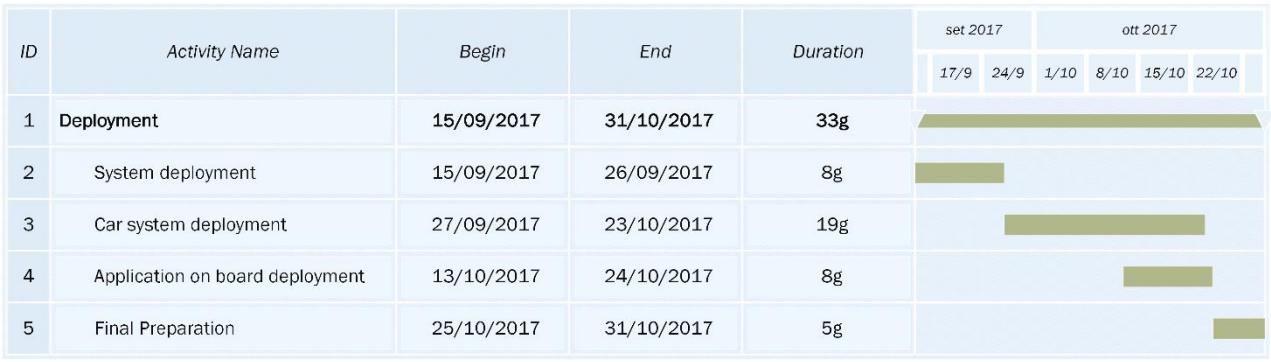
Development and Testing phases proceed together; in fact, after an initial development of the base components, it is necessary to test them before adding their integration. At the end of the development process there will be a beta demonstration of the system for the stakeholders. After that, the remaining tests will be accomplished, and the final version will be released.



ID	Activity Name	Begin	End	Duration
1	**Development**	**27/03/2017**	**07/08/2017**	**96g**
2	Component development	27/03/2017	20/06/2017	62g
3	Component Integration	12/06/2017	04/07/2017	17g
4	Apps Development	05/07/2017	27/07/2017	17g
5	First Demostration to stakeholders	07/08/2017	07/08/2017	1g
6	**Testing**	**05/05/2017**	**14/09/2017**	**95g**
7	Unit Test	05/05/2017	05/07/2017	44g
8	Subsystem testing	12/06/2017	04/08/2017	40g
9	System testing	31/07/2017	14/09/2017	34g
10	Refinements	14/08/2017	18/08/2017	5g
11	Bug Correction	21/08/2017	14/09/2017	19g
12	**Ufficial Demostration**	**14/09/2017**	**14/09/2017**	**1g**

4.3 Detailed description of the Development and Testing phases. Mar 2017 – Sep 2017

Before the market launch, there is the Deployment phase, in which the application server will be initialized and the car's System on Board will be installed. Finally, all the PowerEnJoy electric cars will be tested in the city of Milan.



ID	Activity Name	Begin	End	Duration
1	**Deployment**	**15/09/2017**	**31/10/2017**	**33g**
2	System deployment	15/09/2017	26/09/2017	8g
3	Car system deployment	27/09/2017	23/10/2017	19g
4	Application on board deployment	13/10/2017	24/10/2017	8g
5	Final Preparation	25/10/2017	31/10/2017	5g

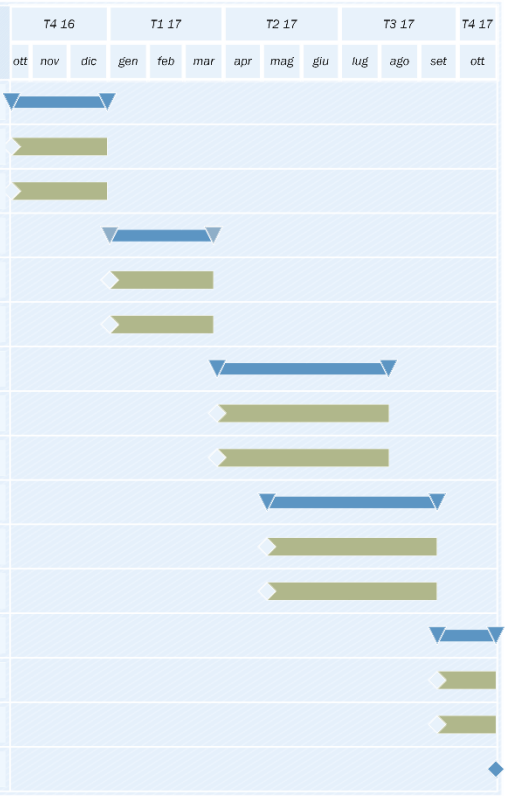
4.4 Detailed description of the Deployment phase. Sep 2017 – Oct 2017

5. RESOURCE ALLOCATION

In this section we are going to represent how the main tasks will be divided between two team members, Giacomo and Marco. As stressed in the next section about risks, no one task has been completely assigned to a single team member. Since the team is very cohesive, Giacomo and Marco will proceed in parallel during all the project phases, starting from the Requirements Analysis and Specification Document.

