Software Engineering 2: PowerEnJoy

Project Plan (PP)

Version 1.0

Politecnico di Milano, A.A. 2016/2017

Agosti Isabella, 874835
Cattivelli Carolina, 879359

January 22, 2017

Contents

1 INTRODUCTION

1.1 Revision History

Version	Date	Author(s)	Summary
1.0	22/01/17	Isabella Agosti, Carolina Cattivelli	Initial release

1.2 Purpose and Scope

The purpose of this document is to define the plan for the PowerEnJoy project, identifying the tasks to be completed, the risks that might occur during its development and the costs of its development. This information can be subsequently used as a guidance to define the required budget, the resources allocation and the schedule of the activities.

1.3 Document organization

The document is organized as follows:

- Section 1, *Introduction*, gives an overview of this document describing its contents, scope etc.
- Section 2, *Project size, cost and effort estimation*, presents an estimate of the expected size of PowerEnJoy in terms of lines of code and of the cost/effort required to actually develop it, based on Function Points and COCOMO approaches.
- Section 3, *Schedule*, presents a possible schedule for the project, covering all activities from the requirements identification to the implementation and testing.
- Section 4, *Resource Allocation*, describes how the tasks will be assigned to each member of the development team.
- Section 5, *Risk Management*, presents the possible risks that PowerEnJoy could face during the development of the project.
- Section 6, *Effort Spent*, includes information on the number of hours each group member has worked towards the fulfillment of this deadline.

1.4 Definitions, Acronyms, Abbreviations

1.4.1 Definitions

Keyword	Definitions
User	A person that interacts with the PowerEnJoy mobile or web application to register to the system.
Registered user	A person who already registered to the system, that interacts with the PowerEnJoy mobile or web application in various ways.
Employee	A member of the PowerEnJoy staff.
Car-sharing service	Model of car rental where people rent cars for short periods of time, often by the hour.
Electric car	Automobile that is propelled by one or more electric motors, using electrical energy stored in rechargeable batteries.
Registration	The act or process of filling out an online form providing credentials and payment information.
Log-in	Process by which a user gains access to the system by identifying and authenticating himself/herself.
Reservation	Arrangement through which a registered user holds a car for his use at a later time.
Safe area	Area whose position is predefined by the management system. Safe areas are the only ones in which a user is allowed to park a car.
Special safe area	Special type of safe area where a car can be recharged.
Discount percentage	Discount applied on the user's last ride only in certain circumstances.
Low battery	The car's battery level is considered "low" when less than 20%.

1.4.2 Acronyms and Abbreviations

Acronym/Abbreviation	Definition
RASD	Requirements Analysis and Specification Document
DD	Design Document
ITPD	Integration Test Plan Document
PP	Project Plan
AA	Anno Accademico (Academic Year)
DB	Database
FP	Function Points
ILF	Internal Logic File
ELF	External Logic File
EI	External Input
EO	External Output

1.5 Reference Documents

- The project description document: *Specifications document: Assignments AA 2016-2017.pdf.*
- The PowerEnJoy Requirements Analysis and Specification Document: *RASD.pdf.*
- The PowerEnJoy Design Document: *DD.pdf.*
- The Integration Test Plan Document: *ITPD.pdf.*
- The Project Planning example: *Project planning example document.pdf.*
- The *COCOMO II Model Definition Manual (version 2.1, 1995 – 2000 Center for Software Engineering, USC).*

2 PROJECT SIZE, COST AND EFFORT ESTIMATION

This section provides some estimations on the expected size, cost and required effort of the PowerEnJoy project.

For the size estimation part we will use the Function Points approach, considering all the main PowerEnJoy functionalities and estimating the correspondent amount of lines of code to be written in Java.

For the cost and effort estimation we will instead rely on the COCOMO approach, using as initial guidance the amount of lines of code computed with the FP approach.

2.1 Size estimation: Function Points

The Function Points approach provides an estimation of a project size taking as inputs the amount of functionalities to be developed and their complexity.

