***PowerEnJoy***



**Project Plan Document**

Authors:

Emanuele Chilà

Giorgio Lazzarinetti

January 20th 2017

SUMMARY

[1. INTRODUCTION 3](#_Toc472633659)

[1.1. PURPOSE AND SCOPE 3](#_Toc472633660)

[1.2. LIST OF DEFINITION AND ABBREVIATIONS 3](#_Toc472633661)

[1.3. LIST OF REFERENCE DOCUMENT 4](#_Toc472633662)

[2. PROJECT SIZE, COST AND EFFORT ESTIMATION 5](#_Toc472633663)

[2.1. SIZE ESTIMATION: FUNCTION POINTS 5](#_Toc472633664)

[2.1.1. INTERNAL LOGIC FILES (ILFs) 6](#_Toc472633665)

[2.1.2. EXTERNAL INTERFACE FILES (EIFs) 7](#_Toc472633666)

[2.1.3. EXTERNAL INPUT (EIs) 8](#_Toc472633667)

[2.1.4. EXTERNAL OUTPUT (EOs) 10](#_Toc472633668)

[2.1.5. EXTERNAL INQUIRY (EQs) 11](#_Toc472633669)

[2.2. COST AND EFFORT ESTIMATION: COCOMO II 12](#_Toc472633670)

[2.2.1. SCALE FACTOR 13](#_Toc472633671)

[2.2.1.1. PRECEDENTEDNESS RATING LEVELS 14](#_Toc472633672)

[2.2.1.2. DEVELOPMENT FLEXIBILITY RATING LEVELS 15](#_Toc472633673)

[2.2.1.3. RISK RESOLUTION RATING LEVEL 16](#_Toc472633674)

[2.2.1.4. TEAM COHESION RATING LEVEL 16](#_Toc472633675)

[2.2.1.5. PROJECT MATURITY RATING LEVEL 17](#_Toc472633676)

[2.2.1.6. SCALE FACTOR RESULT 18](#_Toc472633677)

[2.2.2. COST DRIVERS 18](#_Toc472633678)

[2.2.3. EFFORT EQUATION 24](#_Toc472633679)

[2.2.4. SCHEDULE ESTIMATION 25](#_Toc472633680)

[3. SCHEDULE 25](#_Toc472633681)

[4. RESOURCE ALLOCATION 28](#_Toc472633682)

[5. RISK MANAGEMENT 30](#_Toc472633683)

[6. EFFORT SPENT 32](#_Toc472633684)

[8. REFERENCES 32](#_Toc472633685)

## **INTRODUCTION**

### **PURPOSE AND SCOPE**

The purpose of the Project Plan Document is to define the Size, Effort Cost, Project Schedule and Risks of the PowerEnJoy project.

In the first part of the document we are going to cover the Size and Effort cost estimation.

In order to estimate the Size and Effort Cost of the project we are going to apply an Algorithmic cost modelling, in detail: Function Point approach for the Size estimation and COCOMO for the Effort Cost estimation.

In the second part of the document we are going to cover the Project Schedule planning.

The Project Schedule is used in order to define a possible schedule for the project thus supporting the activities of resource allocation, programming and testing.

In the third part of the document we are going to cover the Risks definition.

The Risks definition allow us to consider the effect and the probability of a possible failure on the development process and so take the necessary precautions.

The last part of the document is dedicated to some general conclusion.

### **LIST OF DEFINITION AND ABBREVIATIONS**

* **RASD**: requirement analysis and specification document;
* **DD**: design document;
* **ITPD**: integration test plan document;
* **PPD**: project plan document;
* **API**: application programming interface, common way to communicate with another system (for example we use this kind of interface to communicate with the Google map services);
* **Car**: we mean, of course, an electric car;
* **Reservation**: It’s a booking made by a user and paid to the use of a certain selected car;
* **Safe Area**: We mean the areas where the user can park the rented car;
* **Special Parking Area**: We mean the areas where the user can both park the rented car and recharge it (all the special parking area are safe area, but not vice versa);
* **Extraordinary situation**: We mean extraordinary events that can happen to the users such as car accident, car’s battery running low etc;
* **Driver:** testing module to perform method invocation of integrated component and so allow the testing of those component;
* **LOC or SLOC:** Line of Code or Source Line of Code, unit of measure that indicate the number of line of code in a project, giving so an idea about the size of it;
* **FP:** Function Point, unit of measure used to estimate the SIZE of a project in the Function Point Approach, the Function Point are easily convertible into LOC measure;
* **PH:** Person Hour;
* **ILF:** Internal Logical File;
* **EIF:** External Interface File;
* **EI:** External Input;
* **EO:** External Output;
* **EQ:** External Inquiry;

### **LIST OF REFERENCE DOCUMENT**

* DD produced before 3.0 version
* RASD produced before 2.0 version
* ITPD 1.0 versione
* Specification Document: Assignment 1,2, 3 and 5 (RASD, DD, ITPD and PPD).pdf

## **PROJECT SIZE, COST AND EFFORT ESTIMATION**

This section is focused on the estimation of the expected size and effort cost of PowerEnJoy.

For the size estimation part, we are going to use an Algorithmic cost modelling approach named Function Point Approach calculating so the total FP that estimate the size of the project.

In this estimation, we are going to consider just the functionalities covered by the business logic (because they are the most important and critical one).

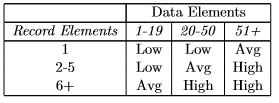
For the effort cost estimation, we are going to use an Algorithmic cost modelling approach named COCOMO approach, from which we are going to calculate the PH estimated to ultimate the project, the COCOMO approach considers the FP estimated for the Size (from the Function Point Approach) and add the FP estimated for the effort that has to be spent.

### **SIZE ESTIMATION: FUNCTION POINTS**

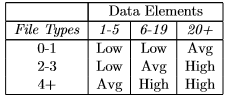
The Function Points approach provides an estimation of the size of a project taking as inputs the number of functionalities to be developed and their complexity.

The estimation is based on the usage of ﬁgures obtained through statistical analysis of real projects, which have been properly normalized and condensed in the following tables:

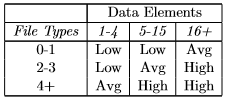
For Internal Logic Files and External Logic Files



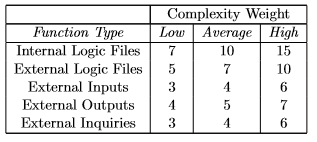
For External Output and External Inquiry



For External Input



UFP Complexity Weights



### **INTERNAL LOGIC FILES (ILFs)**

PowerEnJoy relies on a number of ILFs to store the information it needs to oﬀer the required functionalities. In the next few paragraphs, we will analyse in detail the various ILFs we have identiﬁed.

* The system has to store information about the electric car owned by PowerEnJoy this is done using a single table which contains the car plate number, the car status (Available, Rented, InUse Damaged) the ID of the monitor and the sensors IDs that are installed in the car, the battery status, eventually the PowerGrid in which the car is plugged.
* The system has to store information about the SafeArea, this is done by managing a two level structure, the first level holds the identifier of all the zones and the radius of the circle that identifies the range in which you are still in the safe area while the second one holds the position of the SafeArea, the position contains all the location coordinates as <latitude,longitude> necessary to identify its position.
* The system has to store information about the SpecialParkingArea which has a structure identical to the SafeArea but adds in the first level structure the information about the IDs of the PowerGrid that are installed in that area.
* The system has to store information about the Reservations. This is done with a single table which contains the user that made the reservation and the car that is reserved, the status of the reservation and the time in which the reservation was created.
* The system has to store information about the Payments generated after a reservation. This is done with a single table that contains information about the reservation of which it is connected and the amount of payment including charge and discounts made. Furthermore, the table contains the information about the fact that the payment is fulfilled or not, this is not a very simple task because the system has to interact with banks API’s in order to understand if a payment has or not been fulfilled.
* The system has to store information about the Users, this is done with a single table that contains ID, name, surname, birthdate, username, password, e-mail and the reference to another structure which contains the user’s Credit Card information.
* The system has to store information about the user’s Credit Card, this is done with a single table which contains information about: Credit Card number, CWD, expiration date.

|  |  |  |
| --- | --- | --- |
| **ILF** | **Complexity** | **FPs** |
| Car  Safe Area  SpecialParkingArea  Reservation  Payment  User  Credit Card | Low  Low  Low  Low  Low  Average  Low | 7  7  7  7  7  10  7 |
| Total | | 52 |

**ILF COMPLEXITY RATIONALE:**

The reason why the complexity of the structures that contain this information is low, is the fact that they are all composed of simple tables with a small number of fields, the only exception has been made on the information regarding payment because they must be managed by interfacing with an external system.

