POLITECNICO
MILANO 1863

Software Engineering 2 project: *PowerEnJoy*

A.A. 2016/2017 - Professor E. di Nitto

Project Plan Document

Version **1.0** –22/01/2017

Pietro Avolio Mat 878640
Guido Borrelli Mat 874451

Contents

1 Introduction

1.1 Purpose and scope

The purpose of this document is to provide the estimation of cost and effort needed to implement the PowerEnJoy system..

1.2 Definitions, acronyms and abbreviations

In order to avoid ambiguity and possible misunderstandings, this table is an integration to the definitions, the acronyms and the abbreviations defined in the RASD and DD documents.

PPD	Project Planning Document
FP	Function Point
UFP	Unadjusted Function Points
IFL	Internal Logical File
EIL	External Interface File
PM	Person-Months
SLOC	Source Lines of Code
KSLOC	Kilo Source Lines of Code

1.3 Reference Documents:

- Project Management slides on BeeP
- COCOMO II Model Definition Manual
- International Function Point Users Group website, www.ifpug.org/, visited on 19/01/2017
- Quantitative Software Management website, www.qsm.com, visited on 19/01/2017

1.4 Document Structure

The document is structured in the following chapters:

Introduction This chapter provides a support to understand the aim of the document and its structure.

Function Points estimation This chapter provides an effort estimation made using the Function Points estimation technique.

COCOMO II estimation This chapter provides an effort estimation made using the COCOMO II estimation technique.

Tasks & Schedule This chapter provides detailes about the task and their schedule.

Resource allocation This chapter desribes how team members are allocated to each task of the project.

2 Function Points estimation

The Function Points estimation is a *technique to assess the effort needed to design and develop custom software applications.* It is based on the definition of combination of program characteristics, called Function Points, and on the assignment of a weight to each of these.

In this section we specify the FPs for the PowerEnJoy project and we associate a weight to each of them.

2.1 Internal Logical Files

This function type represents an *homogeneous set of data used and managed by the application.*

Internal Logical File	**Weight**
User	Medium
Vehicle	Simple
Reservation	Medium
Rental	Medium
Safe Area	Simple
Support Request	Simple
System log	High
User log	High
Vehicle position	High
Payment	Medium
Operator	Simple

The assigned weight is proportional to the number of the expected records.

2.2 External Interface Files

This function type represents an *homogeneous set of data used by the application but produced and maintained by other applications.*

External Interface File	**Weight**
Vehicle information	Simple

The assigned weight is proportional to the complexity of the entities.

2.3 External Inputs

External Input represents *elementary operation needed to elaborate data coming from the external environment.*

External Input	**Weight**
User registration	Simple
User login	Simple
User profile management	Simple
User payment info management	Medium
User reservation	Simple
User support request	Simple
User reservation	Simple
Rental start	Simple
Rental end	Medium
Operator login	Simple
Operator driving fine registration	Simple
Operator support request management	Simple

The assigned weight is proportional to number of the internal components involved to process the input.

2.4 External Outputs

This function type represents *elementary operation needed to produce data for the external environment.* Data from logic files are elaborated too.

External Ouput	**Weight**
Payment request	Medium
Push Notification request	Simple
SMS Gateway	Simple
Monthly invoices	Simple
System notifications for operators	Medium

The assigned weight is proportional to the number of the internal components involved to generate the output.

2.5 External Inquiries

This function type represents *elementary operation that involves both input and output: no relevant usage of logic files.*

External Inquiry	**Weight**
Vehicle search	Medium
Vehicle status	Simple
User's activity search by operator	Simple
Support requests search by operator	Simple
System log visualization	Simple
Vehicle activity visualization	Simple

The assigned weight is proportional to the complexity of the queries needed to answer the inquiry.

2.6 Unadjusted Function Points

The scores associated to each weight used to classify the Function Points are described in table 1. Scores follow the Albrecht's approach.

FP	Weight		
	Simple	Medium	Complex
ILFs	7	10	15
EIFs	5	7	10
External Inputs	3	4	6
External Outputs	4	5	7
External Inquiries	3	4	6

Table 1: Weights - Scores table

The total UFP calculation for the PowerEnJoy system is reported in table 2.

Function Type	Score * Weight			Total
	Simple	Medium	Complex	
ILFs	7 * 4	10 * 4	15 * 3	113
EIFs	5 * 1	7 * 0	10 * 0	5
External Inputs	3 * 10	4 * 2	6 * 0	38
External Outputs	4 * 3	5 * 2	7 * 0	22
External Inquiries	3 * 5	4 * 1	6 * 0	16
Total	91	59	45	**195**

Table 2: UFP calculation

3 COCOMO II estimation

COCOMO II is a statistical approach to estimate the cost, effort, and schedule when planning a new software development activity. It is based on the following mathematical formula:

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

where:

A	Is a costant that approximate the productivity constant PM/KSLOC. Its value is *2.94*
Size	Is the estimated size of the project in KSLOC
EM	Is the Effort Multiplier
E	Is an aggregation of five Scale Factors

In this section we calculate all the params listed below for the PowerEnJoy project and we calculate the final PM.

