POLITECNICO
MILANO 1863

Project Plan

Lo Bianco Riccardo - Manzoni Mirco - Mascellaro Giuseppe

January 21, 2017
v1.0

Contents

1 Introduction **1**
1.1 Purpose and scope . . . 1
1.2 Acronyms . . . 2

2 Project size, cost estimation and effort estimation **3**
2.1 Size estimation: Function Points . . . 3
2.1.1 Internal Logic Files (ILFs) . . . 4
2.1.2 External Interface Files (EIFs) . . . 5
2.1.3 External Inputs (EIs) . . . 5
2.1.4 External Inquiries (EQs) . . . 6
2.1.5 External Outputs (EOs) . . . 6
2.2 Overall estimation . . . 7

3 COCOMO II analysis **8**
3.1 Software scale drivers analysis . . . 8
3.2 Software cost drivers analysis . . . 9
3.2.1 Product factors analysis . . . 9
3.2.2 Platform factors analysis . . . 9
3.2.3 Personnel factors analysis . . . 10
3.2.4 Project factors analysis . . . 10

4 Effort estimation **11**
4.1 Screen view . . . 11
4.2 Effort (average) estimation . . . 12
4.3 Effort (over) estimation . . . 12

5 Scheduling **13**
5.1 RASD . . . 14
5.2 DD . . . 15
5.3 Development . . . 16
5.4 Deployment and Start Up . . . 17

6 Resource allocation **18**
6.1 Resource allocation - Lo Bianco Riccardo . . . 19
6.2 Resource allocation - Manzoni Mirco . . . 20
6.3 Resource allocation - Mascellaro Giuseppe . . . 21

7 Risk analysis **22**

8 References **23**

9 Hours of work **23**

1 Introduction

1.1 Purpose and scope

This document represents the Project Plan Document for the PowerEnJoy project. Its main purpose is to analyze the expected complexity of PowerEnJoy and assist the project leader in the phase of cost and effort estimation. This information can be subsequently used as a guidance to define the required budget, the resources allocation and the schedule of the activities, as well as a compedium for stakeholders to decide wether to invest in the project.
In the first section, we use the approach based on Function Points, in order to provide an estimate of the expected size of PowerEnJoy both in terms of lines of code and of the cost/effort required to actually develop it.
In the first part we use the Function Points approach to define the size of the system to be developed in terms of number of functionalities to be provided. Then we use the results to perform an appropriate analysis through the use of COCOMO II, a tool capable of providing an accurate estimation of the cost of the project in terms of time and money.
In the second section we explore the possible flow of the developing through its different phases, in order to allocate the resources in an optimal way along the time necessary to complete the job.
Finally, we discuss on the possible risks that PowerEnJoy could face during the various phases of the project and elaborate general conclusions.

1.2 Acronyms

- FP: Function Points.
- ILF: Internal Logic File.
- EIF: External Interface File.
- EI: External Input.
- EO: External Output.
- EQ: External Inquiry.
- DBMS: Database Management System.
- API: Application Programming Interface.

1.3 Reference documents

- **RASD** (RASD.pdf)
- **DD** (DD.pdf)
- **CSSE** (www.csse.usd.edu)
- **Wikipedia** (www.wikipedia.org)
- **Sunset COCOMO II Guide** (www.sunset.usc.edu)
- **Project Management Basics + Advanced** (Slides of the course)

2 Project size, cost estimation and effort estimation

In this paragraph we provide our extimation regarding the expected size, cost and required effort for the development of PowerEnJoy application.
For what concerns the size estimation we used function points approach, taking into account a realistic amount of lines of code for each of the major functionalities of the application. All the analysis is brought on from the point of view of the business logic, without taking into account the user interface.

2.1 Size estimation: Function Points

Through the following chapter, we present considerations related to the different software elements composing the system. Here we resume their role one by one, reporting the pre-defined set of weights used to compute the function points:

- **ILFs**: files used to store persistent information necessary for the functions of the logic.
- **EIFs**: files used to store persistent information necessary for the functions that manage the interaction with the external components (ex. APIs).

For Internal Logic Files and External Interface Files

	Data Elements		
Record Elements	*1-19*	*20-50*	*51+*
1	Low	Low	Avg
2-5	Low	Avg	High
6+	Avg	High	High

- **EIs**: elementary external input activities

For External Inputs

	Data Elements		
File Types	*1-4*	*5-15*	*16+*
0-1	Low	Low	Avg
2-3	Low	Avg	High
4+	Avg	High	High

- **EOs**: elementary external output activities
- **EQs**: elementary external output activities

For External Outputs and External Inquiries

	Data Elements		
File Types	*1-5*	*6-19*	*20+*
0-1	Low	Low	Avg
2-3	Low	Avg	High
4+	Avg	High	High

So we can resume the weights we will use along the discussion as follows.

UFP Complexity Weights

	Complexity Weight		
Function Type	*Low*	*Average*	*High*
Internal Logic Files	7	10	15
External Interfaces Files	5	7	10
External Inputs	3	4	6
External Outputs	4	5	7
External Inquiries	3	4	6

2.1.1 Internal Logic Files (ILFs)

Here we report the ILFs used by PowerEnJoy to store the information it needs to offer the required functionalities. The ILFs presented below reflects the entities designed for the DB of the application (refer to the DD for a complete description of the DB structure).

ILF	Complexity	FPs
PaymentInfo	Low	7
Users	Average	10
Operators	Average	10
Invoices	Low	7
Notifications	Low	7
Rides	Low	7
Cars	Average	10
MinorIssues	Low	7
SafeParkingAreas	Average	10
SpecialParkingAreas	Average	10
Total		85

2.1.2 External Interface Files (EIFs)

Our system manages some operations with the aid of external APIs. We need to interact with Maps API and Navigation API in order to show the map and to guide the user during his trip. Similarily, we use the Payment Manager API in order to process our clients' payments and Driving License Authority API to elaborate information about the driving licence of a new user that join our service, and lastly we use an e-mail API to send mail to users.

EIF	Complexity	FPs
Maps API	Low	7
Navigation API	Average	10
Payment Information API	Average	10
Driving License Authorithy API	Low	7
E-Mail API	Low	7
Total		34

2.1.3 External Inputs (EIs)

In this paragraph we take into account the transactional function point, allowing users to maintain Internal Logical Files (ILFs) through the ability to add, change and delete the data.
Between the functional requirements listed in the RASD, we have identified the following functionalities, labelled as External Input.

EI	Complexity	FPs
Sign up	Average	4
Log in	Low	3
Reservation request	Low	3
Cancel a reservation	Low	3
Search for cars located nearby or by specifying an address	Low	3
Navigation to a car	Low	3
Navigation to a place	Low	3
Enable money saving option	Low	3
Make a report about a car	Average	4
Unlock reserved car	Low	3
Request a new password	Low	3
Request a new password	Low	3
Show last rides' invoices	Low	3
Insertion of password in order to ignite the engine	Low	3
Total		42

2.1.4 External Inquiries (EQs)

External Inquiries represent data retrieval requests performed by a user that don't require complex computations.
PowerEnJoy offers to the user information about invoices, reservations, safe and special parking areas and cars. It also offers to our operators information about minor issues and rides.