### **EXTERNAL INTERFACE FILES (EIFs)**

There are two kind of external data source managed by PowerEnJoy: the information related with the Mapping Services, the information related with the sensor system installed into the car, the information related to the status of a payment (bank side).

* For the Mapping Services, we need information about the graphical representation of the city (for the client side) and the coordinates related to a certain given address
* For the Sensor System in the car, we need information about the parameters measured by the sensors.
* For the payments from the bank side point of view, we need the information about the status of the payment.

|  |  |  |
| --- | --- | --- |
| **EIF** | **Complexity** | **FPs** |
| Map data retrieval  Reverse geocoding  Sensor measure  Payment status retrieval | Low  Low  Low  Low | 10  10  10  10 |
| Total | | 40 |

**ELF COMPLEXITY RATIONALE:**

The reason why the complexity of the structures that contain this information is low, is the fact that they are individual values or small groups of values ​​to which it is easy to access.

### **EXTERNAL INPUT (EIs)**

We identify an internal output as an elementary operation that elaborate data coming from the external environment.

These are the offered feature by PowerEnJoy we include the rationale for which we decided what complexity give to every operation:

* **Login/Logout**: simple operations, low level complexity 3FPs each.
* **Password retrieval**: simple operation, low level complexity 3FPs.
* **Change settings**: simple operation, low level complexity 3FPs.
* **Search for an available car**: Hard operation because it involves the Customer Application, Server, Management Application and DBMS, High level complexity 6FPs.
* **Rent a car**: Hard operation because it involves the Customer Application, Server, Management Application and DBMS, High level complexity 6FPs.
* **Lock/Unlock the rented car**: These operations involve Customer Application Management Application and Car application and it has to interact with the Car System in order to open it so they have a High level complexity with 6FPs each.
* **Unlock for the first time the rented car/Lock the car to finish the reservation**: These operations are the same as Lock and Unlock they have some more operations that manage the reservation on the tail of the tasks to do, since we can reuse the code of the operation listed before we don’t change the complexity, High level complexity 6FPs each.
* **Register a new account:** Average difficulty operation because we have to check the Credit Card credential 4FPs.
* **Insert, Delete and Update SafeArea and SpecialParkingArea:** High level complexity because there are involved a great number of component, 6FPs each.
* **Insert, Delete and Update cars:** High level complexity because there are involved a great number of component, 6FPs each.
* **Calculate reservation cost:** This is a simple operation that can be done with just one method, low level complexity 3FPs.
* **Communicate with the Call Centre:** Average difficulty operation because we to get to communicate the Call Centre Application with the Customer Application, Average level complexity 4FPs.
* **Open a new accident dossier:** simple operation because it is a simple query to the DBMS, low level complexity 3FPs.
* **Change car status (manually by the call centre application):** simple operation because it is a simple query to the DBMS, low level complexity 3FPs.

|  |  |  |
| --- | --- | --- |
| **EI** | **Complexity** | **FPs** |
| Login/Logout  Password retrieval  Change settings  Search for an available car  Rent a car  Lock/Unlock the rented car  Unlock for the first time the rented car  Lock the car to finish the reservation  Register a new account.  Insert, Delete and Update SafeArea and SpecialParkingArea  Insert, Delete and Update cars  Calculate reservation cost  Communicate with the Call Centre  Open a new accident dossier  Change car status (manually by the call centre application) | Low  Low  Low  Hard  Hard  Hard  Hard  Hard  Average  Hard  Hard  Low  Average  Low  Low | 3x2  3  3  6  6  6x2  6  6  4  6x3  6x3  3  4  3  3 |
| Total | | 101 |

### **EXTERNAL OUTPUT (EOs)**

We identify an external output as an elementary operation that generates data for the external environment.

These are the functionalities that PowerEnJoy uses in order to communicate through an output with the user:

* Notify the user that the reservation has been accepted;
* Notify the user that the expiration time for the first unlock of the car is fading;
* Notify the user that the expiration time for the total time available for the rental is fading;
* Notify the user that the car he is riding has low battery;
* Notify the user about the nearest SafeArea/SpecialParkingArea;
* Notify the user that the reservation has concluded successfully;
* Show the user given a certain position and a distance all the available car in that zone;
* Notify the user that the car has been Locked/Unlocked;
* Notify the user that he cannot rent a car because he has an unfulfilled payment;

|  |  |  |
| --- | --- | --- |
| **EO** | **Complexity** | **FPs** |
| Reservation accepted  Notification about the first unlock timer  Notification about the maximum time available for the rental  Notification about the battery running low  Nearest SafeArea/SpecialParkingArea  Notification about end of reservation  Show available cars  Notification about Lock/Unlock of the car  Notification about the impossibility to rent a car | Low  Low  Low  Low  Low  Low  Average  Low  Low | 4  4  4  4  4  4  5  4  4 |
| Total | | 37 |

**EO COMPLEXITY RATIONALE:**

The reason why the complexity of this function is low, is the fact that they are very simple operations that don’t involve the interaction with many components. The only exception is the “Show available cars operation” which is more complex because it involves the interaction between Customer Application, Server, Management Application and DBMS and is part of one of the core operations of the system that is Search for available cars.

### **EXTERNAL INQUIRY (EQs)**

We identify an external inquiry as a data retrieval operation performed by the user.

The functionalities of this type that PowerEnJoy supports are:

* A User can retrieve the position of the rented car;
* A User can retrieve his payment history;
* A User can retrieve general information about his current reservation;
* A User can retrieve the reservation expiration time;
* A User can retrieve the current cost of the rental;
* A User can retrieve the list of near SafeArea and SpecialParkingArea;

|  |  |  |
| --- | --- | --- |
| **EQ** | **Complexity** | **FPs** |
| Retrieve car Position  Retrieve Payment history  Retrieve general information about the reservation  Retrieve reservation expiration time  Retrieve the current rental cost  Retrieve the list of near SafeArea and SpecialParkingArea | Low  Low  Low  Low  Low  Low | 3  3  3  3  3  3 |
| Total | | 18 |

**EQ COMPLEXITY RATIONALE:**

The reason why the complexity of this function is low, is the fact that they are very simple operations that don’t involve the interaction with many components.

* + 1. **OVERALL ESTIMATION**

The following table summarizes the results of our estimation activity

|  |  |
| --- | --- |
| **EQ** | **Value** |
| ILF  EIF  EI  EO  EQ | 52  40  101  37  18 |
| Total | 248 |

Considering Java Enterprise Edition as a development platform we can estimate the total number of lines of code.

The following calculation uses as parameter the average number of SLOC per FP.

### **COST AND EFFORT ESTIMATION: COCOMO II**

In this section, we are going to use the COCOMO II approach to estimate the effort cost needed to develop PowerEnJoy, the size parameter used in the COCOMO’s equation is the size calculated in chapter 2.1. with the Function Point approach.

