

PROJECT for

SOFTWARE ENGINEERING 2

**Project Management v1**

Politecnico di Milano

A.A. 2016-2017

Prof.ssa Elisabetta Di Nitto

Students:

Diego Gaboardi

Giorgio Giardini

Riccardo Giol

INDEX

* 1 Introduction
  + 1.1 Purpose and Scope
  + 1.2 Definition, Acronyms
    - 1.2.1 Definitions
    - 1.2.2 Acronyms
  + 1.3 Reference documents
* 2 Project size, cost and effort estimation
  + 2.1 Introduction
  + 2.2 Size estimation: functions points
    - 2.2.1 Internal Logic Files (ILFs)
    - 2.2.2 External Interface Files (EIFs)
    - 2.2.3 External Inputs (EIs)
    - 2.2.4 External Inquiries (EQs)
    - 2.2.5 External Outputs (EOs)
    - 2.2.6 Overall estimation
  + 2.3 Cost and effort estimation: COCOMO II
    - 2.3.1 Scale Drivers
    - 2.3.2 Cost Drivers
    - 2.3.3 Effort equation
    - 2.3.4 Schedule estimation
* 3 Schedule
* 4 Resource Allocation
* 5 Risk management

1 INTRODUCTION

* 1. PURPOSE AND SCOPE
  2. DEFINITIONS AND ACRONYMS
  3. REFERENCE DOCUMENTS

1. SIZE, COST AND EFFORT ESTIMATION
   1. INTRODUCTION

In this section we are going to estimate the following properties about our project: the expected size, the cost and the required effort.

So we are going to use the Function Points approach in order to obtain a good estimation of the size of our project in term of hypothetic number of lines of code.

Then for the cost and effort estimation we will use the COCOMO approach. In this way starting from the number of lines of code we will estimate the number of person hours necessary for developing the whole project.

* 1. SIZE ESTIMATION: FUNCTION POINTS

The Function Points approach allows us to obtain a good estimation of the size of our project considering the functions that are meaningful for our software. In particular we will consider the following program characteristics called function types:

* Internal Logic Files
* External Interface Files
* External Input
* External Output
* External Inquiry

In order to have a good estimation we will use this table obtained with statistical procedures from real projects.

|  |  |  |  |
| --- | --- | --- | --- |
| FUNCTION TYPES | WEIGHT | | |
| **Simple** | **Medium** | **Complex** |
| Internal Logic Files | 7 | 10 | 15 |
| External Interface Files | 5 | 7 | 10 |
| External Inputs | 3 | 4 | 6 |
| External Outputs | 4 | 5 | 7 |
| External Inquiries | 3 | 4 | 6 |

1. INTERNAL LOGIC FILES (ILFs)

In this section I am going to list and analyse the most important Internal Logical File (ILF) of our application and so the homogeneous sets of data used and managed by the application.

First of all our system has to store all the information about clients and so the following fields: name, surname, mobile phone number, email address, password, code, driving licence, credit card and his state (Dismounted, Reserving, Driving, OnBreak). All this attributes are strings (except the state) and are stored in a single table. In a similar way also the assistants are structured even if they don’t have associated the driving licence, the credit card and the state.

Then another data managed by PowerEnjoy is the car which has several fields: code, model, battery, capacity which are stored as strings or integers. Then we have also the state of the car (Available, Reserved, OnCourse, OnBreak and NonAvailable) and two Boolean variable pointing out that the car is locked and in recharging state.

Then we store also the information about the town of Milan that are necessary for our application. So we consider the set of safeAreas which are identified by a code and by their coordinates. In a similar way also the parking stations are stored. However in this situation we must also save the information about rechargers which have a code and a Boolean that indicates if they are free or occupied.

Then there are the courses which are more complex than the previous one and so contains more attributes: the time of start and end, the data of start and end, the cash amount, the position of start and end, the discounts used, the client and the car involved and the correspondent reservation. All this field are stored in a single table.

Finally there are the reservations that must contains the client, the car and the course associated, and the information about the countdown.

All these sets of data have an extremely simple structure being composed of a small number of fields. The only exceptions are courses which are more structured and therefore can be considered of medium complexity.

The following table summarizes what has been said in this paragraph.

|  |  |  |
| --- | --- | --- |
| ILF | Complexity | FPs |
| Clients  Assistants  Cars  SafeAreas  ParkingStation  Courses  Reservations | Simple  Simple  Simple  Simple  Simple  Medium  Simple | 7  7  7  7  7  10  7 |
| Total | | 52 |

1. EXTERNAL INTERFACE FILES (ELFs)

With External Interface Files (EIF) we mean a user identifiable group of logically related data that is used for reference purposes only. This kind of data resides entirely outside the application boundary and is maintained by another application external inputs. The external interface file is an internal logical file for another application.

In our project the only external data sources are the following:

* The mapping service which manages all the information about the positions in term of latitude and longitude;
* The payment system which handles the payments that occur in the services offered by the application.