EQ	Complexity	FPs
Retrieve last invoices	Average	4
Retrieve reservation info	Low	3
Retrieve safe parking area	Low	3
Retrieve special parking area	High	6
Retrieve car's information	Average	4
Retrieve minor issue	Average	4
Retrieve rides	Average	4
Total		28

2.1.5 External Outputs (EOs)

This Transactional Function point gives the user the ability to produce outputs. The results displayed are derived using data that is maintained and data that is referenced. In function point terminology the resulting display is called an External Output (EO). In accordance to the RASD documents, we have identified these points:

EO	Complexity	FPs
Send password to user	Low	4
Sign up confirmation	Low	4
Show current reservation time	Low	4
Show reservation confirmation	Low	4
Show last rides' invoices	Average	5
Show canceling reservation	Low	4
Show navigation tips	Average	5
Show cars located nearby or by specifying an address	Average	5
Show money saving option navigation tips	Average	5
Show almost empty battery notification	Low	4
Show non-uniform cars' distribution notification	High	7
Show technical issue notification	High	7
Show driver recognition confirmation	Low	4
Show engine deactivation notification	Low	4
Show 8 hours' driving time exceeded notification	Low	4
Total		70

2.2 Overall estimation

Here we present the summation of the total FP calculated in the previous steps:

Function Type	Value
Internal Logic Files	85
External Interface Files	34
External Inputs	42
External Inquiries	28
External Outputs	70
Total	259

With regard to the FP method, the total value is correlated to the expected effort to be put into the deployment of the system. We use the constant values reported in http://www.qsm.com/resources/function-point-languages-table MODIFICA for calculating the lower and upper bounds in the number of lines of code deployed in J2EE language: in particular, we chose the *average* multiplier value (46 for J2EE) for calculating the lower bound and the *high* multiplier (67 for J2EE) for calculating the upper bound.
Depending on the conversion rate, we have a lower bound of:

$$\text{SLOC} = 259 * 46 = 11914$$

and an upper bound of

$$\text{SLOC} = 259 * 67 = 19684$$

3 COCOMO II analysis

3.1 Software scale drivers analysis

In the following lines we present a brief description of the software scale drivers taken into consideration by the COCOMO II software, with:

- **Precedentedness (PREC)**: it reflects the previous experience of the team with the development of projects similar to the current one. Since we are not expert in the field, we chose to set the value to *very low*.
- **Development flexibility (FLEX)**: it reflects the degree of flexibility in the development process with respect to the external specification and requirements. Since there are very strict requirements on the functionalities but we were left with relative freedom in the choice of the technology to be used, this value is set to *low*.
- **Risk resolution (RESL)**: it reflects the level of awareness and reactiveness with respect to risks threatening the system. The risk analysis we performed covered the majority of possible bounds, so the value is set to *very high*.
- **Team cohesion (TEAM)**: it's an indicator of how well the team members work together in a cooperative way. For our team, the value is set to *very high*.
- **Process maturity (PMAT)**: the level reached in the creation of the system. On the 18 values scale, we reached the 11th level, so the value is set to *high*, coherently with the prescriptions of CSSE.

3.2 Software cost drivers analysis

In the following lines we present a brief description of the software cost drivers taken into consideration by the COCOMO II software. The cost drivers were evaluated using a post-architecture perspective to obtain an accurate analysis of the costs implied by the project. Cost drivers can be divided into four different subsections: product factors, platform factors, personnel factors and project factors.

3.2.1 Product factors analysis

- **Required software reliability (RELY)**: it represents the measure of the extent to which the software must perform its function over a period of time. We chose to set the value to *low*.
- **Data base size (DATA)**: it represents the effective size of our database. We estimated the fraction D/P and the result was around the value of 11, so we chose to set the value to *nominal*.
- **Product complexity (CPLX)**: it is the evaluated complexity of the produced software. The value is set to *nominal*, due to a complete analysis of device-dependent operations, data management operations, and user interface management operations.
- **Required reusability (RUSE)**: it represents the possibility to reuse the components in different environment. The value is set to *nominal*.
- **Documentation match to life-cycle needs (DOCU)**: it represents the suitability of the project's documentation to its life-cycle needs. This value is set to *nominal*, following the prescriptions of CSSE.

3.2.2 Platform factors analysis

- **Time constraint (TIME)**: it is a measure of the execution time constraint imposed upon a software system. This value is set to *high*, since we plan to use 70% of the execution time resource.
- **Main storage constraint (STOR)**: it is a measure of the degree of main storage constraint imposed on a software system or subsystem. This value is set to *nominal*, since we plan to use up to 50% of the main storage resources.
- **Platform volatility (PVOL)**: it is a measure of the amount of time that should pass before minor/major changes. This value is set to *low*, since we plan to make minor changes every month and major changes every year once the software is fully deployed.

3.2.3 Personnel factors analysis

- **Analyst capability (ACAP)**: it represents the Analysis and Design ability, efficiency and thoroughness, and the ability to communicate and cooperate of the analysts. This value is set to *high*.
- **Programmer capability (PCAP)**: it represents the ability of the programmers employed in the project. This value is set to *high*.
- **Personnel continuity (PCON)**: it represents the percentage of personnel that does the turnover. This value is set to *very high*.
- **Applications experience (AEXP)**: it represents the previous experience of the project team in development on the same applications used in the current software. This value is set to *very low* (2 months previous experience).
- **Platform experience (PEXP)**: it represents the prevision for the necessity of a post-architecture. This value is set to *nominal* (1 year).
- **Language and tool experience (LTEX)**: it represents the previous experience of the project team in development in the same language used in the current project. This value is set to *very low*.

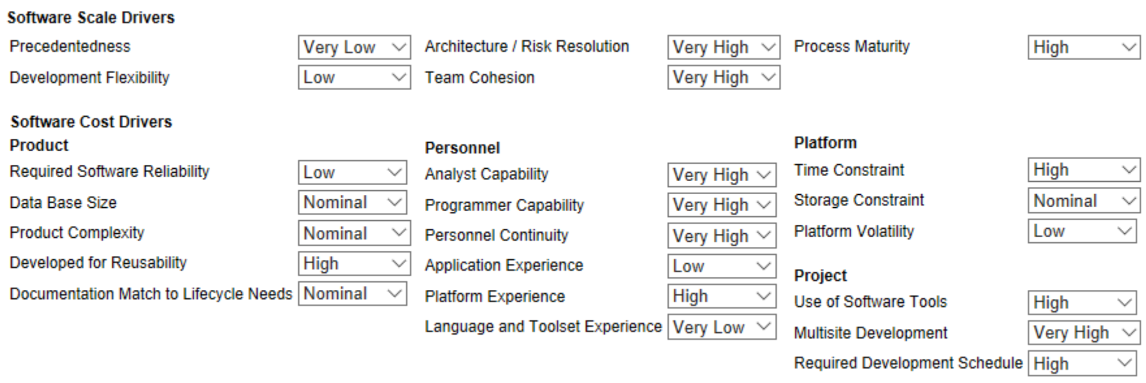
3.2.4 Project factors analysis

- **Use of software tools (TOOL)**: it represents the level of maturity of the software used. This value is set to *high*.
- **Multisite development (SITE)**: it represents the level of the communication when the team worked in separated location. This value is set to *extra-high* (interactive multimedia used).
- **Requirement development schedule (SCED)**: it represents the schedule constraint imposed on the project team developing the software. This value is set to *high*.

