Politecnico di Milano

PowerEnjoy Service - Project Plan Document

January 17, 2017

Version 1.1

Authors:

- Domenico FAVARO (Mat. 837995)

- Matheus FIM (Mat. 876069)

- Caio ZULIANI (Mat. 877266)

Prof. Elisabetta DI NITTO

Contents

1 Introduction

1.1 Revision History

This section records all revisions to the Document.

Version	Date	Authors	Summary
1.1	22/01/16	Domenico Favaro, Caio Zuliani, Matheus Fim	Initial Release

1.2 Purpose and Scope

The Project Plan Document (PPD) is fundamental to organize the production of the System. The main purpose of the Document is to study the complexity of the System, analize the difficulties that can be present in any stage of its development and deployment and give an overall estimation of the effort and cost that can be needed up to the final release of the product. The analysis will be done in terms of size, time and budget needed. This is fundamental to present our stakeholders a preview before the start of production in case they need to do any project changing decision before hand.

The second part of the Document helps the organization of our production by giving time estimations for each step and managing the workload of our resources accordingly. This serves our production team to make any project management decision but also our development team to know how they can organize themselves to follow the schedule and if by any reason the schedule can't be followed, they can notify the production. This is fundamental to keep track of the project developing at every stage of production.

Finally we present possible risks that can arise during the development and for each risk we have identified a contingency plan to avoid or minimize the possibility of harming the production.

1.3 Definitions and Abbreviations

- **RASD:** Reqirements And Specifications Document.

- **DD:** Design Document.

- **ITPD:** Integration Test Plan Document.

- **PPD:** Project Plan Document

- **App:** Application, refering to Web or Mobile App.

- **LOC:** Lines of Code, is a software metric used to measure the size of a computer program by counting the number of lines in the text of the program's source code.

- **FSM:** Functional Size Measurement, technique for measuring software in terms of the functionality it delivers.

- **FP:** Function Points. Unit of measurement used to express the amount of functionality for an information system. We'll be using them to compute a FSM of software.

- **VAF:** Value Adjustment Factor, used to adjust the Function Points estimation to the context of the project.

- **UFP:** Unadjusted Function Points, raw measurement of the total FP before the VAF is applied.

- **ILF:** Internal Logic Files, FP Function Type.

- **EIF:** External Interface Files, FP Function Type.

- **EI:** External Input, FP Function Type.

- **EO:** External Output, FP Function Type.

- **EQ:** External Inquiries, FP Function Type.

- **COCOMO:** Constructive Cost Model is a procedural software cost estimation model that we'll be using for this document.

For other concepts concerning the Service definition look in the **Glossary** section of the RASD, DD and ITPD.

1.4 Reference Documents

- Specification Document: Assignments AA 2016-2017.pdf

- PowerEnjoy Requirements And Specifications Document (RASD)

- PowerEnjoy Design Document (DD)

- PowerEnjoy Integration Test Plan Document (ITPD)

- Example Document - Project planning example document.pdf

- Project Planning Tools Documentation:

    - Function Points Table
        - http://www.qsm.com/resources/function-point-languages-table
    - COCOMO II - Manual
        - COCOMO Model Definition Manual
    - Project Management Software: Jira/Trello
        - https://www.atlassian.com/software/jira
        - https://trello.com/

2   Project Size, Cost and Effort Estimation

This section deals with the estimation of the expected size, cost and effort needed to develop the PowerEnjoy Service. For the size estimation we'll use the Function Points approach, once calculated we'll have an approximated size in LOC, and this measurement will be used later with COCOMO to predict the cost and amount of effort that will be required to develop the program.

2.1   Size Estimation: Function Points

The Function Mode approach is based in giving to each function needed in the code of the program a Category and a Complexity, once this is done, using Function Point Tables, a number of UFP is given to them. The VAF is then applied to the total of UFP. After the Total FP are calculated, it can then be translated into LOC. The Categories for FP are 5, and each has its own rating table for complexity according to the number of Data Elements present:

- Internal Logic Files (ILF):

  Translate to the sets of data used and managed by the application.

- External Interface Files (EIF):

  Similar to ILF set of data used by the application but generated and maintained by other applications.

  Their complexity table is the same as ILF.

Record Types	1-5	6-19	> 19
0-1	Low	Low	Avg
2-3	Low	Avg	High
> 3	Avg	High	High

- External Inputs (EI):

  Defined as operations to receive and manage data coming from the external environment.