The COCOMO ‘s II evaluation function is the following:

Where:

A = is a constant which value is 2.94 (for COCOMO II)

EAF = product of all cost driver

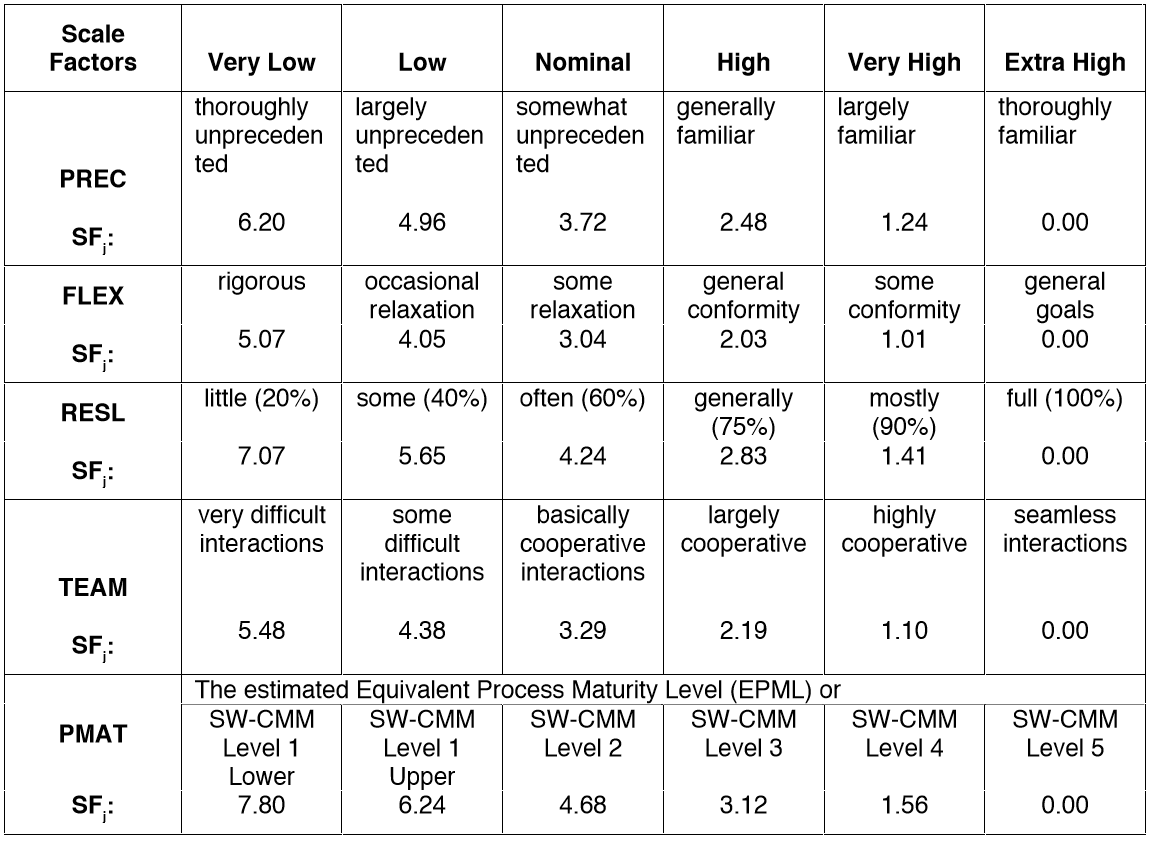
E = Scale Factor

On chapter 2.2.3 we are going to precisely explain how these parameters are calculated.

### **SCALE FACTOR**

We are now going to estimate the scale factor parameter value, to do so we refer to the following official COCOMO II table.

Scale Factor values, , for COCOMO II Models

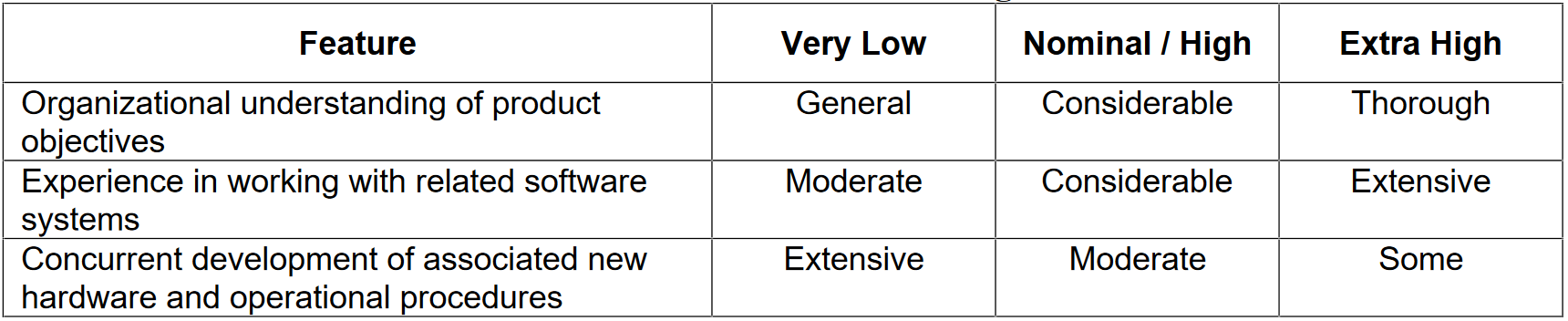
****

A brief description of each scale driver with the rationale for which we decided to take that value:

* **Precedentedness**: it reﬂects the previous experience of our team with the development of large scale projects
* **Development ﬂexibility**: it reﬂects the degree of ﬂexibility in the development process with respect to the external speciﬁcation and requirements.
* **Risk resolution**: reﬂects the level of awareness and reactiveness with respect to risks.
* **Team cohesion**: it’s an indicator of how well the team members know each other and work together in a cooperative way.
* **Process Maturity**: this parameter refers to the Capability Maturity Model (CMM) attributable to the company that is going to develop the project.

Let’s consider the parameters one by one and then assign a value.

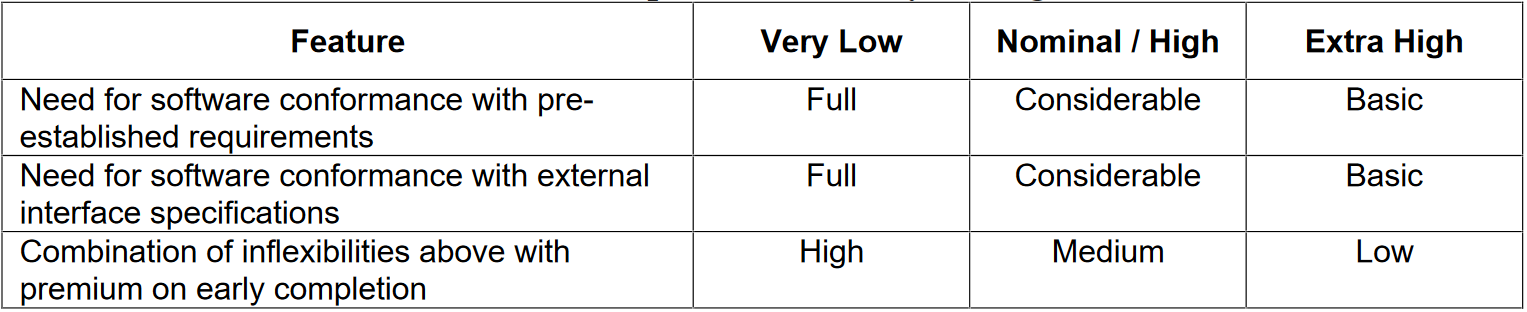
### **PRECEDENTEDNESS RATING LEVELS**

****

Choice + Rationale:

* **Organization understanding of product objectives**: Very Low since we don’t have any experience regarding past project or related projects.
* **Experience in working with related software system**: Very Low since as already stated we don’t have any experience on past similar projects.
* **Concurrent development of associated new hardware and operational procedures**: Very Low since we don’t have any control on the hardware that is going to be installed and developed.