4 Effort estimation

We represent below the result of the analysis operated by the official online tool of COCOMO II. The results presented refers to the upper and lower bounds pointed out in the first chapter and to the drivers' values set in the previous chapter. For computing the estimated cost of the project we referred to an average wage of 3000 dollars per month for every employee.

4.1 Screen view



Software Scale Drivers

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

Software Cost Drivers

Product

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

Personnel

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

Platform

Time Constraint	High
Storage Constraint	Nominal
Platform Volatility	Low

Project

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

4.2 Effort (average) estimation

Results

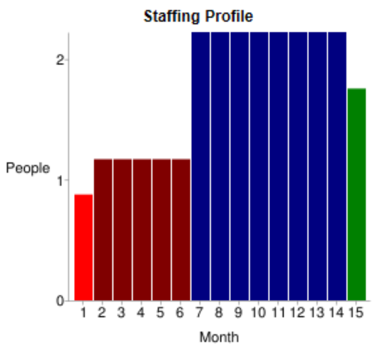
Software Development (Elaboration and Construction)

Effort = 25.3 Person-months
Schedule = 13.9 Months
Cost = $76007

Total Equivalent Size = 11914 SLOC

Acquisition Phase Distribution

Phase	Effort (Person-months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	1.5	1.7	0.9	$4560
Elaboration	6.1	5.2	1.2	$18242
Construction	19.3	8.7	2.2	$57766
Transition	3.0	1.7	1.8	$9121



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

Phase/Activity	Inception	Elaboration	Construction	Transition
Management	0.2	0.7	1.9	0.4
Environment/CM	0.2	0.5	1.0	0.2
Requirements	0.6	1.1	1.5	0.1
Design	0.3	2.2	3.1	0.1
Implementation	0.1	0.8	6.5	0.6
Assessment	0.1	0.6	4.6	0.7
Deployment	0.0	0.2	0.6	0.9

4.3 Effort (over) estimation

Results

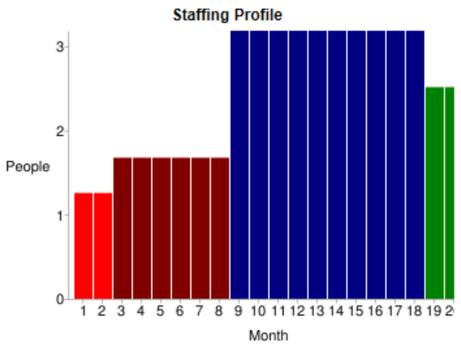
Software Development (Elaboration and Construction)

Effort = 43.3 Person-months
Schedule = 16.5 Months
Cost = $129991

Total Equivalent Size = 19684 SLOC

Acquisition Phase Distribution

Phase	Effort (Person-months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	2.6	2.1	1.3	$7800
Elaboration	10.4	6.2	1.7	$31198
Construction	32.9	10.3	3.2	$98794
Transition	5.2	2.1	2.5	$15599



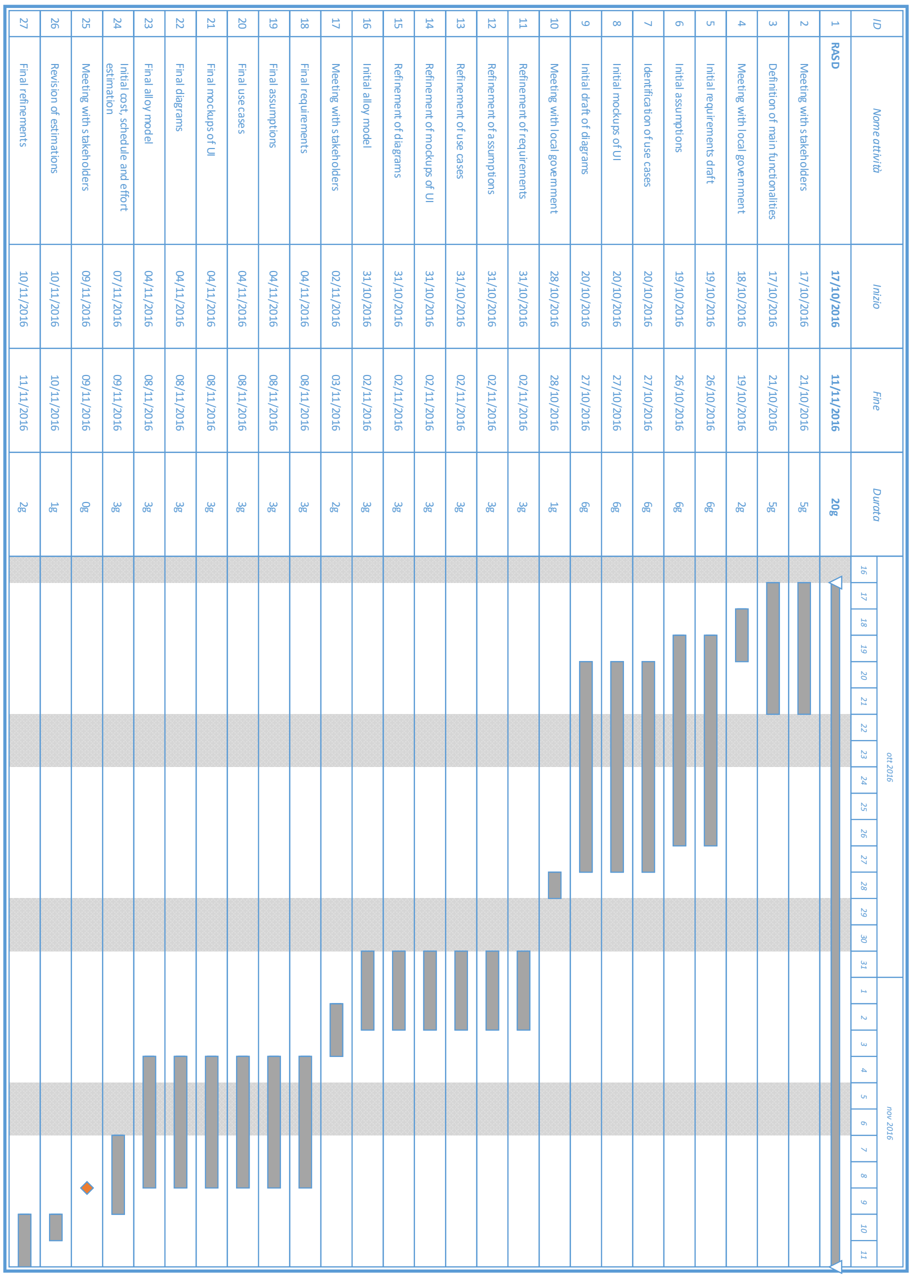
Software Effort Distribution for RUP/MBASE (Person-Months)