The complexity is evaluated based on the characteristics of the application and described in the following table:

Function Type	***Complexity Weight***		
	Simple	Medium	Complex
Internal Logic Files	7	10	15
External Logic Files	5	7	10
External Inputs	3	4	6
External Outputs	4	5	7
External Inquiries	3	4	6

Figure 1: UFP Complexity Weights

2.1.1 Internal Logic Files (ILFs)

In this paragraph we analyze in details the various ILFs we have identified.

First of all, the system has to store information about users, registered users and employees. These data are condensed in a three-level structure. The first level holds id, first name, surname, username, password, driver license code and province of birth as strings, together with email address, card number and card code as contact information. A secondary table contains the location coordinates (as $< latitude, longitude >$ pairs) necessary to identify the location of the registered user. A third table contains all the reservations related to the registered user.

As for the cars, they are stored using a three-level structure. The first level of the structure holds plate number, brand and model of the car as strings. It also stores the car's status (reserved, broken, blocked, ignited, plugged, with low battery), number of active weight sensors, number of open doors, battery level and number of seats. A secondary table contains the location coordinates (as $< latitude, longitude >$ pairs) necessary to identify the car's location. A third table contains the information related to the car's screen, such as price, total price, hours, minutes and percentage.

Reservations are stored in a dedicated table that holds all the information about the registered user who booked them, the reservation code, the starting and ending time and the reserved car.

Rides are also stored in a dedicated table with a similar structure, the only difference being the absence of the reservation code and the presence of an extra field for the total price.

Finally, the system keeps a list of safe areas, identified by a location and an attribute that indicates whether the safe area is special or not.

ILF	**Complexity**	**FPs**
User	Complex	15
Registered User	Complex	15
Employee	Simple	7
Car	Complex	15
Reservation	Medium	10
Ride	Medium	10
Safe Area	Simple	7
Total		79

2.1.2 External Logic Files (ELFs)

In this paragraph we present the ELFs we have identified, which are homogeneous set of data used by the application but generated by other applications. The only EIF of our system is the interface of a Mapping Service we use for:

- Show the location of all PowerEnJoy cars in a certain geographical region.
- Given an address, get the correspondent pair of coordinates (reverse geocoding).
- Retrieve the graphical representation of the city map to be displayed on the car's screen.

ELF	**Complexity**	**FPs**
Cars location retrieval	Medium	7
Reverse geocoding	Medium	7
Map data retrieval	Medium	7
Total		21

2.1.3 External Inputs (EIs)

PowerEnJoy supports many kind of interactions with different categories of users.

In this paragraph we are going to summarize the impact of the offered features, grouping them by user category.

Users

- **Registration** and **First login**: this operation has a *high* complexity, as it involves a lot of components (Notification Manager, Registration Manager and Account Generator).

Registered User

- **Login/Logout/Manage personal information**: these are *simple* operations that involve only the Account Manager.
- **Make reservation**: this operation has a *high* complexity, as it involves a lot of components (Account Manager, Reservation Manager, Car Manager and Reservation Generator).

- **Report issue**: this operation has a *medium* complexity, as it involves Account Manager, Reservation Manager and Car Manager.
- **Cancel current reservation**: this operation has a *medium* complexity, as it involves Account Manager, Reservation Manager and Car Manager.

Employee

- **Login/Logout**: these are *simple* operations that involve only the Account Manager.
- **Manage car information**: this operation has a *simple* complexity since it involves only the Car Manager.

EI	**Complexity**	**FPs**
Registration and First login	Complex	6
Login	Simple	3
Logout	Simple	3
Manage personal information	Simple	3
Make reservation	Complex	6
Report issue	Medium	4
Cancel current reservation	Medium	4
Manage car information	Simple	3
Total		32

2.1.4 External Inquiries (EQs)

As specified by the FP guidelines, an inquiry is essentially a data retrieval request performed by a user.

PowerEnJoy supports a few interactions of this type that do not require complex computations:

- A registered user can retrieve his/her reservations history.
- A registered user can view his/her profile.
- A registered user can retrieve the active promotions.