3.1 Project sizing

SLOC can be derived from the UFP value calculated in section 2.6 using the following mathematical formula:

$$SLOC = AVC * UFP$$

where:

AVC	Is a language-dependent factor. For Java EE its value is 46

Thus, for the PowerEnJoy project we obtain:

$$SLOC = 46 * 195 = \mathbf{8970}$$

3.2 Effort estimation

3.2.1 Scale factors

Scale factors, which appears as the E exponent in the main COCOMO II formula, are:

Precedentedness reflects the experience of the team members with similar, previously accomplished, projects. Low values mean no previous experience, high values mean high experience.

Development flexibility reflects the degree of flexibility in the development process. Low values mean no flexibility, high values mean no prescriptions.

Risk resolution reflects the existence and the deepening of a risk analysis. Low values mean little analysis, high values mean a complete risk analysis.

Team cohesion reflects how well the group members know each other and the capability of the whole team to share vision and commitment.

Process maturity reflects the process maturity of the organization.

Considered these definitions, the estimated scale factors and the total computed E param for the PowerEnJoy project are described in table 3.

Code	**Factor**	**Value**
PREC	Low	4.96
FLEX	Nominal	3.04
RESL	Nominal	4.24
TEAM	Very High	1.10
PMAT	Low	6.24
Total	$E = 0.91 + 0.01 * \sum_i SF_i$	**1.1058**

Table 3: Scale Factors

3.2.2 Cost drivers

Cost drivers are parameters that depend on whether the project is a case of post-architecture or early design project. From the cost drivers we can derive the Effort Multiplier used in the COCOMO II formula.

Stated that the PowerEnJoy project is a case of **post-architecture**, the estimated cost drivers for the project, together with the total computed PM param, are described in table 4.

Code	**Name**	**Factor**	**Value**
Product Factors			
RELY	Required Software Reliability	High	1.10
DATA	Data Base Size	Nominal	1.00
CPLX	Product Complexity	Nominal	1.00
RUSE	Developed for Resuability	High	1.07
DOCU	Documentation Match to Life-Cycle Needs	Nominal	1.00
Platform Factors			
TIME	Execution Time Constraint	Low	n/a
STOR	Main Storage Constraint	Nominal	1.00
PVOL	Platform Volability	Low	0.87
Personnel Factors			
ACAP	Analyst Capability	High	0.85
PCAP	Programmer Capability	Nominal	1.00
PCON	Personnel Continuity	High	0.90
APEX	Applications Experience	Very Low	1.22
PLEX	Platform Experience	Low	1.09
LTEX	Language and Tool Experience	Low	1.09
Project Factors			
TOOL	Use of Software Tools	Nominal	1.00
SITE	Multisite Development	Very High	0.86
SCED	Required Development Schedule	Nominal	1.00
Total	$\prod_{1 \leq i \leq n} C_i$		**0.9765**

Table 4: Cost Driver

3.2.3 Effort

Finally, we are able to solve the Effort Equation using the computed params:

$$PM = 2.94 * \frac{8970}{1000}^{1.1058} * 0.9765 = \mathbf{32.48}$$

3.3 Duration

We can calculate the amount of calandar time using the following mathematical formula

$$TDEV = C * (PM)^F$$

where

C	is a parameter for COCOMO II. Its value is 3.67
D	is a paramater for COCOMO II. Its value is 0.28
B	is a parameter for COCOMO II. Its value is 0.91
E	is the same exponent in the COCOMOII effort equation
F	$D + 0.2 * (E - B)$

We obtain

$$TDEV = 3.67 * (32.48)^{0.28+0.2*(1.1058-0.91)} = \mathbf{11.15\,months}$$

By diving the calculated PM by the TDEV value we obtain the estimated number of developers needed for the project:

$$\frac{32.48}{11.15} = \mathbf{2.91\,developers} \simeq \mathbf{3\,developers}$$

4 Scheduling & Task allocation

4.1 Schedule

The main goal that this section desires to achieve is to provide a global view of the project to all team members, in order to let people involved in the project know in which context are they working, their importance and stimulate them to achieve together the final goal.