Phase/Activity	Inception	Elaboration	Construction	Transition
Management	0.4	1.2	3.3	0.7
Environment/CM	0.3	0.8	1.6	0.3
Requirements	1.0	1.9	2.6	0.2
Design	0.5	3.7	5.3	0.2
Implementation	0.2	1.4	11.2	1.0
Assessment	0.2	1.0	7.9	1.2
Deployment	0.1	0.3	1.0	1.6

5 Scheduling

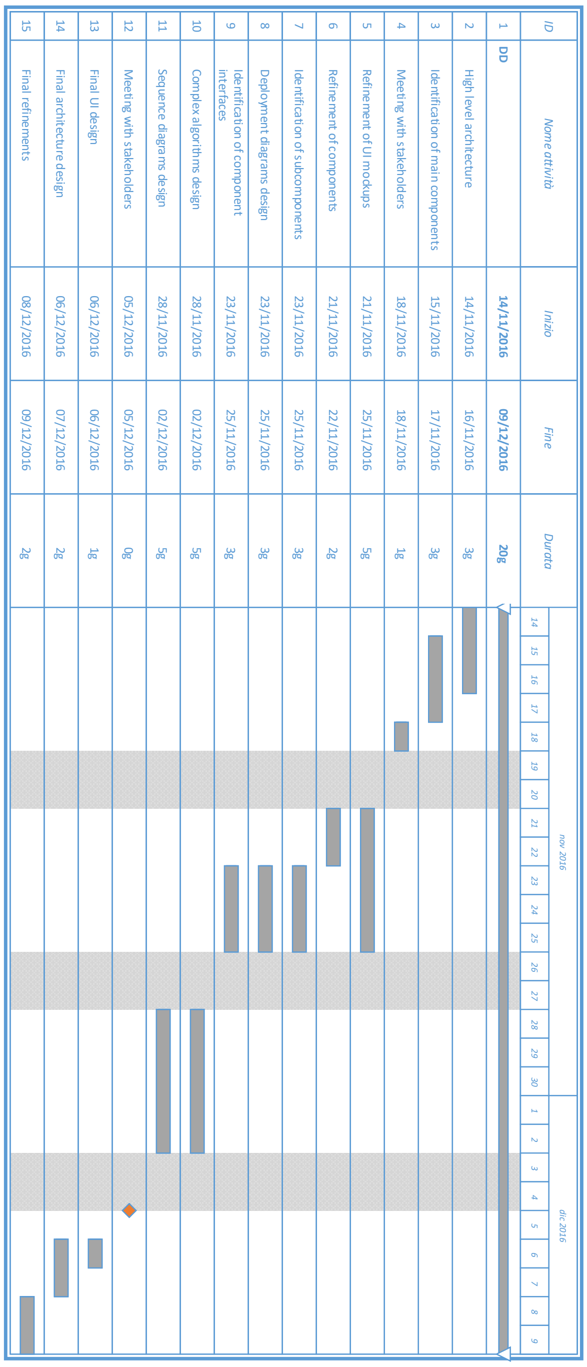
In this chapter provide a general, high-level project schedule. More accurate schedules will be defined during the project development to manage the internal organization of the single development phases, referring to the average evaluation made in the previous chapter. It is important to notice that, while this project is made for didactic purposes and no implementation, testing, deployment and start-up will be performed, we have nevertheless considered these steps as part of our schedule. This was made to try to take into account what could be the full development of this project, should it be entirely completed.
As COCOMO II modeling pointed out that for our project the lower bound work effort's prevision is worth 25.3 person-months. Since we are a three people group, we would need 25.3 / 3 = 8.43 months for developing our project entirely. We assumed to start our project (as it really happened) on October 17th and predicted to complete it within July 7th.
We used Gantt diagrams to highlight tasks duration, useful to schedule, coordinate and track each task providing a clear illustration of project's progress. In order to maintain readability, we have split the schedule in four main parts, each related to its main milestones. The first covers a time period which goes from October 17th to November 11th and concerns the RASD as actually happened. The second time period goes from November 13th to December 9th and concerns the DD as actually happened. The third is about the development phase and it is extended in a time period which goes from December 12th to June 2nd. The last is composed of two phases: deployment (June 5th – June 23rd) and start-up (June 26th – July 7th). All of these dates are designed considering that all the three members would have worked eight hour per day, five days per week.

5.1 RASD



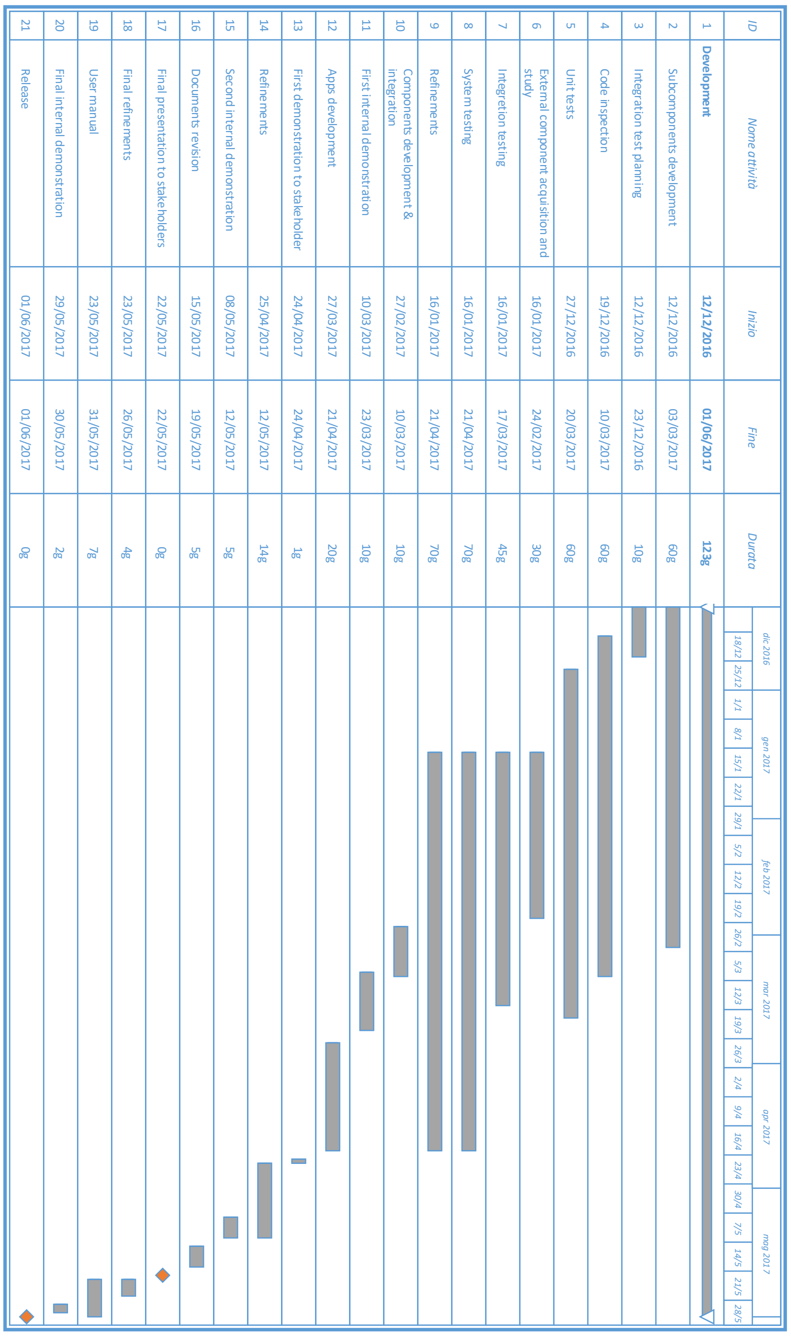
ID	Nome attività	Inizio	Fine	Durata
1	**RASD**	**17/10/2016**	**11/11/2016**	**20g**
2	Meeting with stakeholders	17/10/2016	21/10/2016	5g
3	Definition of main functionalities	17/10/2016	21/10/2016	5g
4	Meeting with local government	18/10/2016	19/10/2016	2g
5	Initial requirements draft	19/10/2016	26/10/2016	6g
6	Initial assumptions	19/10/2016	26/10/2016	6g
7	Identification of use cases	20/10/2016	27/10/2016	6g
8	Initial mockups of UI	20/10/2016	27/10/2016	6g
9	Initial draft of diagrams	20/10/2016	27/10/2016	6g
10	Meeting with local government	28/10/2016	28/10/2016	1g
11	Refinement of requirements	31/10/2016	02/11/2016	3g
12	Refinement of assumptions	31/10/2016	02/11/2016	3g
13	Refinement of use cases	31/10/2016	02/11/2016	3g
14	Refinement of mockups of UI	31/10/2016	02/11/2016	3g
15	Refinement of diagrams	31/10/2016	02/11/2016	3g
16	Initial alloy model	31/10/2016	02/11/2016	3g
17	Meeting with stakeholders	02/11/2016	03/11/2016	2g
18	Final requirements	04/11/2016	08/11/2016	3g
19	Final assumptions	04/11/2016	08/11/2016	3g
20	Final use cases	04/11/2016	08/11/2016	3g
21	Final mockups of UI	04/11/2016	08/11/2016	3g
22	Final diagrams	04/11/2016	08/11/2016	3g
23	Final alloy model	04/11/2016	08/11/2016	3g
24	Initial cost, schedule and effort estimation	07/11/2016	09/11/2016	3g
25	Meeting with stakeholders	09/11/2016	09/11/2016	0g
26	Revision of estimations	10/11/2016	10/11/2016	1g
27	Final refinements	10/11/2016	11/11/2016	2g