EQ	Complexity	FPs
Retrieve reservations history	Simple	3
View profile	Simple	3
Retrieve active promotions	Simple	3
Total		9

2.1.5 External Outputs (EOs)

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

- Notify a user that he/she has received a password via email.
- Notify a registered user that more than ten minutes have passed since he/she parked the car in a non-safe area.
- Notify a registered user that his/her report on a car issue has been received.

EO	Complexity	FPs
Receive password notification	Simple	4
Timeout notification	Simple	4
Report issue notification	Simple	4
Total		12

2.1.6 Overall estimation

The following table summarizes the results of our estimation activity:

Function type	Value
Internal Logic Files	79
External Logic Files	21
External Inputs	32
External Inquiries	9
External Outputs	12
Total	153

Considering Java as the development language, the conversion multiplier from function points to SLOC is 53. We can thus estimate the total number of lines of code:

SLOC = 153 * 53 = 8109

2.2 Cost and effort estimation: COCOMO II

In this paragraph we present the estimation of cost and effort needed to develop the PowerEnJoy application, by using the COCOMO II approach.

2.2.1 Scale Drivers

In order to evaluate the values of the scale drivers, we refer to the following official COCOMO II table:

Scale Factors	**Very Low**	**Low**	**Nominal**	**High**	**Very High**	**Extra High**
PREC SF_j	thoroughly unprecedented 6.20	largely unprecedented 4.96	somewhat unprecedented 3.72	generally familiar 2.48	largely familiar 1.24	thoroughly familiar 0.00
FLEX SF_j	rigorous 5.07	occasional relaxation 4.05	some relaxation 3.04	general conformity 2.03	some conformity 1.01	general goals 0.00
RESL SF_j	little (20%) 7.07	some (40%) 5.65	often (60%) 4.24	generally (75%) 2.83	mostly (90%) 1.41	full (100%) 0.00
TEAM SF_j	very difficult interactions 5.48	some difficult interactions 4.38	basically cooperative interactions 3.29	largely cooperative 2.19	highly cooperative 1.10	seamless interactions 0.00
PMAT SF_j	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

A brief description for each scale driver:

- **Precedentedness**, that reflects the previous experience of our team with the development of this type projects. Since we have some experience in software design but most of the notions required in this project are new to us, the precedentedness value is low.
- **Development Flexibility**, that reflects the degree of flexibility in the development process with respect to the external specifications and requirements. Since there are very strict requirements on the functionalities but nothing specific is stated regarding the technology to be used, this value will be low.
- **Risk Resolution**, that reflects the level of awareness and reactiveness with respect to risks. The risk analysis we performed is quite specific, so the value will be set to high.
- **Team Cohesion**, that is an indicator of how well the team members know each other and work together in a cooperative way. Since the cohesion and communication among the two of us is optimal, the value is very high.
- **Process Maturity**, that is the process maturity of the organization. Although we had some problems during the development of the project, the goals have been successfully achieved. Since this is our first project of this kind, this value is set to level 3.

The result of our evaluation is the following:

Scale Driver	**Factor**	**Value**
Precedentedness (PREC)	Low	4.96
Development flexibility (FLEX)	Low	4.05
Risk resolution (RESL)	High	2.83
Team cohesion (TEAM)	Very high	1.10
Process maturity (PMAT)	Level 3	3.12
Total		16.06

2.2.2 Cost Drivers

Required Software Reliability (RELY)

Since the system represents the only way to reserve electric cars in the city, a malfunctioning could lead to significant financial losses. For this reason, the RELY cost driver is set to high.

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

Data Base Size (DATA)

This measure considers the effective size of our database. We don't have the ultimate answer, but our estimation given the tables and fields we have is to reach a 3MB database. Since it is distributed over 8.000 SLOC, the ratio D/P (measured as testing DB bytes/program SLOC) is 393, resulting in the DATA cost driver being high.

DATA Cost Driver						
DATA Descriptors		$\frac{D}{P} < 10$	$10 \leq \frac{D}{P} < 10^2$	$10^2 \leq \frac{D}{P} < 10^3$	$\frac{D}{P} \geq 10^3$	
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 (CPLX)

Set to nominal according to the COCOMO II rating scale.