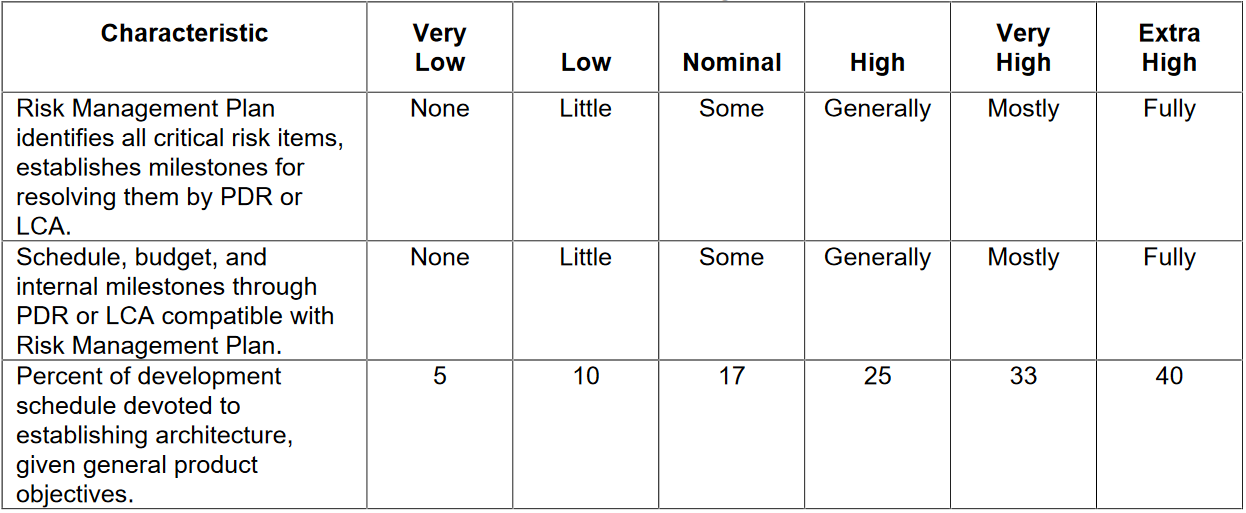
### **DEVELOPMENT FLEXIBILITY RATING LEVELS**



Choice + Rationale:

* **Need for software conformance with pre-established requirements:** Nominal / High since the customer have a clear depiction of what the project has to do but didn’t give any specific constrain and requirements that have to be fulfilled.
* **Need for software conformance with external interface specifications**: Extra High since this system is quite a standalone one, the system has interaction just with the Google Map API and the DBMS API which are not complicated constrain to fulfill.
* **Combination of inflexibilities above with premium on early completion:** Very Low since the system needs to be up and running by the time specified to the contractor’s customers because the company can start to make profit just after the completion of the project.

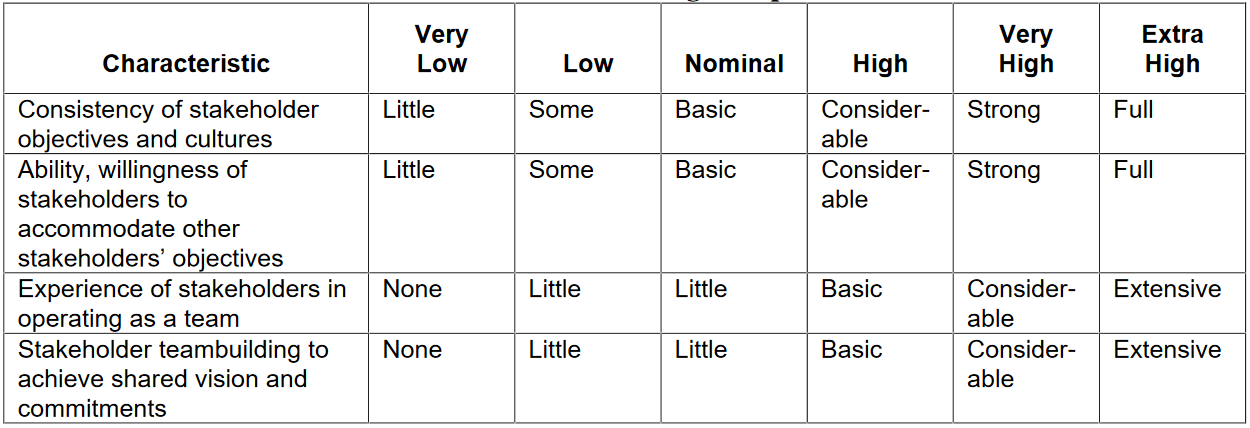
### **RISK RESOLUTION RATING LEVEL**



Choice + Rationale:

* **Risk Management Plan:** Nominal since this evaluation has been done with much theoretical knowledge but low practical one.
* **Schedule, budget, and internal milestone**: Nominal since the lack of practical experience in this compound balances with the theoretical one.
* **Percent of development schedule devoted to establish architecture, given general product objective:** Very High since in this type of project we believe in providing the documentation to define these arguments

### **TEAM COHESION RATING LEVEL**

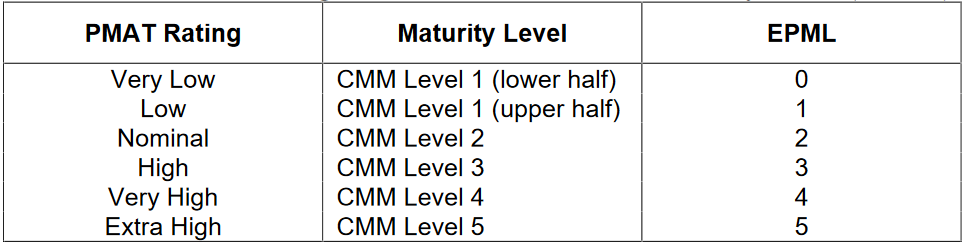
****

Choice + Rationale:

* **Consistency of stakeholder objective and cultures:** Extra High
* **Ability, willingness of stakeholders to accommodate other stakeholders’ objective**: Extra High
* **Experience of stakeholders in operating as a team:** Extra High
* **Stakeholders teambuilding to achieve shared vision and commitments** Extra High

All these parameters are classified as Extra High since the stakeholders can be considered the professors which assigned to us the project and therefore they are motivated to be cooperative in the best possible way

### **PROJECT MATURITY RATING LEVEL**



Choice + Rationale:

We cannot give a precise estimation of the CMM value of the company but one of the objectives of the project is to strengthen the process of development activities so even if helped just by the theoretical knowledge we believe that we can classify us as at least a level 2 Maturity level.

### **SCALE FACTOR RESULT**

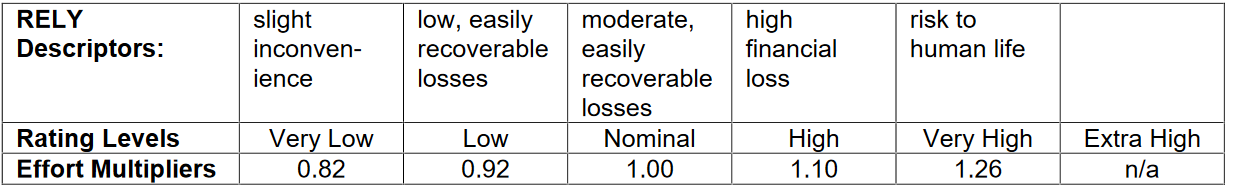
The result of the evaluation is the following:

|  |  |
| --- | --- |
| Scale Driver | Value |
| PREC FLEX RESL TEAM PMAT | 6.20  2.53  3.29  0  4.68 |
| Total | 16.7 |

### **COST DRIVERS**

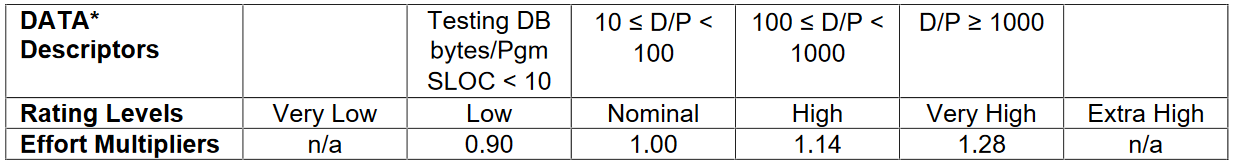
* **Required Software Reliability:**

Since the system is the only way in which PowerEnJoy makes profit a malfunction could lead to disastrous financial losses, the RELY cost driver is for this reason set to Very High.



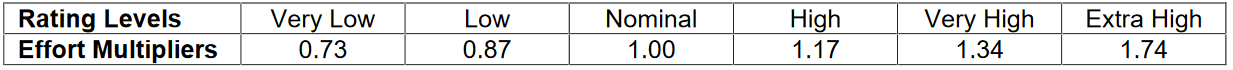
* **Database Size:**

This cost driver attempts to capture the effect large test data requirements have on product development. The rating is determined by calculating D/P, the ratio of bytes in the testing database to SLOC in the program. Since we would like to have a 5GB testing database and the SLOC are 10000 the ratio D/P is 500 setting the DATA cost driver to High.