5.2 DD



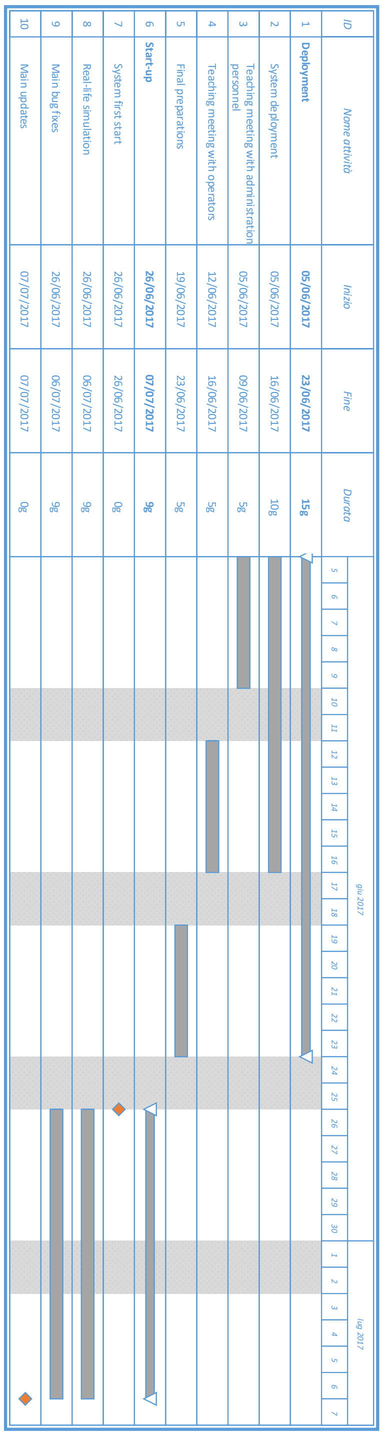
ID	Nome attività	Inizio	Fine	Durata
1	**DD**	**14/11/2016**	**09/12/2016**	**20g**
2	High level architecture	14/11/2016	16/11/2016	3g
3	Identification of main components	15/11/2016	17/11/2016	3g
4	Meeting with stakeholders	18/11/2016	18/11/2016	1g
5	Refinement of UI mockups	21/11/2016	25/11/2016	5g
6	Refinement of components	21/11/2016	22/11/2016	2g
7	Identification of subcomponents	23/11/2016	25/11/2016	3g
8	Deployment diagrams design	23/11/2016	25/11/2016	3g
9	Identification of component interfaces	23/11/2016	25/11/2016	3g
10	Complex algorithms design	28/11/2016	02/12/2016	5g
11	Sequence diagrams design	28/11/2016	02/12/2016	5g
12	Meeting with stakeholders	05/12/2016	05/12/2016	0g
13	Final UI design	06/12/2016	06/12/2016	1g
14	Final architecture design	06/12/2016	07/12/2016	2g
15	Final refinements	08/12/2016	09/12/2016	2g

5.3 Development



ID	*Nome attività*	*Inizio*	*Fine*	*Durata*
1	**Development**	**12/12/2016**	**01/06/2017**	**123g**
2	Subcomponents development	12/12/2016	03/03/2017	60g
3	Integration test planning	12/12/2016	23/12/2016	10g
4	Code inspection	19/12/2016	10/03/2017	60g
5	Unit tests	27/12/2016	20/03/2017	60g
6	External component acquisition and study	16/01/2017	24/02/2017	30g
7	Integretion testing	16/01/2017	17/03/2017	45g
8	System testing	16/01/2017	21/04/2017	70g
9	Refinements	16/01/2017	21/04/2017	70g
10	Components development & integration	27/02/2017	10/03/2017	10g
11	First internal demonstration	10/03/2017	23/03/2017	10g
12	Apps development	27/03/2017	21/04/2017	20g
13	First demonstration to stakeholder	24/04/2017	24/04/2017	1g
14	Refinements	25/04/2017	12/05/2017	14g
15	Second internal demonstration	08/05/2017	12/05/2017	5g
16	Documents revision	15/05/2017	19/05/2017	5g
17	Final presentation to stakeholders	22/05/2017	22/05/2017	0g
18	Final refinements	23/05/2017	26/05/2017	4g
19	User manual	23/05/2017	31/05/2017	7g
20	Final internal demonstration	29/05/2017	30/05/2017	2g
21	Release	01/06/2017	01/06/2017	0g

