



# **POLITECNICO**

## **MILANO 1863**

### **POWER ENJOY**

#### **Project Plan Document**

**Lorenzo Casalino - 877421**

**Tommaso Castagna**

Document version 1.0

# Contents

## 1 Introduction

1.1	Revision History	.....
1.2	Purpose	.....
1.3	Scope	.....
1.4	Definitions, Acronyms and Abbreviations	.....
1.5	Reference Documents	.....

## 2 Project Size, Effort and Cost Estimation

2.1	Size Estimation	.....
2.1.1	Internal Logic File	.....
2.2	External Interface File	.....
2.2.1	External Input	.....
2.3	Effort and Cost Estimation	.....

# 1 Introduction

## 1.1 Revision History

The history of document revisions is here recorded in tabular format, mapping the document version with the changes brought to document itself.

The current version of the document is highlighted by the version number in bold format.

Version	Revision
<b>1.0</b>	First released version.

## 1.2 Purpose

## 1.3 Scope

## 1.4 Definitions, Acronyms and Abbreviations

## 1.5 Reference Documents

## 2 Project Size, Effort and Cost Estimation

In this section the estimation process of the three key aspects for an effective project planning, namely size, effort and costs expected, is described in details, pointing out the rationale of each single step of the process itself.

The size estimation process is led by a **functionality-provided** based approach, whereby the estimation is made according to the functionalities that the software product is planned to provide. To support this strategy, *Function Points* technique is used.

Regarding the effort and cost estimation, the process is based on **algorithmic approach**, that is the use of an algorithmic model based on a simple equation which output depends on several factors regarding the project. The algorithmic model here used is based on *COCOMO II*.

### 2.1 Size Estimation

As explained in the introductory paragraph to this section, the size estimation effort is based on the estimation of the so-called *Function Points*. Function points are a statistical method of estimate the size of a software project evaluating the different functionalities provided by the software product in exam.

According to this approach, functionalities are divided into 5 **function types**, or category:

- **Internal Logical File.**
- **External Logical File.**
- **External Input.**
- **External Output.**
- **External Inquiry.**

For each of these type, a **weight** is associated. These weights are statistically determinated and vary according to the **complexity** of the function type. The complexity of a function type can be derived consulting the related *rating tables*.

RET	Data elements		
	1 → 19	20 → 50	51+
1	Low	Low	Average
2 → 5	Low	Average	High
6+	Average	High	High

In order to retrieve the number of function points assigned to the software product given a specific function type, a simple equation is applied:

$$PFP_t = N_t * FP_t, t \in T = [ILF, ELG, EI, EO, EINQ]$$

This equation returns the *partial function points* obtained by multiplying the number of functionalities of a certain category for the weight associated to that category.

The total number of function points assigned to the whole project is computed by the *Unadjusted Function Points* equation, which simply compute the sum of each PFP previously calculated.

$$UFP = \sum_t PFP_t, t \in T$$

### 2.1.1 Internal Logic File

RET	Data elements		
	1 → 19	20 → 50	51+
1	Low	Low	Average
2 → 5	Low	Average	High
6+	Average	High	High

- **User account informations:** this file contains the informations that any user is asked to type at registration time to the service. The informations (Name, surname, password, email, telephone number, postal code, city, birthday, driving license, payment informations) are recorded inside one type of record.
- **Operator account informations:** the purpose is similar to the logic file storing the user account informations, but with the difference of not containing the driving license and the payment informations. Only one type of record is exploited.
- **Priviledges:** to avoid the execution of actions not allowed to some category of users, this file is used to map the priviledge level associated to each account category. Only one record type is used.
- **Parking area informations:** area informations, such as area identifier and coordinates describing the geographical boundaries of the parking area, are stored in this internal file, by means of one record type.
- **Safe parking area informations:** extension of the previous internal logic file, which adds the presence of the total number of parking spots inside the area.