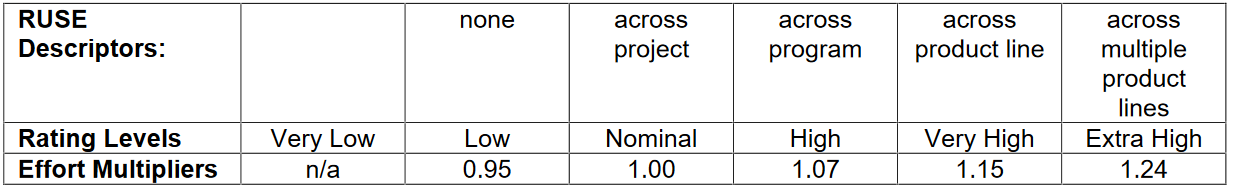
* **Product Complexity:**

Set to Very High according to the COCOMO II rating scale.



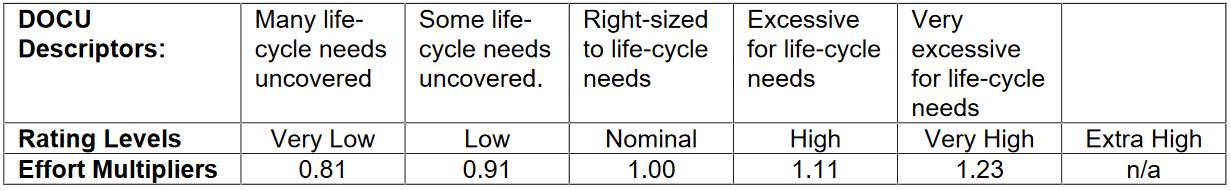
* **Required Reusability:**

Since there are no constrain set about the reusability of the project we take an average value, so we set the RUSE cost driver to Nominal.



* **Documentation Match to Life-Cycle Needs:**

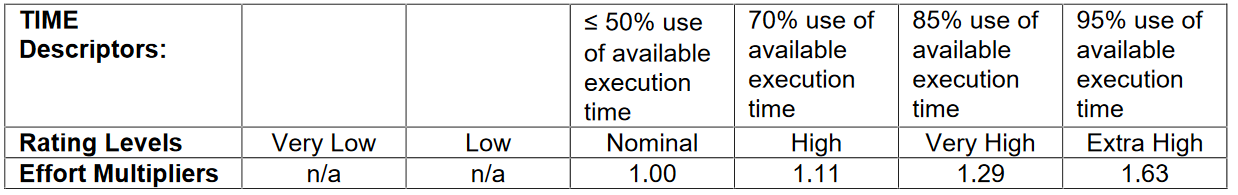
This parameter describes the relationship between the documentation and the application requirements. In our case, every need of the product life-cycle is already foreseen in the documentation, so the DOCU cost driver is set to Nominal.



* **Execution Time Constraint:**

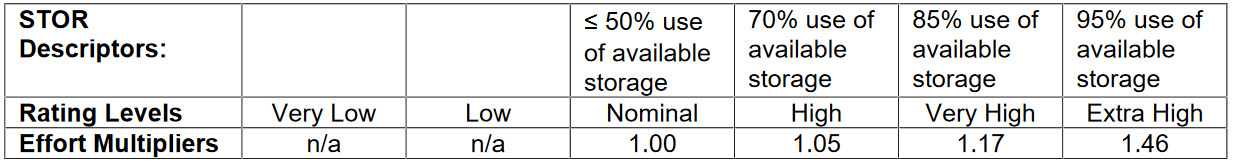
This parameter is a measure of the execution time constraint imposed upon a software system. The rating is expressed in terms of the percentage of available execution time expected to be used by the system or subsystem consuming the execution time resource.

As PowerEnJoy is a quite complex software that still, by the way has to run on a considerable number of different smartphone we set the TIME cost driver to High.



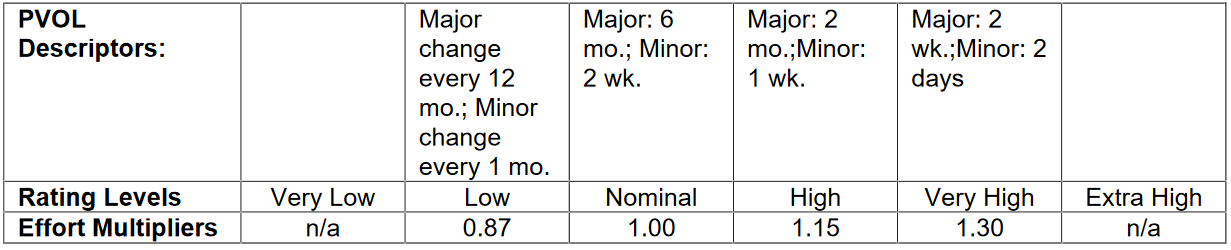
* **Storage Constraint:**

This parameter describes the expected amount of storage usage with respect to the availability of the hardware. As current disk drives can easily contain several terabytes of storage, this value is set to Nominal.



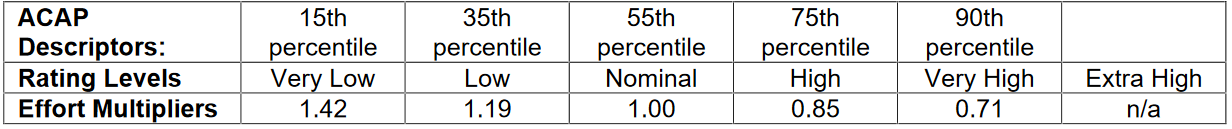
* **Platform Volatility:**

This parameter describes the expected amount of changes in the project over time. We don’t expect the business logic to change very often, but the client application may requires minor changes every month and a Major change every year so we set the PVOL cost driver to Low.



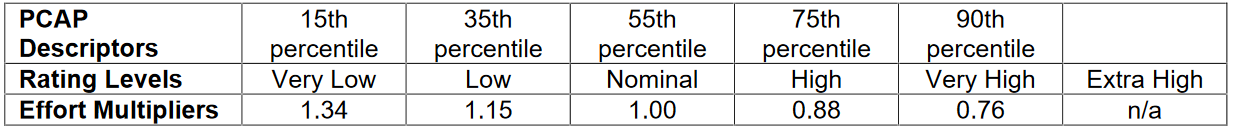
* **Analyst Capability:**

System analysis has been made in a sufficiently extensive and detailed way so we set the ACAP cost driver to High.



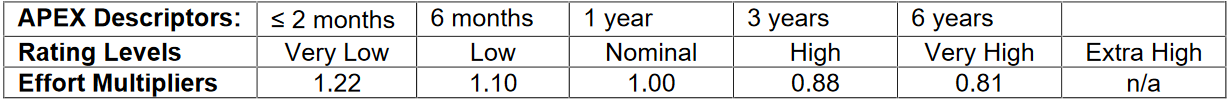
* **Programmer Capability:**

We cannot estimate very precisely this parameter, but with the theoretical knowledge and the small amount of experience in developing we fell like to evaluate the PCAP cost driver to Nominal.



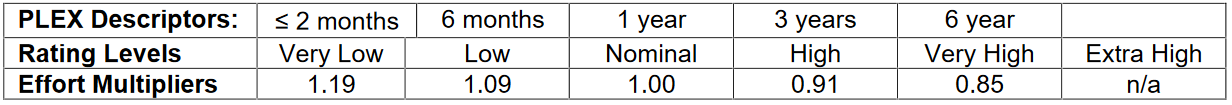
* **Application Experience:**

We have some experience in the development of Java applications, but we never tackled a Java EE system of this kind. For this reason, we’re going to set this parameter to Low.