In both the situation the amount of data that is used is extremely small and so their complexity is considered simple.

|  |  |  |
| --- | --- | --- |
| EIF | Complexity | FPs |
| Map position  Paiments | Simple  Simple | 5  5 |
| Total | | 10 |

1. EXTERNAL INPUTS (EIs)

In this section we are going to consider all the typologies of interactions with external users of our application. In fact with External Input we mean data that are coming external to the application and may be used to maintain one or more internal logical files. In order to be clear the inputs will be divided according to the types of user.

Clients:

* Login/Logout: the client must insert his user name and the password in the proper form.
* Registration: the client must insert all required data which are immediately verified.
* Modify profile: the client after the login must insert the new credentials.
* Reserve a car: the client must insert his current position or a specific one and then has to select the car that he wants to pick up. He can also enable the saving option in order to find safe areas near the destination inserted. This is a slightly more structured operation and so can be considered of medium complexity.
* End course/Leave car in break: the client must answer if he wants to end is ride or leave the car in break.

Assistant:

* Login/Logout: it is the same situation as in the client
* Change car state: the assistant must insert the new state of the car. If the modified state in NonAvailable he must add the type of damage.

All these external inputs are extremely easy because they involve few inputs.

The first exceptions is client registration in which the user has to insert a lot of data that has to be verified and so it can be considered of medium complexity.

Then the reservation of a car is complex because there are interactions with several other entities such as car, course, client and safe area.

In the following table we show the function points associated with the External Inputs.

|  |  |  |  |
| --- | --- | --- | --- |
| EI | Complexity | | FPs |
| Client login/logout  Client registration  Modify profile  Reserve a car  End course / leave the car  Assistant login/logout  Change car state | | Simple  Average  Simple  Complex  Simple  Simple  Simple | 2x3  4  3  6  3  2x3  3 |
| Total | | | 31 |

1. EXTERNAL INQUIRIES (EQs)

In this section we are going to analyse the External Inquiries that occur in our application. They are elementary processes that send data or control information outside the application boundary in order to present data to a user through the retrieval of data or control information from an ILF or EIF.

In PowerEnjoy there are the following External Inquiries:

* A client can retrieve the actual reservation and the historical one.
* A client can retrieve the actual course and the historical one
* A client can retrieve all available cars near a position inserted or calculated by the GPS.
* A client can retrieve the sets of safe areas near a position inserted (this process occurs when he activates the saving option).
* An assistant can retrieve the list of cars and the information about their state.

All these operations can be carried out easily through query that extract data from a singular table and so can be considered of simple complexity.

|  |  |  |  |
| --- | --- | --- | --- |
| EQ | Complexity | | FPs |
| Retrieve reservation  Retrieve course  Retrieve car available near position  Retrieve safe areas near position  Retrieves cars and their state (by assistants) | | Simple  Simple  Simple  Simple  Simple | 3  3  3  3  3 |
| Total | | | 15 |

1. EXTERNAL OUTPUTS (EOs)

In this section we are going to examine External Output and so the elementary processes that send data or control information outside the application boundary. The primary purpose of external outputs is to present information to a user through processing logic other than, or in addition to the retrieval of data or control information.

The situation in which our system has to communicate to the user outside the context of an inquiry are the following:

* Notify the client that the registration has been completed successfully.
* Notify the client that the reservation has been accepted.
* Notify the client that the payment has been completed successfully.
* Notify the client that the car has been locked properly.
* Notify the client that the ReservationCountdown or the CourtesyCountdown expired
* Notify the assistant that the car state has been changed correctly.

All these operations are extremely easy because they involve only few data in the interactions with the external users and for this reason they can be considered of simple complexity.

In the following table the External Output are listed with the Function Points associated.

|  |  |  |  |
| --- | --- | --- | --- |
| EO | Complexity | | FPs |
| Registration completed notification  Reservation accepted notification  Payment completed notification  Car locked notification  Countdown expired notification  Car state changed notification | | Simple  Simple  Simple  Simple  Simple  Simple | 4  4  4  4  4  4 |
| Total | | | 24 |

1. OVERALL ESTIMATION

Finally in this last section we summarize the results obtained in the previous paragraphs. In particular the following table shows the function points associated with all function types

|  |  |
| --- | --- |
| Function type | FPs |
| Internal Logic Files  External Logic Files  External Inputs  External Inquiries  External Outputs | 52  10  31  15  24 |
| Total | 132 |

So in conclusion we have 132 Function Points that according to the following table can be converted in lines of code.

* High: SLOC = 132 \* 67 = 8844
* Median: SLOC = 132 \* 49 = 6468
* Average: SLOC = 132 \* 46 = 6072
* Low: SLOC = 132 \* 15 = 1980

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| LANGUAGE | SLOC/FP | | |  |
| **Average** | **Median** | **Low** | **High** |
| JEE2 | 46 | 49 | 15 | 67 |

* 1. COST AND EFFORT ESTIMATION: COCOMO II