5.4 Deployment and Start Up

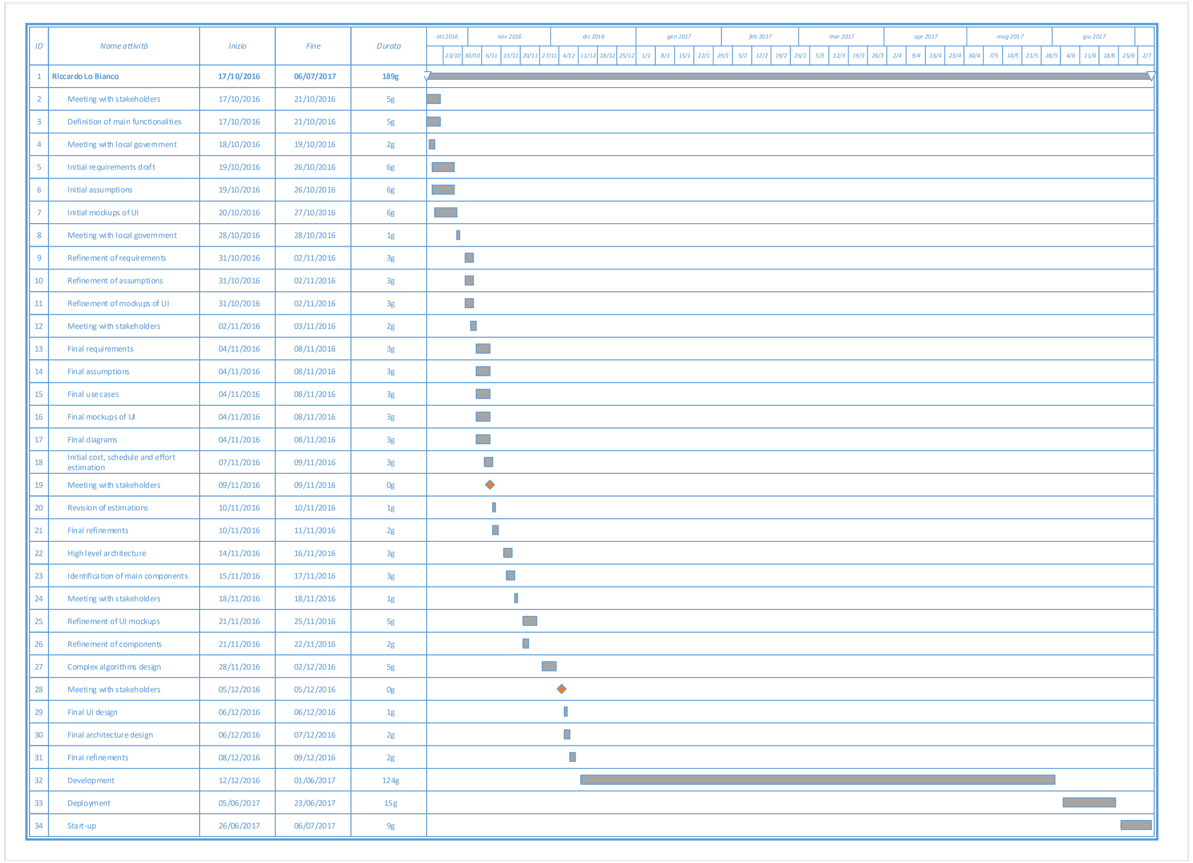


ID	Nome attività	Inizio	Fine	Durata
1	**Deployment**	**05/06/2017**	**23/06/2017**	**15g**
2	System deployment	05/06/2017	16/06/2017	10g
3	Teaching meeting with administration personnel	05/06/2017	09/06/2017	5g
4	Teaching meeting with operators	12/06/2017	16/06/2017	5g
5	Final preparations	19/06/2017	23/06/2017	5g
6	**Start-up**	**26/06/2017**	**07/07/2017**	**9g**
7	System first start	26/06/2017	26/06/2017	0g
8	Real-life simulation	26/06/2017	06/07/2017	9g
9	Main bug fixes	26/06/2017	06/07/2017	9g
10	Main updates	07/07/2017	07/07/2017	0g

6 Resource allocation

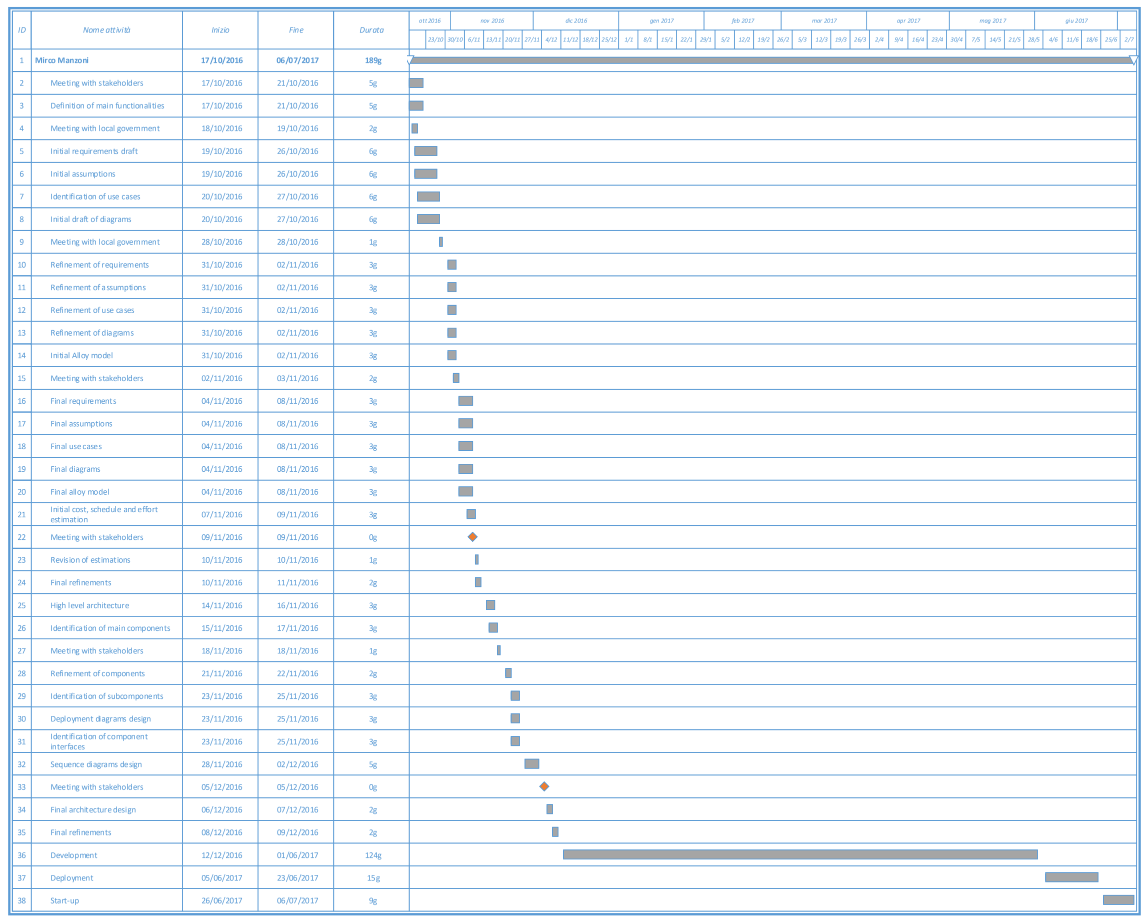
In this chapter we're going to provide a general overview of how the tasks defined in the schedule in the previous section will be divided among the three members of the development team. More accurate schedules will be defined during the project development itself to manage the internal organization of each development phase.
As we already mentioned in the previous section, we have also included activities in the requirement analysis and design phases that won't actually take place, like the stakeholder's meetings, as well as the implementation, the deployment and the start-up phases. This has been purposefully done to have a more realistic depiction of how the development process should actually be brought on.
In order to maintain a good level of readability, we have split the document in three parts, one for each team member.