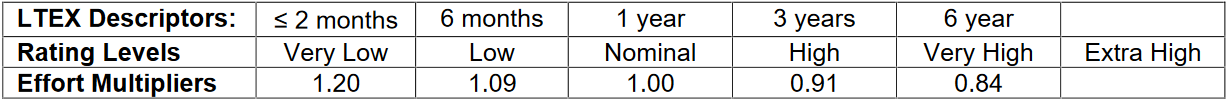
* **Platform Experience:**

We don’t have any experience with the Java EE platform, but we have some previous experience with databases, user interfaces and server side development. For this reason, we’re going to set this parameter to Nominal.



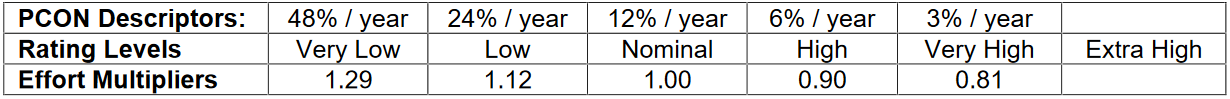
* **Language and Tool Experience:**

This is a measure of the level of programming language and software tool experience of the project team developing the software system or subsystem. A low rating is given for experience of less than 2 months. A very high rating is given for experience of 6 or more years. We cannot rely on any experience except for the one acquired during the Software Engineering course 6 month circa, for this reason we set the LTEX cost driver to Low



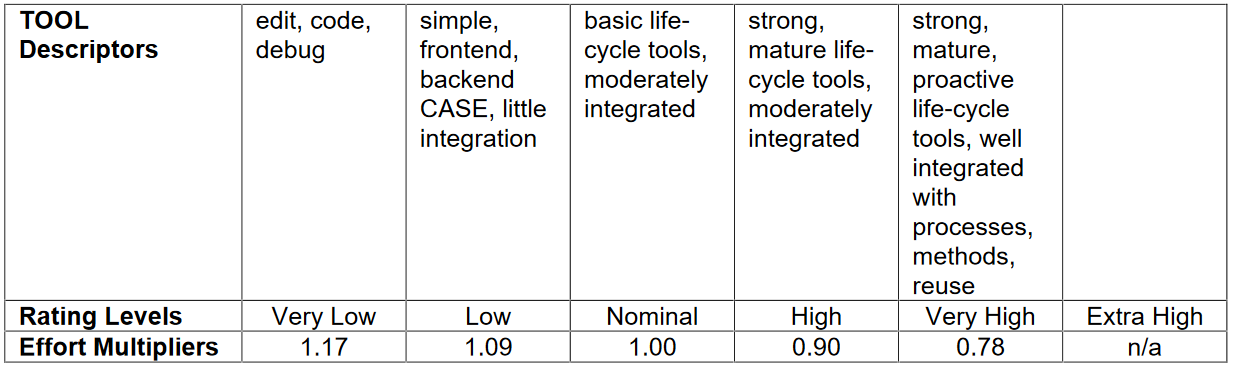
* **Personnel Continuity:**

We assume to be newly graduated engineers and then be able to work full-time on the project, and being just in two we think that it will not occur the problem of turnover, for this reason we set the PCON cost driver to High.



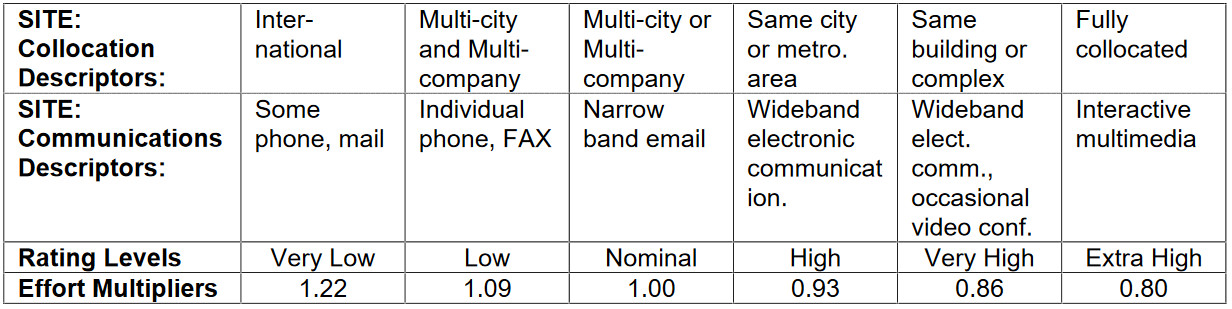
* **Usage of Software Tools:**

Our application environment is complete but not enough to consider it above average so we set the TOOL cost driver to Normal.



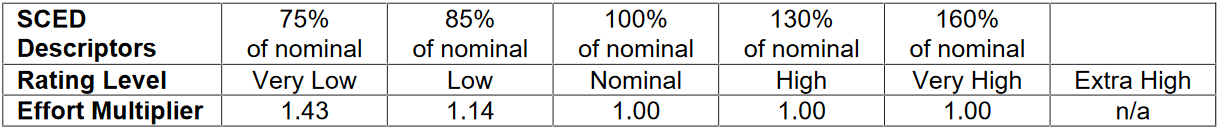
* **Multisite Development:**

Although we live in two diﬀerent cities, we have collaborated relying hugely on wideband Internet services including social networks and emails. For this reason, we’re going to set this parameter to Very High.



* **Required Development Schedule:**

We cannot make an accurate estimate of this parameter because we do not have a precise deadline for the end of the project, anyway we spent a lot of time in the early stages to define a complete and accurate structure of the project, for this reason we set the SCED cost driver to High.



The overall result is expressed in this table:

|  |  |  |
| --- | --- | --- |
| Cost Driver | Factor | Value |
| RELY  DATA CPLX RUSE DOCU TIME STOR PVOL ACAP PCAP APEX PLEX LTEX PCON TOOL SITE SCED | Very High  High  Nominal  Nominal  Nominal  High  Nominal  Low  High  Nominal  Low  Nominal  Low  High  Nominal  Very High  High | 1.26  1.14  1.00  1.00  1.00  1.11  1.00  0.87  0.85  1.00  1.10  1.00  1.09  0.90  1.00  0.86  1.00 |
| Total | | 1.0942 |

### **EFFORT EQUATION**

This ﬁnal equation gives us the eﬀort estimation measured in Person-Months (PM):

Where:

A = is a constant which value is 2.94 (for COCOMO II)

EAF = product of all cost driver (1.0942)

E = Scale Factor computed as:

B is equal to 0.91 by definition for COCOMO II

With these parameters, we can compute the eﬀort value:

44.2648 PM

### **SCHEDULE ESTIMATION**

Regarding the final schedule, we are going to use the following formula:

Where:

C = 3.67 by definition

D = 0.28 by definition

0.3154

We calculate so the average staffing needed to complete the project within this duration

The result seems to be a reasonable estimation for a medium size project.

## **SCHEDULE**

We are now going to present the high-level schedule of our project. This schedule represent just a track of the entire development process, as a more detailed schedule for each phase of the process will be given during the project itself.