CPLX Cost Driver						
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

Developed for Reusability (RUSE)

In our case, the reusability requirements are limited in scope to the project itself, so the RUSE cost driver is set to nominal.

RUSE Cost Driver						
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 (DOCU)

This parameter is evaluated in terms of the suitability of the project's documentation to its life-cycle needs. In our case, every life-cycle need presented in the documentation is covered, so the DOCU cost driver is set to nominal.

DOCU Cost Driver						
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 (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 available execution time expected to be used by the system or subsystem consuming the execution time resource. As PowerEnJoy is a quite complex piece of software, we expect that its CPU usage will be high.

TIME Cost Driver						
TIME Descriptors			$\leq 50\%$ 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

Main Storage Constraint (STOR)

This rating represents the degree of main storage constraint imposed on a software system or subsystem. As current disk drives can easily contain several terabytes of storage, this value is set to nominal.

STOR Cost Driver						
STOR Descriptors			$\leq 50\%$ 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	1.05	1.17	1.46

Platform Volatility (PVOL)

The client applications may require at least a major release once every six months in order to be aligned with the development cycle of the main mobile operating systems. For this reason, this parameter is set to nominal.

PVOL 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 (ACAP)

Analysts are personnel who work on requirements, high-level design and detailed design. We think the analysis of the problem has been conducted in a thorough and complete way with respect to a potential real world implementation. For this reason, this parameter is set to high.

ACAP Cost Driver						
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 evaluation is based on our capability as a team rather than as individuals. Major factors which should be considered in the rating are ability, efficiency and thoroughness, and the ability to communicate and cooperate. We have not implemented the project, so this parameter is just an estimation; however we are fairly in our programming abilities, so we will set this parameter to high.

PCAP Cost Driver						
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

Personnel Continuity (PCON)

This parameter is set to nominal.

PCON Cost Driver						
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	

Application Experience (APEX)

The rating for this cost driver depends on the level of applications experience of the project team developing the software system. We have some experience in the development of Java applications, but we never tackled a Java EE system of this kind. For this reason we are going to set this parameter to low.

APEX Cost Driver						
APEX Descriptors	$\leq$ 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)

We do not have any experience with the Java EE platform, but we have some previous experience with databases, user interfaces and server side development. For this reason, we are going to set this parameter to low.

PLEX Cost Driver						
PLEX Descriptors	$\leq$ 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)

This is a measure of the level of programming language and software tool experience of the project team developing the software system. We do not have any experience with the Java EE language, but we have some previous experience with the development environment, so we are going to set this parameter to low.

LTEX Cost Driver						
LTEX Descriptors	$\leq$ 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

Use of Software Tools (TOOL)

Our application environment is basic and moderately integrated, so we will set this parameter to nominal.

TOOL Cost Driver						
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)

This involves the assessment and judgement-based averaging of two factors: site collocation (from fully collocated to international distribution) and communication support (from surface mail and some phone access to full interactive multimedia). Since we live in the same city and we have collaborated relying hugely on wideband Internet services, we are going to set this parameter to high.

SITE Cost Driver						
SITE Descriptors	International	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 communication	Wideband elect. comm., occasional video conf.	Interactive multimedia
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

Required Development Schedule (SCED)

This rating measures the schedule constraint imposed on the project team developing the software. Although our efforts were well distributed over the available development time, the definition of all the required documentation took a consistent amount of time, especially for the requirement analysis and the design phases. For this reason, this parameter is set to high.

SCED Cost Driver						
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

Overall, our results are expressed by the following table:

Cost Driver	**Factor**	**Value**
Required Software Reliability (RELY)	High	1.10
Data Base Size (DATA)	High	1.14
Product Complexity (CPLX)	Nominal	1.00
Developed for Reusability (RUSE)	Nominal	1.00
Documentation Match to Life-Cycle Needs (DOCU)	Nominal	1.00
Execution Time Constraint (TIME)	High	1.11
Main Storage Constraint (STOR)	Nominal	1.00
Platform Volatility (PVOL)	Nominal	1.00
Analyst Capability (ACAP)	High	0.85
Programmer Capability (PCAP)	High	0.88
Personnel Continuity (PCON)	Nominal	1.00
Application Experience (APEX)	Low	1.10
Platform Experience (PLEX)	Low	1.09
Language and Tool Experience (LTEX)	Low	1.09
Use of Software Tools (TOOL)	Nominal	1.00
Multisite Development (SITE)	High	0.93
Required Development Schedule (SCED)	High	1.00
Total		1.2655