- **Vehicle informations:** this file contains both the usual informations characterizing a vehicle (Plate number, frame number, model, matriculation date, fuel type, shift type) and the informations regarding the actual status of the vehicle (Fuel percentage, availability, position). Two records are used to store, respectively, the type of informations described above.
- **Reservation informations:** the informations regarding each reservation is stored by the support of three records. The first record stores an identifier to the reservation, the begin and end date of the reservation, the identifier of the reserving user, the plate number of the reserved vehicle and the total charge. The second structure is used to keep track of events that trigger a policy rule. An identifier to the event, the identifier of the policy rule, event date, event condition and effect are stored. The third record is used to map a reservation to one or more event.
- **Policy rules:** to implement the detection of good or bad behaviours, according to the policy adopted by the car sharing society, the rules are encoded and stored in an appropriated format. This logic file serves this purpose, using one record structure to store the rule identifier, the encoded conditions and encoded effects.
- **Maintenance tasks:** maintenance tasks assigned to a certain operator are stored by this logic file, using two record type. The first used to store the identifier task and its description, while the second is used as a support, mapping tasks to a specific operator using his/her identifier.

Making reference to the ILF rating table, we can evaluate the partial function point value:

User account informations	Low	7
Operator account informations	Low	7
Priviledges	Low	7
Parking area informations	Low	7
Safe parking area informations	Low	7
Vehicle informations	Low	7
Reservation informations	Low	7
Policy rules	Low	7
Maintenance tasks	Low	7
Partial Function Point	63	

## 2.2 External Interface File

:

The external services used by our system are developed by third parts. Consequently, it's quite hard to understand how many record types and data elements

	Data elements		
RET	1 → 4	5 → 15	15+
i 2	Low	Low	Average
2	Low	Average	High
i 2	Average	High	High

of each external file are used. In order to perform a meaningful estimations, the following assumptions about the external services are taken in consideration.

- **Driver licenses validator web service files:** we assume that the web service needs only to use informations regarding stored driver licenses and drivers informations, hence use of two record structure types.
- **Payment system web service files:** as for the previous web service, it's reasonable think that the payment web service relies only the payment informations and payment service's customers informations.
- **Google maps web service files:** the kind of informations needed by this service are unknown to us, but we assume that the operations performed for map plotting and so on involves a complex external file.

Driver licenses validator web service	Low	5
Payment system web service	Low	5
Google maps web service	High	10
Partial Function Point	20	

### 2.2.1 External Input

- **Login:** elementary operation that involves the user account informations (or the operator account informations).
- **Logout:** elementary operation that does not require the involvement of any internal or external file to achieve the result.
- **Register account:** functionality that relies on the user account internal logic file.
- **Delete account:** operation relying only on user account internal logic file.
- **Update account informations:** this operation is more complex with respect to the last two since modification of a user information can involves other internal or external files. Indeed, for instance, the payment informations editing needs the intervention of the payment web service in order to verify the validity of the updated informations, hence the involvement of the payment system external file. This external input relies on

the user account internal file, the payment system web service and driver license web service external files.

- **Create reservation:** the creation of a reservation from a user is a quite simple operation, involving only the reservation and the vehicle information internal logic files.
- **Delete current reservation:** simple operation relying only on the reservation informations internal file.
- **Terminate reservation:** complex operation that need the support of the reservation informations, vehicle informations internal logic file and the payment web service external file.
- **Update reservation:** as for the terminate reservation, this operation is quite complex and involves the reservation informations and vehicle informations.
- **New vehicle insertion:** simple operation relying only on the vehicle informations internal logic file.
- **Vehicle deletion:** simple operation relying only on the vehicle informations internal logic file.
- **Vehicle informations update:** simple operation where vehicle informations internal file is used.
- **Insert new policy rule:** operation through which the policy rules information file is modified.
- **Remove policy rule:** elementary operation involving only the policy rules information file.
- **Update policy rule:** complex operation involving the policy internal file and the reservation internal file.
- **Insert parking area:** the insertion of a new parking area, either a normal area or a safe area, is an operation which involves the parking area and the safe parking area internal files.
- **Remove parking area:** basic operation involving the parking area and the safe parking area internal files.
- **Update parking area informations:** same complexity and same internal files involved as for the previous two parking area-related functionalities.
- **Insert new task:** new tasks to be assigned to an operator is a simple operation where only the maintenance task informations file is involved.
- **Remove task:** simple functionality modifying the maintenance task information file.



- **Update task:** simple operation involving solely the maintenance task information file.

Login	Low	3
Logout	Low	3
Register account	Low	3
Delete account	Low	3
Update account	Average	4
Partial Function Point	20	

## 2.3 Effort and Cost Estimation