POLITECNICO
MILANO 1863

22 January 2017

PowerEnJoy
Project Plan

Blanco Federica 875487
Casasopra Fabiola 864412

Software Engineering 2 Project
2016/2017

Contents

List of Figures

List of Tables

1 Introduction

In the Project Plan document, we will show the following information:

- evaluation of the estimated size of our project;
- evaluation of the estamated effort and cost;
- identification of the tasks to be carried out and their schedule;
- allocation of the available resources to the various tasks previously defined;
- evaluation of the risks for the project.

1.1 List of Definitions and Abbreviations

Here there is the acronims and abbreviations list:

API Application Programming Interface

COCOMO COnstructive COst MOdel

DD Design Document

EI External Input

EIF External Interface File

EO External Output

EQ External Inquiry

FP Function Point

ILF Internal Logic File

ITPD Integration Test Plan Document

JEE Java Platform, Enterprise Edition

PP Project Plan

RASD Requirements Analysis and Specifications Document

SLOC Source Lines Of Code

UFP Unadjusted Function Poin

1.2 List of Reference Documents

- Specification document: Assignments AA 2016-2017.pdf
- Requirements Analysis and Specifications Document: RASD.pdf (`https://github.com/fabiola-casasopra/sw-eng-2-project/tree/master/RASD/RASD.pdf`)
- Design Document: DD.pdf (`https://github.com/fabiola-casasopra/sw-eng-2-project/blob/master/DD/DD.pdf`)
- Integration Test Plan Document: ITPD.pdf (`https://github.com/fabiola-casasopra/sw-eng-2-project/blob/master/ITDP/ITDP.pdf`)

1.3 Document overview

Here we show the stucture of the document, with a brief overview of each section.

Section 1 There is an introduction with general information about this document.

Section 2 There is the application of Function Points to estimate the size of our project

Section 3 There is the application of COCOMO to estimate effort and cost of our project.

Section 4 Here, we are going to identify the tasks to be carried out for our project and their schedule. Moreover, we are going to allocate the available resources to the various tasks.

Section 5 Here, we are going to define the risks for the project, their relevance and the associated recovery actions.

Section 6 Here there are given additional information that may be useful to the reader, such as the tools used and the time spent to redact this document.

2 Function Points Approach

The Function Point approach has been defined in 1975 by Allan Albrecht at IBM. This technique allows to evaluate the total dimension of the program, making the assumption that the dimension of software can be characterized based on the functionalities that it has to offer. Once we have estimated the size of our project, we can therefore estimete the effort needed to complete it.

The Albrecht's method identifies and count the number of function types, that represent the different functionality of the application. In particular, there are five distinct function types:

- **Internal Logic File**: ILF is the homogeneous set of data used and managed by the application;
- **External Interface File**: EIF is the homogeneous set of data used by the application but generated and maintained by other applications;
- **External Input**: EI is the elementary operation to elaborate data coming from the external environment;
- **External Output**: EO is the elementary operation that generates data for the external environment and it usually includes the elaboration of data from logic files.
- **External Inquiry**: EQ is the elementary operation that involves input and output, without significant elaboration of data from logic files.

In order to identify the development effort basing on these elements, Albrecht considered 24 applications basing on which he defined the number of FP as the sum of Function Types weighted and he compiled the following table:

Function Types	Simple	Medium	Complex
EI	3	4	6
EO	4	5	7
EQ	3	4	6
ILF	7	10	15
EIF	5	7	10

Table 1: Correlation between FP weight and the development effort

In this way, we can calculate the FPs of the our system as the sum of the FPs obtained by evaluating each one of the five different types of Functional

Points. In particular, we are going to define a number of items for every type of FPs and assign them a weight representing its complexity. As far as the list of functionalities is concernerd, we obtain it from the previously written documents: the Requirements Analysis and Specifications Document and theDesign Document.

- **Internal Logic File**: our application includes a number of ILFs used to store the information about *users*, *cars*, *reservation*, *transactions*. *User* and *Car* entities have a simple structure as they are composed of a small number of field;*Transaction* entities have a medium complex structure; *Reservation* entity instead has a complex structure. Thus, we decide to adopt the `simple weight` for *User* and *Car*, the `medium weight` for *Transaction* and the `complex weight` for *Reservation*. In a mathematical way, this means that we have $2 * 7 + 1 * 10 + 1 * 15 = 49$ FPs concerning ILFs.
- **External Interface File**: our application features only one EIF, in order to manage the interaction with **Google Maps** APIs. This information results in only one entity, that we guess has a complex structure. Thus, we decide to adopt a `complex weight`. In a mathematical way, this means that we have $1 * 10 = 10$ FPs concerning EIFs.
- **External Input**: our application interacts with the user to allow them to:
  - *Login*, *Logout* and *Sign Up*: these are simple operations, so we can adopt the `simple weight` for them: $3 * 3 = 9$ FPs.
  - *Update Account*: this is a simple operation, so we can adopt a `simple weight`: $1 * 3 = 3$ FPs.
  - *Reserve a Car*: this is not a simple operation, so we think that it is suitable to adopt the `medium weight`: $1 * 4 = 4$ FPs.
  - *Cancel a Reservation*: this is not a simple operation, so we think that it is suitable to adopt the `medium weight`: $1 * 4 = 4$ FPs.

  As a result, we get $9 + 3 + 4 + 4 = 20$ FPs.
- **External Output**:
  - *Look for Safe Areas and Power Grid Stations*: after a specific request, our application shows the user the *Safe Areas* and the *Power Grid Station* around a choosen location. For this operation, we think that it is suitable to adopt the `complex weight`: $2 * 6 = 8$ FPs.

- *Amount to Pay*: when the user parks the car, our application provides a receipt, with the eventual discounts or charges already applied. For this operation, we think that it is suitable to adopt the `medium weight`: $1 * 5 = 5$ FPs.

  As a result, we get $8 + 5 = 13$ FPs.

- **External Inquiry**: our application allows users to access some information.

  - *See Transactions' Details*: this is a simple operation, so we can adopt a `simple weight`: $1 * 3 = 3$ FPs.

  As a result, we get 3 FPs.

Now, we can compute the value for the Unadjusted Function Poin, with the following formula:

$$UFP = \sum(\#elements\_of\_a\_given\_type * weight)$$

Considering the previously part, we have that the total UFP is computed as following: $UFP = 49 + 10 + 20 + 13 + 3 = 95$

Thanks to this value, we can estimate the effort in two ways:

- *directly*: this approach is viable only in case we have some historical data, so that we can know how much time it usually take for developing a FP;
- *indirectly*: this approach consists in computing the size of the project in Source Lines Of Code, starting from the total UFP, and then use another approach, such as COCOMO II, in order to estimate the effort.

Since we don't have any historical data, we aren't able to make a direct estimation, so we take in consideration the undirect approach. In order to convert the UFP into SLOC, we decided to use the standard conversion coefficient. We find out that the coefficient related to the specific framework we are using, that is JEE, is equal to 46, as shown in the table at the following link: `http://www.qsm.com/resources/function-point-languages-table` Therefore, we can finallu compute the SLOC for our project:

$$SLOC = 46 * FPs = 4370$$

3 COCOMO Approach

4 Project Schedule and Resource Allocation

5 Project Risks

6 Appendix

In this section, we will give the information about the used tools, the hours of work done by the members of the group.

6.1 Used Tools

In this phase of the project, the following tools have been used:

- LaTeX and TeXMaker editor: to redact and to format this document

6.2 Working Hours

Last Name	First Name	Total Hours
Blanco	Federica	? h
Casasopra	Fabiola	? h