The main tasks for the project are:

- Deliver the Requirement Analysis and Specification Document
- Deliver the Design Document
- Deliver the Integration Testing Plan Document
- Deliver the Project Plan Document
- Prepare a short but clear global presentation summarizing all the main aspects of the project
- Implement the software products and run unit tests
- Perform the integration test phase

The task dependencies are shown in this meaningful statechart graph:

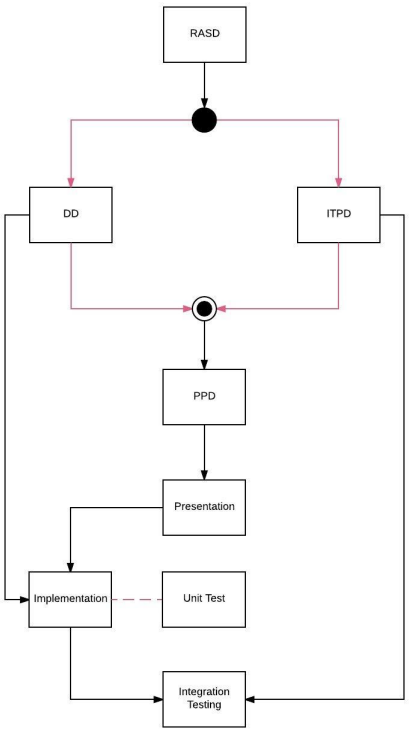

Figure 1: Documents statecharts

Table 5 reports the deadlines for each task. The implentation and integration tasks deadlines are fixed according to the value achieved in chapter 2.6.

Documents	Beginning date	Deadline
RASD	26/10/2016	13/11/2016
DD	30/11/2016	11/12/2016
ITPD	21/12/2016	15/01/2017
PPD	11/01/2017	22/01/2017
Presentation		03/02/2017
Implementation	4/02/2017	4/03/2018
Integration Testing	5/03/2018	5/05/2018

Table 5: Deadlines for each task

4.2 Task allocation

This section shows how each member of the team was involved in each task.

Resource	1st week	2nd week
Pietro	Introduction	Alloy
Guido	Overall Decription	Specific Requirements

Table 6: Resource allocation for the RASD

Resource	1st week	2nd week
Pietro	Introduction	Algorithm Design & User Interface Design
Guido	Architectural Design	Requiments Traceability

Table 7: Resource allocation for the DD

Resource	1st week	2nd week
Pietro	Integration Strategy	Test Description & Program stubs and data required
Guido	Introduction	Tools and Test equipment requested

Table 8: Resource allocation for ITPD

Resource	1st week	2nd week
Pietro	Function Points	Tasks & Schedule
Guido	Introduction	COCOMO & Risks

Table 9: Resource allocation for PPD

Resource	1st week
Pietro	Summarizing documents & slide preparation
Guido	Summarizing documents & slide preparation

Table 10: Resource allocation for Presentation

5 Risks

The risks associated with the project can be classified in three main categories: project risks, technical risks and business risks.

In this section we analyze the potential risks coming from each of these categories.

5.1 Project Risks

Project Risks are listed below, together with a classification of their possible effects provided in table 11.

1. **Lack Of Communication Among Team Members**: This project is our first team work together and one of first in absolute. Mistakes due to unexperience in estimations and arrangement could be highly probable.

2. **Requirements change**: Requirements could change or increased due to the customers or law changes. This could affect deadlines.

3. **Lack of experience in programming with the specific frameworks**: Our team has not much experience with the Java 2EE framework. This will surely affect development.

Risk	Probability	Effects
1	Moderate	Moderate
2	Low	Moderate
3	Certain	Moderate

Table 11: Project Risks

5.2 Technical Risks

Technical Risks are listed below, together with a classification of their possibile effects provided in table 2.Those kind of risks threaten the quality and timeliness of the software to be produced.

1. **DBMSs unreliability**: NoSQL and MySQL may appears unreliable for connectivity and transactions issues.

2. **Data Loss**: Data corruption during relevant communication such as PowerEnJoy Core and vehicles or PowerEnJoy App more than expected.

3. **Security Issues**: PowerEnJoy App could be afflicted by security issues during important communication such as for payments or personal identitifications.

4. **Issues with vehicles**: The system could show unwanted behaviours in case of unpredictable events with vehicles or its software or natural disasters.

Risk	Probability	Effects
1	Low	Serious
2	Moderate	Catastrophic
3	Low	Serious
4	Low	Moderate

Figure 2: Technical Risks

5.3 Business Risks

Business Risks are listed below, together with a classification of their possibile effects provided in table 12. Those kind of risks threaten the viability of the system to be.

1. **Personnel Reduction**: Personnel reduction of the work team due to financial issue.
2. **Accidental Damages**: Testing devices and infrastructure purchased that could accidentally be damaged.
3. **Law Changes**: Violation of future law retroactive.
4. **New Competitors**: Competitors could join the market and erose market shares and profitability.

Risk	Probability	Effects
1	Low	Catastrophic
2	Low	Serious
3	Low	Serious
4	Moderate	Low

Table 12: Business Risks

6 Appendix

6.1 Tools used

- LyX as LaTeX editor.
- Github as version controller.
- LucidChart for diagram

6.2 Effort spent

Pietro Avolio	12
Guido Borrelli	11