2.2.3 Effort equation

This final equation gives us the effort estimation measured in Person-Months (PM):

$$PM = A * Size^E * \prod_{1 \leq i \leq n} EM_i$$

where:

$A = 2.94$ (This approximates a productivity constant in PM/ KSLOC (Person-Months/Kilo-Source Lines of Code)).
$Size$ = estimated size of the project in KSLOC (it can be deducted from UFP).
EM = Effort Multiplier (the method offers an approach to derive them from Cost Drivers).
$E = B + 0.01 * \sum_{1 \leq j \leq 5} SF_j = 0.91 + 0.01 * 16.06 = 0.91 + 0.1606 = 1.0706$

With this parameters we can compute the effort value:

$$PM = A * Size^E * \prod_{1 \leq i \leq n} EM_i = 2.94 * 8.109^{1.0706} * 1.2655 = 56.957 \approx 57$$

2.2.4 Schedule estimation

Regarding the final schedule, we are going to use the following formula taken from COCOMO II manual:

$$TDEV_{NS} = C * (PM_{NS})^F$$

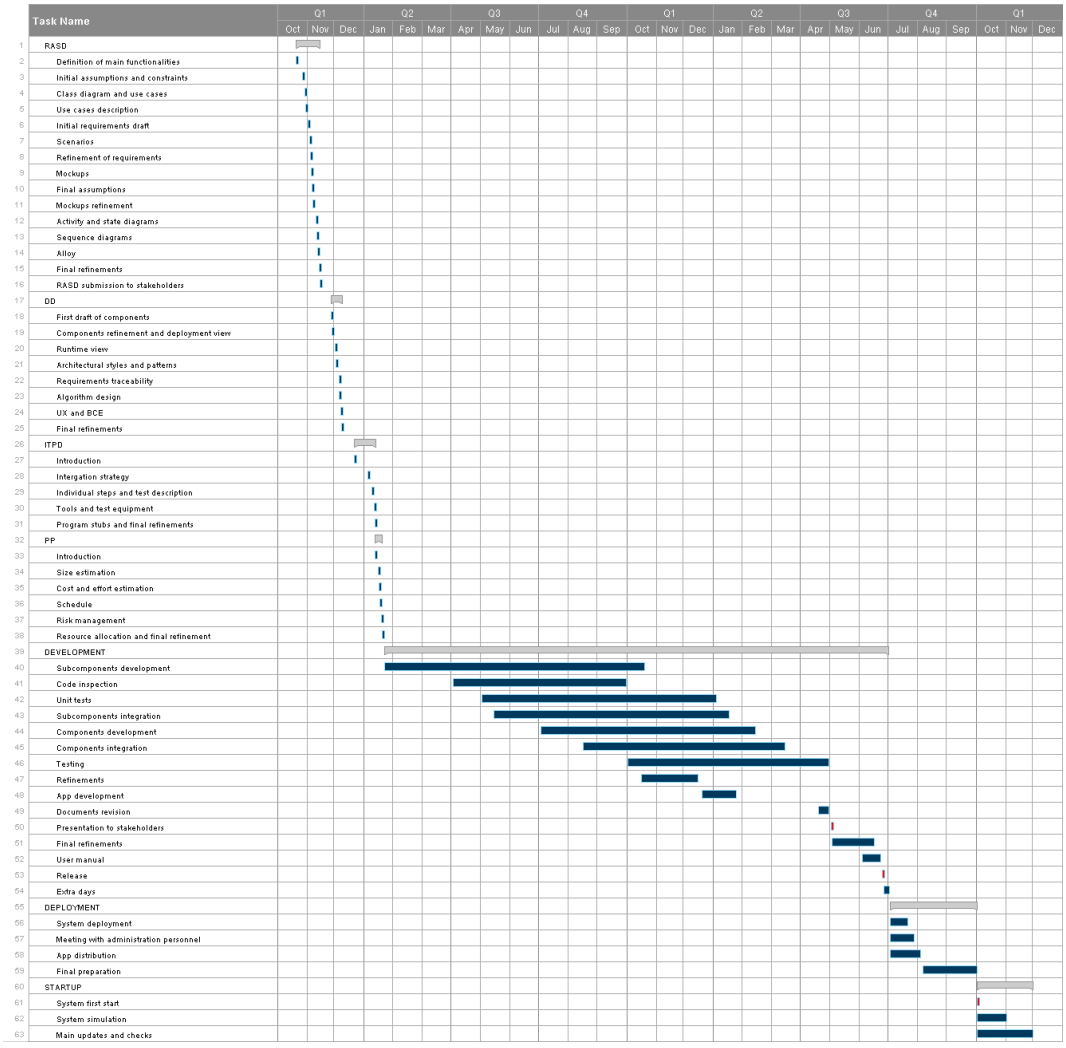
Where:

$F = D + 0.2 * 0.01 * \sum_{j=1}^{5} SF_j = D + 0.2 * (E - B) = 0.28 + 0.2 * (1.0706 - 0.91) = 0.31212$
$PM = 56.957$
$TDEV_{NS} = 3.67 * 56.957^{0.31212} = 12.96$ months

3 SCHEDULE

In this paragraph we provide our project schedule.

It is important to notice that, while this project is made for didactic purposes and no implementation and testing will be performed, we have still considered these steps as part of our schedule to take into account what could be the full development of this project, should it continue.



4 RESOURCE ALLOCATION

In this paragraph we provide a general overview of how the tasks defined in the schedule presented in the previous section will be divided between the two members of the development team.

Each team member contributed equally to every task defined in the schedule. Although this slightly increases the time needed to complete the project, it also allows a more accurate control over the work of each team member.

20/10/2016 ⇒ 13/11/2016			
	1^{st} week	2^{nd} week	3^{rd} week
I. Agosti	Main functionalities	Class diagram, use cases	Activity diagram, state diagram, Alloy
C. Cattivelli	Assumptions, constraints	Scenarios, requirements	Sequence diagrams, Alloy

Table 1: Resource allocation for RASD

26/11/2016 ⇒ 09/12/2016		
	1^{st} week	2^{nd} week
I. Agosti	Components, deployment view, runtime view	Styles and patterns, algorithm, UX
C. Cattivelli	Components, deployment view, runtime view	Requirements traceability, algorithm, BCE

Table 2: Resource allocation for DD

20/10/2016 ⇒ 13/11/2016			
	1^{st} week	2^{nd} week	3^{rd} week
I. Agosti	Introduction	Integration strategy	Tests, tools and test equipment, stubs
C. Cattivelli	Introduction	Integration strategy	Tests, tools and test equipment, stubs

Table 3: Resource allocation for ITPD

26/11/2016 ⇒ 09/12/2016	
	1^{st} week
I. Agosti	Introduction, cost and effort estimation, risk management
C. Cattivelli	Size estimation, schedule, resource allocation

Table 4: Resource allocation for PP

23/01/2017 ⇒ 30/09/2017			
	1^{st} quarter	2^{nd} quarter	3^{rd} quarter
I. Agosti	Subcomponents development	Subcomponents development and integration, code inspection	Subcomponents development and integration, code inspection, components development and integration
C. Cattivelli	Subcomponents development	Subcomponents development and integration, unit tests	Subcomponents development and integration, unit tests, components development and integration

Table 5: Resource allocation for DEVELOPMENT Part 1

01/10/2017 ⇒ 30/06/2018			
	1^{st} quarter	2^{nd} quarter	3^{rd} quarter
I. Agosti	Subcomponents development and integration, components development and integration, testing, refinements, app development	Subcomponents integration, components development and integration, testing	Testing, document revision, final refinements
C. Cattivelli	Subcomponents development and integration, unit tests, components development and integration, refinements, app development	Subcomponents integration, components development and integration, app development	Document revision, final refinements, user manual