By means of the algorithmic modelling approach described in section two, we have estimated that our project development will last about 12 months, starting from October 17, 2016.

ID	Activity Name	Begin	End	Durata.
1	**Requirement Analysis and Specification Document**	**17/10/2016**	**30/12/2016**	**440h**
2	Giacomo Bossi	17/10/2016	30/12/2016	440h
3	Marco Nanni	17/10/2016	30/12/2016	440h
4	**Design Document**	**02/01/2017**	**23/03/2017**	**472h**
5	Giacomo Bossi	02/01/2017	23/03/2017	472h
6	Marco Nanni	02/01/2017	23/03/2017	472h
7	**Development**	**27/03/2017**	**07/08/2017**	**680h**
8	Giacomo Bossi	27/03/2017	07/08/2017	680h
9	Marco Nanni	27/03/2017	07/08/2017	680h
10	**Integration Test**	**05/05/2017**	**14/09/2017**	**320h**
11	Giacomo Bossi	05/05/2017	14/09/2017	320h
12	Marco Nanni	05/05/2017	14/09/2017	320h
13	**Deployment**	**15/09/2017**	**30/10/2017**	**256h**
14	Giacomo Bossi	15/09/2017	30/10/2017	256h
15	Marco Nanni	15/09/2017	30/10/2017	256h
16	**Release**	**31/10/2017**	**31/10/2017**	8h



6. RISK MANAGEMENT

The first possible risks to think about are those related to stakeholders. Green Energy Company - the company that has launched PowerEnJoy - is supported by Electric Car Company and the Italian Government. Electric Car Company, which owns part of the PowerEnJoy actions, will provide all the electric cars for the service. The Italian government has given incentives to the PowerEnJoy project in order to reduce CO_2 emissions throughout the Country, starting from the biggest and most polluted city, Milan. A loss of these two big stakeholders could lead Green Energy Company in bankrupt, but we think that the probability is very low. Electric Car Company is an emerging car company which is expanding its commerce all over the world, since the electric cars distribution is growing every day. Moreover, a political change in the Italian government could not cause big problems given that the project has already been started and a budget has already been established and guaranteed. However, we are going to meet our stakeholders several times, especially during the first phases where there could be more incomprehension and problems.

Another risk could be changes of privacy laws, for example those regarding cameras installed inside the car. The purpose of these sensors is to check if there are passengers inside the car in order to apply the discount to the user. If a new law denies the sensors' employment, they can be replaced by other sensors, like weight sensors.

Also the screens on the cars (System on Board) could get involved by new laws which deny the user to use any kind of electronic device while he/she is driving. In this way, a voice will substitute the screen, indicating the path to follow in order to reach the destination and summarizing the user's details and charges at the end of the ride.

External services our system is going to interface with could also be a risk. In fact, our Payment Gateway always interacts with external bank services in order to accomplish users' transactions. Also the Driving License Gateway communicates with the Italian Driving License Service in order to check the validity of the user's driving license. However, we are sure that all these systems provide a high level of accuracy and availability of their services.

A risk related to our application could be the acceptance and satisfaction of PowerEnJoy to our users. Even if there already exist several car-sharing services, our application should be easy to use and suitable for most of the smartphones available in the market nowadays. Moreover, we are going to implement and release different versions of our product, according to the technological changes.

Risks related to our software development are absences and turnover of some key members of our team. Even if our team is really professional, no one task has been assigned completely to just one person. In this way, a loss of a team member would not lead to catastrophic events. Another risk that involves PowerEnJoy system is the loss of some critical parts of the code and data. To avoid this, we are going to back-up the most critical components and subsystems.

Risk	Probability	Effect
Loss of the stakeholders	Very Low	Catastrophic
Incomprehension with stakeholders	Moderate	Marginal
Camera sensors privacy changes	Moderate	Marginal
System on board (SoB) restrictions	Moderate	Critical
Loss of Payment and Driving License Services	Very Low	Catastrophic
Low level of satisfaction of users about the App	High	Critical
Loss of key team members	Low	Critical
Loss of code and data	Moderate	Critical

Risk	Strategy
Incomprehension with stakeholders	More meetings with stakeholders
Camera sensors privacy changes	Replace camera sensors with weight sensors
System on board (SoB) restrictions	Replace SoB with a vocal command device
Low level of satisfaction of users about the App	Satisfiability analysis, more changes and releases
Loss of key team members	No tasks assigned to exactly one person
Loss of code and data	Backups

7. EFFORT SPENT

Giacomo Bossi:

10/01: 1.5hrs

12/01: 3hrs

13/01: 2.5hrs

14/01: 2hrs

16/01: 3hrs

17/01: 2.5hrs

18/01: 2hrs

20/01: 1hr

21/01: 2hrs

Marco Nanni:

10/01: 1.5hrs

11/01: 2hrs

12/01: 2hrs

14/01: 3hrs

16/01: 3hrs

17/01: 2.5hrs

18/01: 2.5hrs

20/01: 2hr

21/01: 2hrs