



POLITECNICO DI MILANO

SOFTWARE ENGINEERING 2

PowerEnjoy Project Plan v1.0

Alessandro Caprarelli
Roberta Iero
Giorgio De Luca

874206
873513
875598

January 18, 2017

Contents

1	Introduction	2
1.1	Revision history	2
1.2	Purpose	2
1.3	Scope	2
1.4	Definition, acronyms, abbreviations	3
1.4.1	Definition	3
1.4.2	Acronyms	4
1.4.3	Abbreviations	4
1.5	Reference documents	4
1.6	Document overview	5
2	Cost estimation	6
2.1	Function point	6
2.1.1	Brief introduction	6
2.1.2	Internal Logic Files	8
2.1.3	External interface files	8
2.1.4	External Inputs	9
2.1.5	External Outputs	9
2.1.6	External Inquiries	10
3	Used tools and working hours	11
3.1	Used tools	11
3.2	Working hours	12

Chapter 1

Introduction

1.1 Revision history

Table 1.1: Revision history

Version	Date
v1.0	22/01/2017

1.2 Purpose

This document represents the Project Plan for the development of the application of Power EnJoy. The aim of this document is to define the approach and the schedule of the project, in order to give a guideline to follow to the project team. The decisions taken in this document take in account the requests from both the stakeholders and the development team.

1.3 Scope

The aim of the project PowerEnjoy is to provide an automated service of car sharing. After a registration, a client can hire a car near him/her through the web or mobile application and he/she can enjoy of all the extra services offered. The exact position of the client is determined by the GPS signal of the client's device or it's allowed to manually insert a specific address. So the system displays all the available cars

in the client's close area. Then the client could make a reservation of a car, after which the system notifies the client with a message of confirmation with the car identifier. If the reservation procedure successfully ends the chosen car won't be available anymore for other clients. Moreover a client cannot hire more than one car at the same time. After the reservation the client has at most one hour to reach the car, when this time expires the system gives a penalty to the client and the car, previously hired, is available again for other clients. The system allows the client to cancel his/her reservation. When the client reaches the car, he/she can tell the system that is nearby through a specific button in the application and he/she starts to pay as soon as the engine ignites. During the travel the system supervises the current charge of the car and notifies it to the client through a screen located in the car. The system stops charging the amount of money that the client has to pay when he/she communicates through the application his/her decision to stop the rent. When the car is parked in a safe area and the client exits, the system locks the car automatically and starts the procedure of payment. The client is notified with the result of this procedure through an SMS, including the final fare.

1.4 Definition, acronyms, abbreviations

1.4.1 Definition

- **Guest client:** a person that is not already registered in the system or that has to log in.
- **Registered client:** a person who has valid access credentials to log in the system.
- **System administrator:** privileged user, in charge of managing administration processes and of updating business logic.
- **Reservation:** it is the action performed by a registered client that allow him/her to reserve an available car for maximum one hour.
- **Journey time = travel time:** time elapsed since the user starts the engine to the user parks the car and terminates the journey.
- **Available car:** a car that is not reserved by any user and has enough charge to be rented.
- **Unavailable car:** a car that is already reserved or damaged, so impossible to reserve.

- **Gps navigation:** it is the navigation system that is included in the car on board system. It could be used by the user to find direction to the final destination.
- **Final destination:** address where the user wants to go.
- **Safe area:** the region where is permitted to park and leave a car once the rent is terminated.
- **Power grid station:** the area where it's allowed users to park the cars, leaving them attached to the power grid.

1.4.2 Acronyms

- **ITP:** Integration Test Plan
- **DD:** Design Document
- **RASD:** Requirements Analysis and Specification Document
- **API:** Application Programming Interface
- **UI:** User Interface
- **DBMS:** Data Base management system
- **COCOMO:** COnstructive COst MOdel
- **SF:** Scale Factor
- **UFP:** Unadjusted Function Point

1.4.3 Abbrevetations

1.5 Reference documents

- RASD v1.1
- DD v1.0
- ITPD v1.0
- PowerEnjoy specification document (assignment).

- IEEE Std 1016tm-2009 Standard for Information Technology - System Design - Software Design Descriptions.

1.6 Document overview

The ITP is composed of five sections:

- Introduction: this section defines the goal of the document and the main characteristics of the project of PowerEnjoy.
- Cost estimation: in this second part there is the cost estimation of the project, done with two different methods: Function Points and COCOMO II. In order to give a greater understanding of the subject, there is also a brief introduction to both the methods used.
- Project Tasks and Schedule: in this section i's provided a schedule of the different phases of the project.
- Resources and Tasks allocation: the fourth section contains the allocation of the resources and the tasks among the components of the team.
- Project Risks: this last section describes the possible risks the project may be exposed to, distinguished by type.