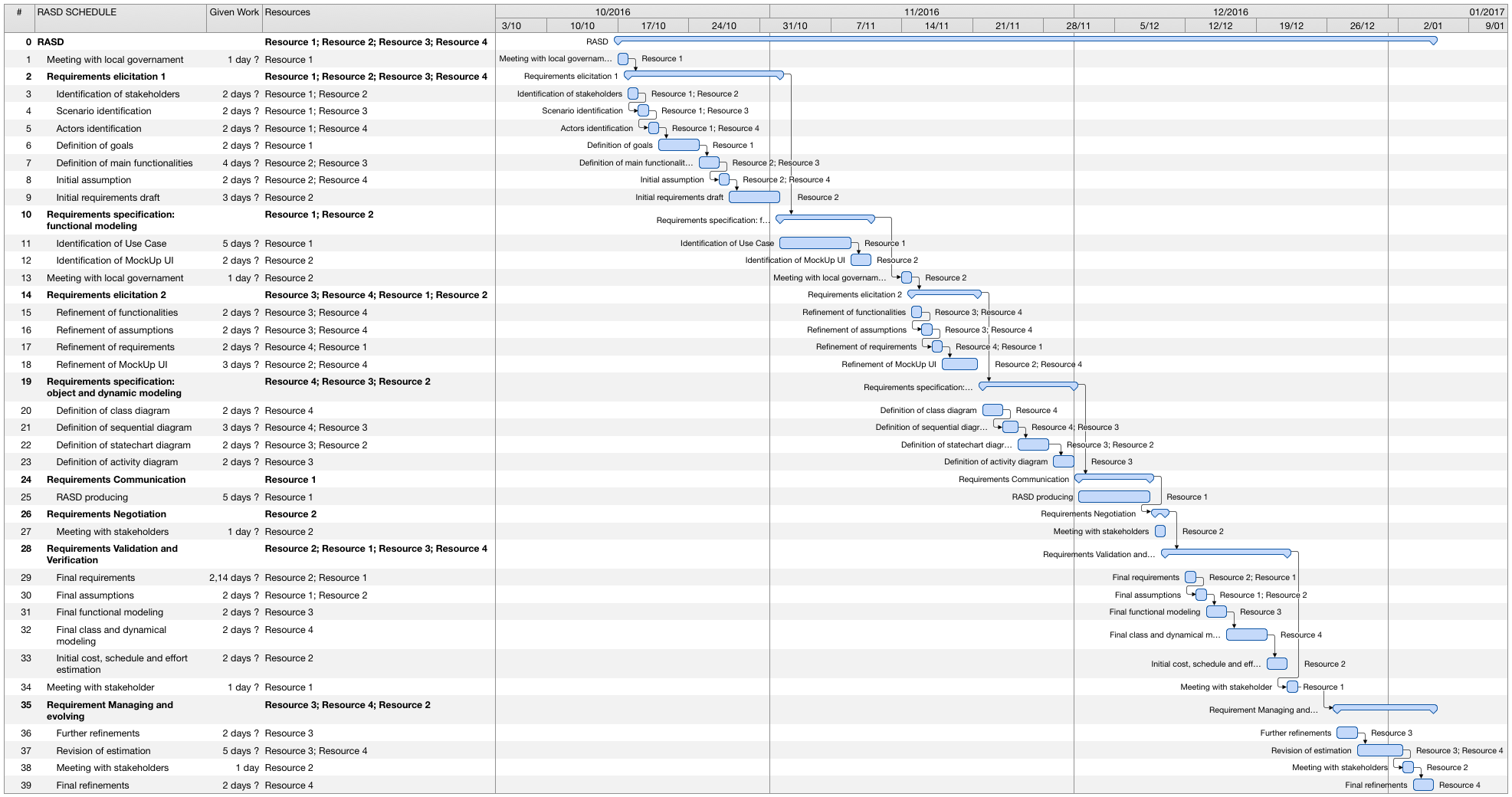
This schedule is divided into four parts. The first one is related to the schedule of the requirements phase, the second one in related to the schedule of the design phase, the third one is related to the development phase and the last one to the deployment and start-up. This separation has been done for readability reason, though the schedules are linked together: in fact, for instance, the DD’s start date corresponds with the RASD’s end date, and so do the Development and the deployment.

The Development phase has been planned according to the cost and effort estimation presented in the previous part of this document. Further details about this will be given later.

We have considered as start date of our project the 16 of October 2016.

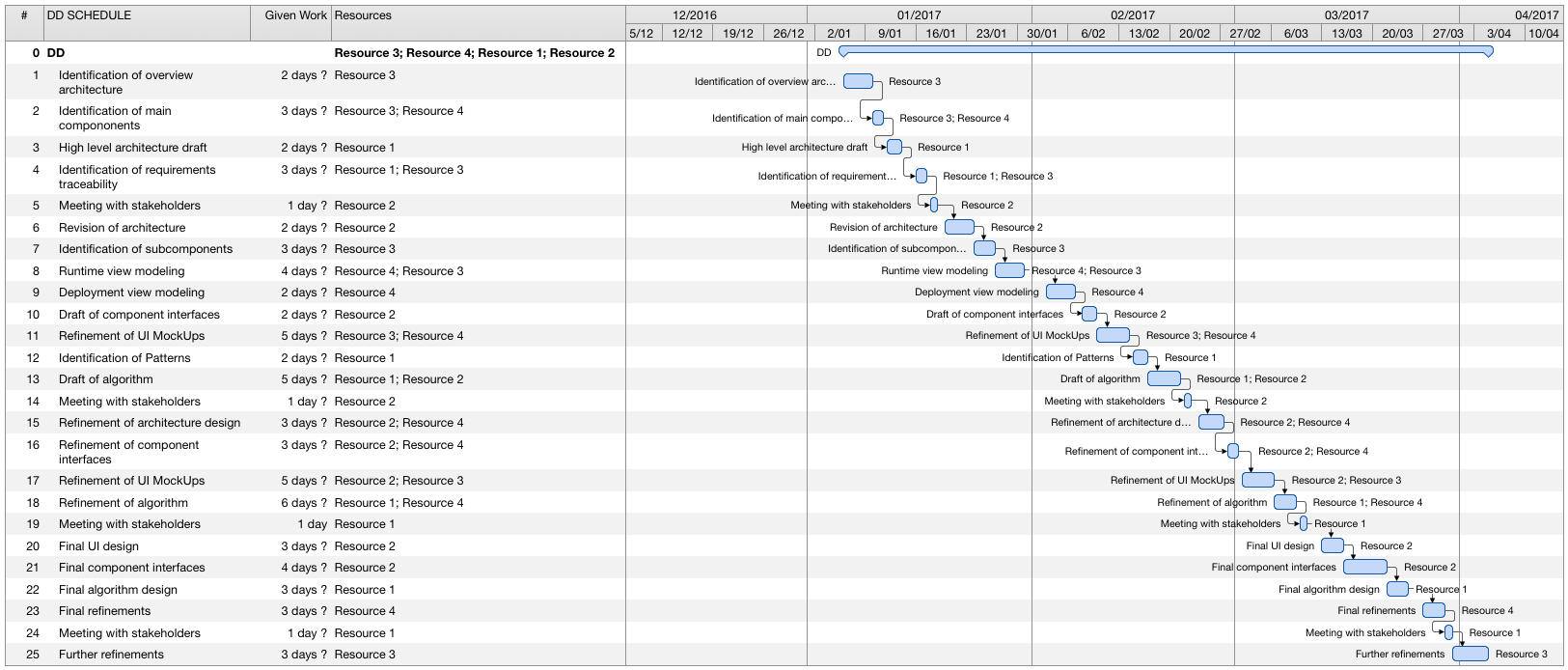
* 1. **RASD SCHEDULE**

This first schedule is related to the requirements phase and finishes with the production of the RASD. It considers as starting date the 16 October 2016 and as ending date the 5 January 2017. For this phase has been scheduled a total of 3, 56 PM and, as this first part has to be performed almost sequentially, the duration of this phase is 2,92 months.

****

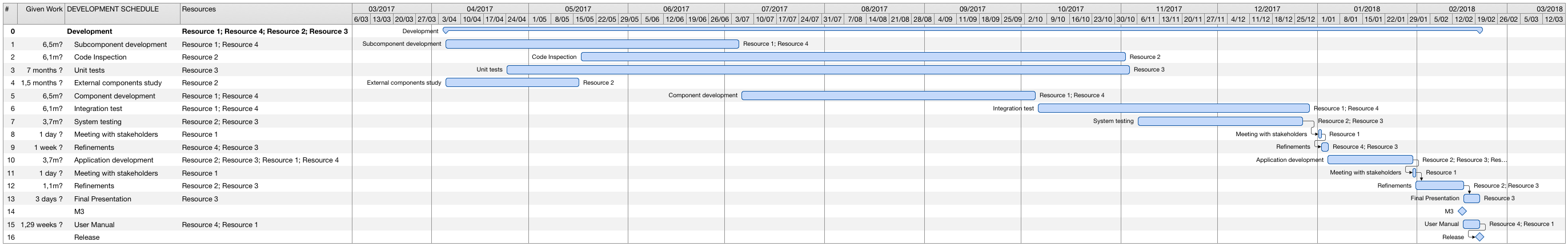
* 1. **DD SCHEDULE**

The second schedule is related to the production of the DD. It considers as starting date the 27 January 2017 and as ending date the 4 April 2017. As for the RASD schedule, also this phase has to be performed almost sequentially and so, as it consider a total of 3,6 PM, it lasts 3,15 month.