6.1 Resource allocation - Lo Bianco Riccardo



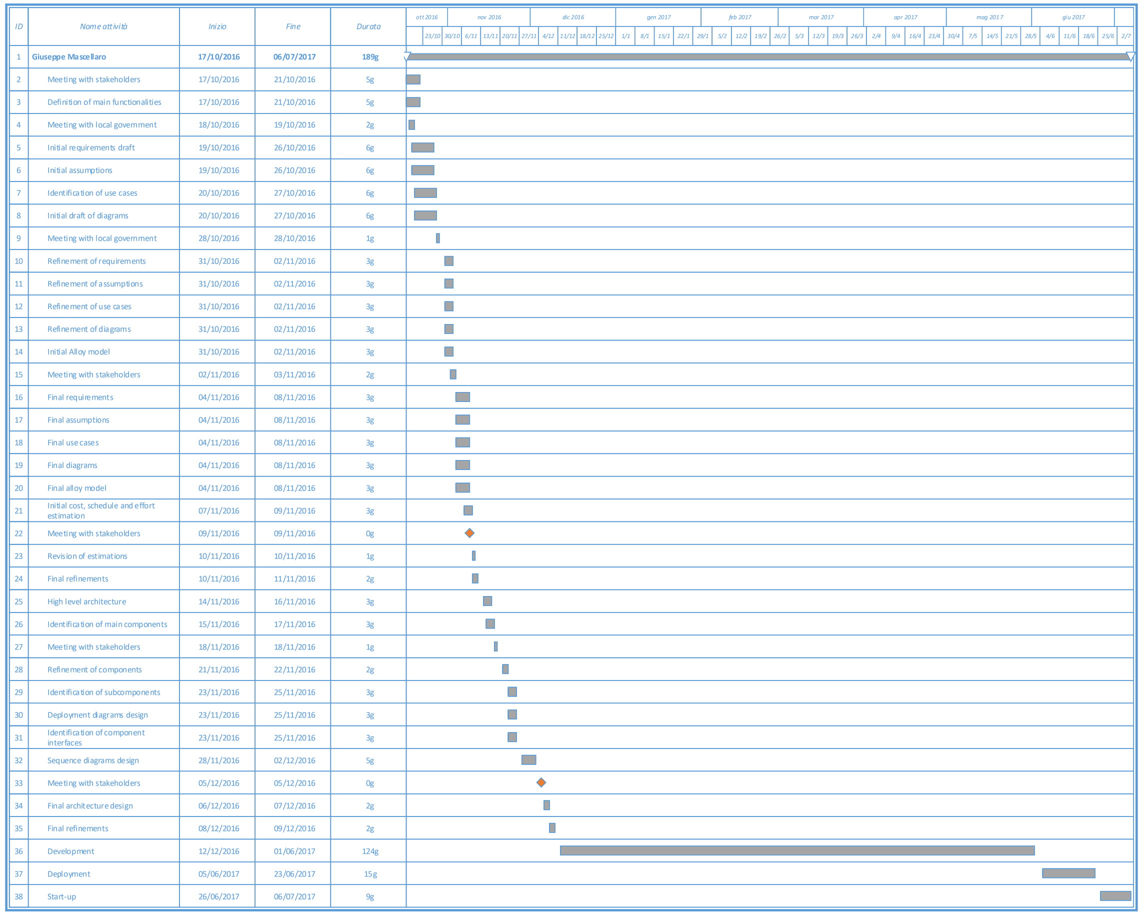
ID	Nome attività	Inizio	Fine	Durata
1	**Riccardo Lo Bianco**	**17/10/2016**	**06/07/2017**	**189g**
2	Meeting with stakeholders	17/10/2016	21/10/2016	5g
3	Definition of main functionalities	17/10/2016	21/10/2016	5g
4	Meeting with local government	18/10/2016	19/10/2016	2g
5	Initial requirements draft	19/10/2016	26/10/2016	6g
6	Initial assumptions	19/10/2016	26/10/2016	6g
7	Initial mockups of UI	20/10/2016	27/10/2016	6g
8	Meeting with local government	28/10/2016	28/10/2016	1g
9	Refinement of requirements	31/10/2016	02/11/2016	3g
10	Refinement of assumptions	31/10/2016	02/11/2016	3g
11	Refinement of mockups of UI	31/10/2016	02/11/2016	3g
12	Meeting with stakeholders	02/11/2016	03/11/2016	2g
13	Final requirements	04/11/2016	08/11/2016	3g
14	Final assumptions	04/11/2016	08/11/2016	3g
15	Final use cases	04/11/2016	08/11/2016	3g
16	Final mockups of UI	04/11/2016	08/11/2016	3g
17	Final diagrams	04/11/2016	08/11/2016	3g
18	Initial cost, schedule and effort estimation	07/11/2016	09/11/2016	3g
19	Meeting with stakeholders	09/11/2016	09/11/2016	0g
20	Revision of estimations	10/11/2016	10/11/2016	1g
21	Final refinements	10/11/2016	11/11/2016	2g
22	High level architecture	14/11/2016	16/11/2016	3g
23	Identification of main components	15/11/2016	17/11/2016	3g
24	Meeting with stakeholders	18/11/2016	18/11/2016	1g
25	Refinement of UI mockups	21/11/2016	25/11/2016	5g
26	Refinement of components	21/11/2016	22/11/2016	2g
27	Complex algorithms design	28/11/2016	02/12/2016	5g
28	Meeting with stakeholders	05/12/2016	05/12/2016	0g
29	Final UI design	06/12/2016	06/12/2016	1g
30	Final architecture design	06/12/2016	07/12/2016	2g
31	Final refinements	08/12/2016	09/12/2016	2g
32	Development	12/12/2016	01/06/2017	124g
33	Deployment	05/06/2017	23/06/2017	15g
34	Start-up	26/06/2017	06/07/2017	9g