Chapter 2

Cost estimation

2.1 Function point

2.1.1 Brief introduction

The Function Point estimation approach is based on the principle of extracting functions from a software, to classify them using a well defined set of classes and estimating their complexity.

This kind of estimation is extremely useful since it can be done at a very early stage of a project life-cycle, ideally after the implementation of the RASD.

This estimation is a single number called UFP that can be computed using a simple formula.

A high-level procedure that explains how to calculate this number is the following:

1. Classify each function of the software to one of these five possible classes called Function Types (explained in detail later):
 - Internal logic files
 - External logic interfaces files
 - External Inputs
 - External outputs
 - External inquiries
2. For each function a complexity is defined, and it can be:

- Low
- Average
- High

The choice of the complexity is based on the analysis of the quantity of data processed by each function and takes into account also the type of interaction required between different components.

3. Calculate the UFP using the formula:

$$\sum_{f \in F, c \in C} ((\# \text{ of function of type } f \text{ and complexity } c) * (\# \text{ weight for type } f \text{ and complexity } c))$$

where F=ILF, ELF, EI, EO, EIQ and C=Low, Average, High.

Refer to this table to determine the proper weight for each type and complexity:

Table 2.1: UFP Complexity Weights			
Function type	Low	Average	High
Internal Logic files	7	10	15
External Interfaces files	5	7	10
External Inputs	3	4	6
External Ouputs	4	5	7
External Inquiries	3	4	6

Further manipulation of the UFP can be done in order to use it in Cost Estimation Models such as COCOMO, but this will be explained later.

2.1.2 Internal Logic Files

The application manages a database which is used to store different kind of entities, each one of them has its own data structure.

These entities are: registered Client, system administrator, car, safe area, reservation/request, rent, transaction.

Table 2.2: ILF recap

ILF	Complexity	FP
Registered client	Low	7
System administrator	Low	7
Car	Low	7
Safe area	Average	10
Reservation/Request	High	15
Rent	Average	10
Transaction	Average	10
Total:		66

2.1.3 External interface files

The application of Power EnJoy must interact with some external services in order to fulfill its main goals.

The external services are about: driving License, maps, payment gateway, SMS gateway, push notification gateway, customer support.

Table 2.3: EIF recap

ELF	Complexity	FP
Driving License	Low	5
Maps	High	10
Payment gateway	Average	7
SMS gateway	Low	5
Push notification gateay	Low	5
Customer support	Low	5
Total:		37

2.1.4 External Inputs

The application has to handle external interactions due to the Registered Client's actions or to the System Administrator's actions.

These external interactions are: registration, login, logout, modify profile, make a reservation, choose a destination, terminate a rent, manage cars (System Administrator).

Table 2.4: EI recap

EI	Complexity	FP
Registration	Low	3
Login	Low	3
Logout	Low	3
Modify profile	Low	3
Make a reservation	High	6
Chose a destination	Average	4
Terminate a rent	High	6
Cancel a reservation	High	6
Manage cars (SystemAdmin)	Average	4
Total:		38

2.1.5 External Outputs

The application generates the following outputs, result of an internal computation on some data, to the Registered Client or to the System Administrator:

Table 2.5: EO recap

EO	Complexity	FP
Best destination for getting a discount	High	7
List of available cars	Low	4
Notifications	Low	4
Total:		38

2.1.6 External Inquiries

The application generates the following outputs to the Registered Client or to the System Administrator:

Table 2.6: EInq recap

EInq	Complexity	FP
Show Registered Client profile	Low	3
Show System Administrator profile	Low	3
Show history of past rents	Low	3
Show current cost, charge and maps through the OnBoard application	Average	4
Total:		14

Chapter 3

Used tools and working hours

3.1 Used tools

The tools used to create this document are the following:

- MikTeX 2.9: to format the document using LaTeX.
- Visual paradigm: to create diagrams.

3.2 Working hours

	Alessandro	Roberta	Giorgio
29/12	3	3	3
30/12			
31/12			
1/1			
2/1	1	1	2
3/1		1	1
4/1	1	1	
5/1		1	2
6/1	2		
7/1	1	1	1
8/1		1	2
9/1	1		
10/1	1	2	
11/1			2
12/1	3	3	3
13/1	2	2	2
Total	15	16	18