File Types	1-4	5-15	>15
0-1	Low	Low	Avg
2	Low	Avg	High
> 2	Avg	High	High

- External Outputs (EO):

  They are elementary operations that generates data for the external environment. It usually includes managing data from logic files

- External Inquiries (EQ):

  Operation that involves both input and output, managing the data themselves without significant intervation from logic files.

  Their complexity table is the same as EO.

File Types	1-5	6-19	>19
0-1	Low	Low	Avg
2-3	Low	Avg	High
> 3	Avg	High	High

To measure the complexity in FP, weights are given to each Function Type using the following table:

Function Type	Low	Avg	High
ILF	7	10	15
EIF	5	7	10
EI	3	4	6
EO	4	5	7
EQ	3	4	6

2.1.1 Internal Logic Files (ILF)

The PowerEnjoy System is organized to store a number of information to provide its functionalities. This information corresponds to the Entities that compose the System and will be manage by the Entity Beans in the Controllers *(see DD)*. The ILF will then correspond to managing the information of these entities, which are:

- Cars: Cars will be stored in the system DB and replicated by persistent entities in the JEE2 environment for managing their status. Their information include CarID, Location, Status, BatteryStatus, if it's PluggedIn and Number of Passengers.

  Car element count is 6 Fields.

- Users: Users tuples will be added by the System in the DB as well for the Registration feature, their attributes include UserID, if it's LoggedIn, UserName, Password, Email, DriverLicense, PaymentInfo and Location.

  User count is 8 Fields.

- CRM: Although not directly added by the System, the loggedIn status of the CRMs is important for the Chat feature. It stores their CRMID, Password, Name, Email and if it's LoggedIn.

  CRM count is 5 Fields.

- Reservations: Will link Users with Cars and will have a status to know wether the reservation is active or not. Their attributes are ReservationID, UserID, CarID, Status, ReservationTime, PickUp-Time.

  Reservation elements count is 6 and references 2 Record Types.

- Rides: Once the Reservation has been confirmed a Ride is created, it refers to the Reservation and it tracks the use of the Car. It has RideID, ReservationID, CarID, Status, StartTime, EndTime, LocationPath.

  Ride count is 7 and references 2 Record Types.

- Payments: When the Ride is finished it generates the Payment for the User to pay, it tracks the input made by the Car, Rides and Users and include also eventual Discounts and Extra Fees. It has PaymentID, RideID, TotalFee, PassengerDiscount, BatteryDiscount, PluggedInDiscount, LowBatteryFee, NoPlugNearFee.

  Payment has 8 fields, references 1 Record Type.

- User Reports: Tracks any fault in the Service by Users or Cars and are handled by CRMs. They have the following fields: UserReportID, UserID, CRMID, CarID, Description.

  User Report has 5 fields, references 3 Record Types.

- Locations: Although not managed by the System, the record of safe parking and plug in areas has to be kept and accessed by the LocationHelper. They have basic Info like LocationID, Latitude, Longitude, Type.

  Locations have 4 fields.

Using the Complexity Table for the ILF we obtain the following.

ILF	Complexity	FP
Car Data	Low	7
User Data	Low	7
CRM Data	Low	7
Reservations	Avg	10
Rides	Avg	10
Payments	Low	7
User Reports	Low	7
Locations	Low	7

The Total UFP for ILF is **62**.

2.1.2 External Interface Files (EIF)

2.1.3 External Inputs (EI)

2.1.4 External Outputs (EO)

2.1.5 External Inquiries (EQ)

2.1.6 Overall Estimation

2.2 Cost and Effort Estimation: COCOMO II

2.2.1 Scale Drivers

2.2.2 Cost Drivers

2.2.3 Effor Equation

2.2.4 Schedule Estimation

3 Project Plan Schedule

4 Resource Allocation

5 Risk Management

6 Effort Spent

Date	Domenico	Caio	Matheus
15/01/17	2h	2h	2h
16/01/17	-	-	-
17/01/17	4h	-	-
18/01/17	-	-	-
19/01/17	-	-	-
20/01/17	-	-	-
21/01/17	-	-	-
22/01/17	-	-	-

7   Changelog

As the project and design decisions may change during the development this document is also prone to change. We'll document every version in this part.

- **Version 1.1:** 22/01/2017