6.2 Resource allocation - Manzoni Mirco



ID	Nome attività	Inizio	Fine	Durata
1	**Mirco Manzoni**	**17/10/2016**	**06/07/2017**	**189g**
2	Meeting with stakeholders	17/10/2016	21/10/2016	5g
3	Definition of main functionalities	17/10/2016	21/10/2016	5g
4	Meeting with local government	18/10/2016	19/10/2016	2g
5	Initial requirements draft	19/10/2016	26/10/2016	6g
6	Initial assumptions	19/10/2016	26/10/2016	6g
7	Identification of use cases	20/10/2016	27/10/2016	6g
8	Initial draft of diagrams	20/10/2016	27/10/2016	6g
9	Meeting with local government	28/10/2016	28/10/2016	1g
10	Refinement of requirements	31/10/2016	02/11/2016	3g
11	Refinement of assumptions	31/10/2016	02/11/2016	3g
12	Refinement of use cases	31/10/2016	02/11/2016	3g
13	Refinement of diagrams	31/10/2016	02/11/2016	3g
14	Initial Alloy model	31/10/2016	02/11/2016	3g
15	Meeting with stakeholders	02/11/2016	03/11/2016	2g
16	Final requirements	04/11/2016	08/11/2016	3g
17	Final assumptions	04/11/2016	08/11/2016	3g
18	Final use cases	04/11/2016	08/11/2016	3g
19	Final diagrams	04/11/2016	08/11/2016	3g
20	Final alloy model	04/11/2016	08/11/2016	3g
21	Initial cost, schedule and effort estimation	07/11/2016	09/11/2016	3g
22	Meeting with stakeholders	09/11/2016	09/11/2016	0g
23	Revision of estimations	10/11/2016	10/11/2016	1g
24	Final refinements	10/11/2016	11/11/2016	2g
25	High level architecture	14/11/2016	16/11/2016	3g
26	Identification of main components	15/11/2016	17/11/2016	3g
27	Meeting with stakeholders	18/11/2016	18/11/2016	1g
28	Refinement of components	21/11/2016	22/11/2016	2g
29	Identification of subcomponents	23/11/2016	25/11/2016	3g
30	Deployment diagrams design	23/11/2016	25/11/2016	3g
31	Identification of component interfaces	23/11/2016	25/11/2016	3g
32	Sequence diagrams design	28/11/2016	02/12/2016	5g
33	Meeting with stakeholders	05/12/2016	05/12/2016	0g
34	Final architecture design	06/12/2016	07/12/2016	2g
35	Final refinements	08/12/2016	09/12/2016	2g
36	Development	12/12/2016	01/06/2017	124g
37	Deployment	05/06/2017	23/06/2017	15g
38	Start-up	26/06/2017	06/07/2017	9g

6.3 Resource allocation - Mascellaro Giuseppe



ID	Nome attività	Inizio	Fine	Durata
1	**Giuseppe Mascellaro**	**17/10/2016**	**06/07/2017**	**189g**
2	Meeting with stakeholders	17/10/2016	21/10/2016	5g
3	Definition of main functionalities	17/10/2016	21/10/2016	5g
4	Meeting with local government	18/10/2016	19/10/2016	2g
5	Initial requirements draft	19/10/2016	26/10/2016	6g
6	Initial assumptions	19/10/2016	26/10/2016	6g
7	Identification of use cases	20/10/2016	27/10/2016	6g
8	Initial draft of diagrams	20/10/2016	27/10/2016	6g
9	Meeting with local government	28/10/2016	28/10/2016	1g
10	Refinement of requirements	31/10/2016	02/11/2016	3g
11	Refinement of assumptions	31/10/2016	02/11/2016	3g
12	Refinement of use cases	31/10/2016	02/11/2016	3g
13	Refinement of diagrams	31/10/2016	02/11/2016	3g
14	Initial Alloy model	31/10/2016	02/11/2016	3g
15	Meeting with stakeholders	02/11/2016	03/11/2016	2g
16	Final requirements	04/11/2016	08/11/2016	3g
17	Final assumptions	04/11/2016	08/11/2016	3g
18	Final use cases	04/11/2016	08/11/2016	3g
19	Final diagrams	04/11/2016	08/11/2016	3g
20	Final alloy model	04/11/2016	08/11/2016	3g
21	Initial cost, schedule and effort estimation	07/11/2016	09/11/2016	3g
22	Meeting with stakeholders	09/11/2016	09/11/2016	0g
23	Revision of estimations	10/11/2016	10/11/2016	1g
24	Final refinements	10/11/2016	11/11/2016	2g
25	High level architecture	14/11/2016	16/11/2016	3g
26	Identification of main components	15/11/2016	17/11/2016	3g
27	Meeting with stakeholders	18/11/2016	18/11/2016	1g
28	Refinement of components	21/11/2016	22/11/2016	2g
29	Identification of subcomponents	23/11/2016	25/11/2016	3g
30	Deployment diagrams design	23/11/2016	25/11/2016	3g
31	Identification of component interfaces	23/11/2016	25/11/2016	3g
32	Sequence diagrams design	28/11/2016	02/12/2016	5g
33	Meeting with stakeholders	05/12/2016	05/12/2016	0g
34	Final architecture design	06/12/2016	07/12/2016	2g
35	Final refinements	08/12/2016	09/12/2016	2g
36	Development	12/12/2016	01/06/2017	124g
37	Deployment	05/06/2017	23/06/2017	15g
38	Start-up	26/06/2017	06/07/2017	9g

7 Risk analysis

Through this chapter we take into account the risks the team may face through the development of the project. These risks may come from different sources, such as political and economical factors and environmental needs, but also related to the possible technical issues that may happen.
One possible risk that must be taken into account is the adversion of pre-existing companies that are directly or indirectly in competition with the services offered by PowerEnJoy, such as the "classical" car sharing services and the taxi drivers association. Of course these categories will try to prevent PowerEnJoy from entering the market by pushing on legal factors.
Another issue concerning this field is the fact that the city administration is one of the main stakeholders in a project like PowerEnJoy but, being a public entity, it must undergo strict dictates from the high level public authority.
These problems brings up the necessity of a phase of accurate research in the field of law as a starting point for the entire project, as well as the integration of the stakeholders as active elements in the management of the project along all its steps. In particular the times of the project plan must be modeled in order to take into account the need of several revisions with the stakeholders themselves. The process of legal revision of the project must be iterative and it should take place at least once every six months, plus every time a big change in national laws takes place (of course, in advance with the approval of the changes).
The environmental issues can concern the dismission of exhaust batteries and broken pieces, which requires a special contract for the dismission in stocks. Consulting an expert in this field is fundamental in the first phase of the project, and all the operators must undergo a special training provided by the company in order to be well aware of their role and tasks.
For what concerns the risks that may arise in the development phase, we must consider that the evaluation regardind the ability of our programmers with the language and the tools used may not match the previsions and lead to delay in the date of the deployment. Too face this eventuality, the deadlines we planned are highly flexible and incremental, so that eventual issues can be faced with recurrent analysis of the code and with minor changes to the planning. It must be noticed that, regardless of the isssues we may encounter during the coding phase, no other programmers will be hired. This is a common behaviour when developing huge projects, since the time necessary to introduce new team members in the environment most of times leads to additional delays.
Possible problems regarding the loss of parts of code due to issues in communication are handled by defining a strict schedule of backups, that the programmers must follow no matter what.
Problems can also arise during the design of the interactions with the external elements, such as APIs. This problem can be partially overcome trying to develop a code as portable as possible, so that eventual changes in the contracts with the partners can be handled in a not disruptive way, just by modifying a few lines of code, thus exploiting the information hiding principle to the fullest.

8 Hours of work

The team divided the work into equivalent parts, even when modeling different parts of the document. In the following table we present a resume of the work division.

Lo Bianco Riccardo	12h
Manzoni Mirco	12h
Mascellaro Giuseppe	12h