Table 6: Resource allocation for DEVELOPMENT Part 2

01/07/2018 ⇒ 30/09/2018			
	1^{st} month	2^{nd} month	3^{rd} month
I. Agosti	System deployment, meeting with administration personnel, app distribution	App distribution, final preparation	Final preparation
C. Cattivelli	System deployment, meeting with administration personnel, app distribution	App distribution, final preparation	Final preparation

Table 7: Resource allocation for DEPLOYMENT

01/10/2018 ⇒ 30/11/2018		
	1^{st} month	2^{nd} month
I. Agosti	System simulation, main updates and checks	Main updates and checks
C. Cattivelli	System simulation, main updates and checks	Main updates and checks

Table 8: Resource allocation for STARTUP

5 RISK MANAGEMENT

In this paragraph we describe the main risks that the project development may face. Some of them represent technical issues, while others are related to political or financial challenges.

5.1 Economical Risks

In this subsection we describe the economical risks PowerEnJoy could face.

Competitors

It may happen that other companies offer better products at a lower price, causing the exclusion of PowerEnJoy from the market.

In order to avoid this risk, PowerEnJoy should frequently improve the quality of its products trying to meet its users' needs.

Lack of fundings

It may happen that the stakeholders are such dissatisfied with the PowerEnJoy service that they decide to cut its fundings.

In order to avoid this risk, the PowerEnJoy company should plan monthly meetings with its stakeholders, to let them express their opinion about the service.

Market risks

It may happen that people are not interested in using our application and prefer traditional methods, causing the company's income to decrease.

A good feasibility study helps avoid this risk.

Changes in regulation

Another issue concerns the possible changes in local and State regulators, that could change the PowerEnJoy regulation at any time.

The only thing we can do in order to avoid this risk is to keep an eye on these laws, which typically take months to be approved, and be ready to move fast before the legislation is actually enacted.

5.2 Project Risks

In this subsection we describe the project risks PowerEnJoy could face.

Project scheduling

Even though an initial overall schedule is provided in this document, it may happen that the project requires more time than expected, due to possible issues that may arise during its development.

In order to avoid this risk, some extra time should been allocated at the end of each major activity to consider the possibility of refinements.

Changes in requirements

It may happen that the client changes his mind about requirements during the development of the project.

This risk cannot be prevented, but it can be mitigated by writing reusable code.

Lack of experience

It may happen that our programmers and engineers' knowledge regarding a specific matter or programming technique is overestimated.

In order to avoid this risk, the company should worry about hiring qualified personnel from the beginning.

Lack of communication

It may happen that the personnel has to work remotely, causing misunderstandings of various kind.

In order to avoid this risk, the company should clearly define the work division among its personnel and have a clear idea regarding the required requirements.

5.3 Technical Risks

In this subsection we describe the technical risks PowerEnJoy could face.

Loss of data

It may happen that the company looses some data due to hardware failure, misconfigured software or external attacks.

In order to avoid this risk, the company should make use of reliable backup techniques distributed over multiple locations, far from the system.

Downtime

It may happen that the system goes down for any reason, such as excessive load, software bugs, hardware failure or power outages.

In order to avoid this risk, the company should build multiple and redundant systems and perform testing at all levels.

Integration testing failure

It may happen that the components do not pass the integration testing phase after their implementation.

In order to avoid this risk, the company should perform an early integration testing making use of stubs and drivers, and define in details the interfaces between components and subsystems.

6 EFFORT SPENT

This section includes information about the number of hours each group member has worked towards the fulfillment of this deadline.

Since we decided to work together every day, the worked hours are going to be the same for each group member. We think this is the best way to achieve good results.

6.1 Agosti Isabella

- 13/01/2017: 1h
- 16/01/2017: 3h
- 17/01/2017: 4h
- 18/01/2017: 2h
- 19/01/2017: 4h
- 20/01/2017: 5h

6.2 Cattivelli Carolina

- 13/01/2017: 1h
- 16/01/2017: 3h
- 17/01/2017: 4h
- 18/01/2017: 2h
- 19/01/2017: 4h
- 20/01/2017: 5h