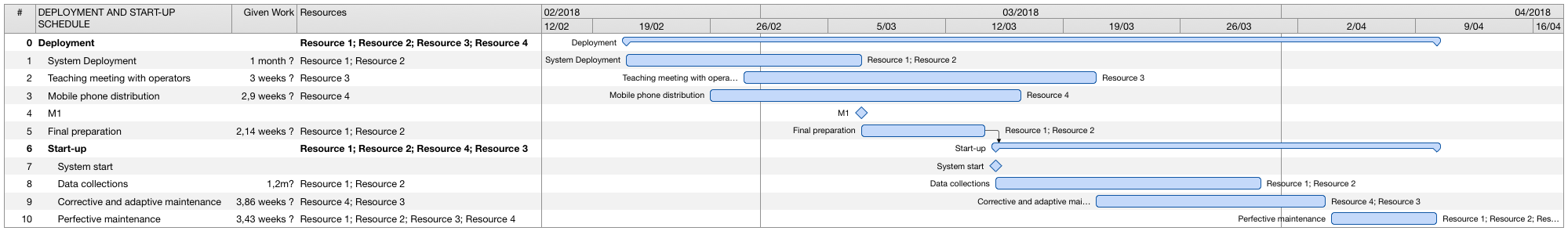
* 1. **DEVELOPMENT SCHEDULE**

The third schedule deals with the development phase. This part has been projected according to the PM calculated using COCOMO II. As we calculated and average of 45 PM distributed over 12,19 months we have tried to schedule this part using a number of PM around 45distributed over a 12 or 13 months. In fact, we have scheduled 43,08 PM with a duration of 12,48 months. The schedule considers as starting date the 5 April 2017 and as ending date the 20 February 2018. Most of the tasks of this schedule can be performed in parallel, this will be considered for the resource allocation.

****

* 1. **DEPLOYMENT AND STARTUP SCHEDULE**

This last schedule deals with the deployment phase and the start-up phase. The schedule considers as starting date for the deployment the 21 February 2018 and as ending date the 15 of March 2018 with the final preparations. The Start-up phase is concluded then the 10 April 2018.

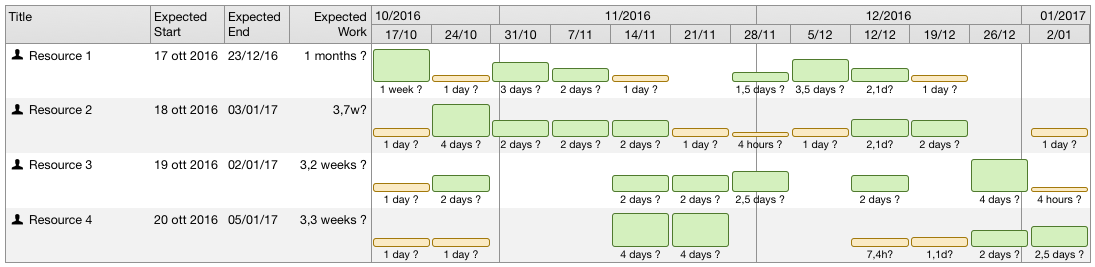


## **RESOURCE ALLOCATION**

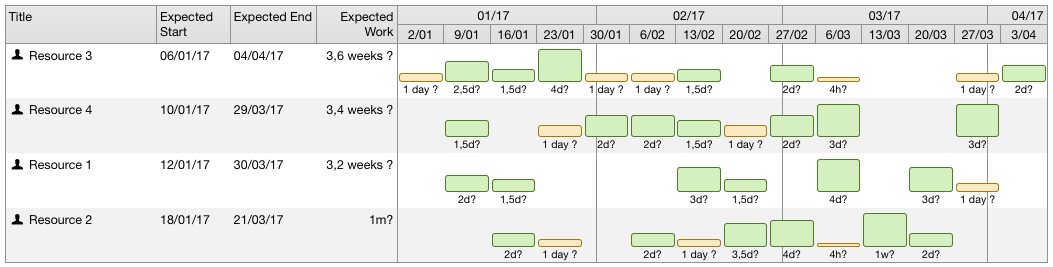
In this section we are going to show how the different tasks defined in the schedule can be divided between the members of the group. The schedule that has been presented in the previous section takes into account the estimation done with COCOMO II. However, that estimation calculates a duration (12,19 months) for the development phase suggesting a number of 4 people working at it. So, in order to respect the schedule presented and in order to deal with more realistic situation, we have to consider for our project 4 members. The resource allocation presented is just a general overview, more refined schedule will be defined during the project.

In the previous section, the schedule presented already contains the resource allocation according to the consideration done right now. So, in order to refer to the single task of each resource we can just give a look to the previous schedule. To summarize and to understand how the work is divided into the members of the groups we present the schedule of the hours of work per week per each members of the group.

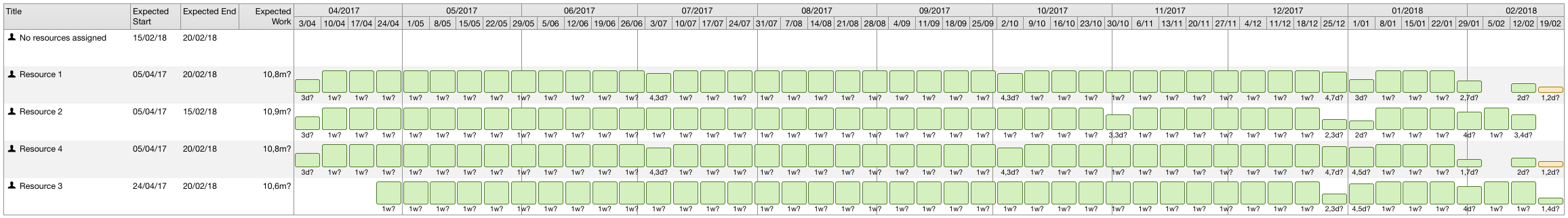
* 1. **RASD RESOURCE ALLOCATION**

****

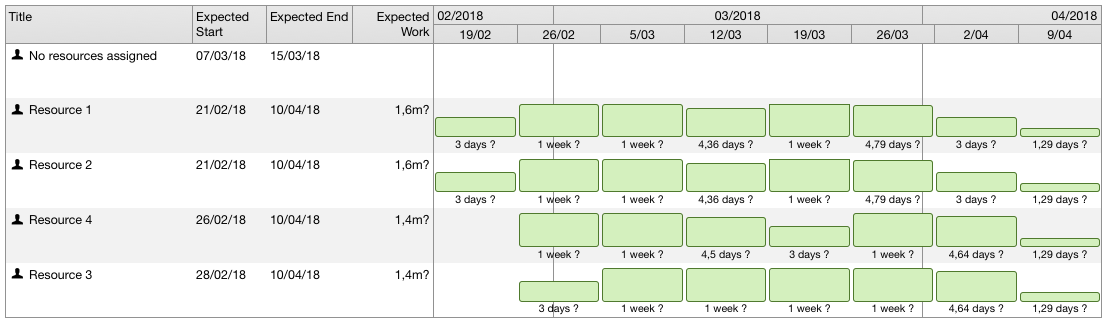
* 1. **DD RESOURCE ALLOCATION**

****

* 1. **DEVELOPMENT RESOURCE ALLOCATION**

****

* 1. **DEPLOYMENT AND START-UP RESOURCE ALLOCATION**

****

## **RISK MANAGEMENT**

In this section we are going to identify the main risks that the project development may face and a way to del with them. Risks could be categorized in different way: they could be project risks (they threaten the project plan); they could be technical risks (they threaten the quality and the timeliness of the software to be produced); they could be business risks (they threaten the viability of the software to be built); they could be political or financial risks (they threaten with local government city administration and legislation). Our intent is to adopt a proactive risk strategy, that means, firstly identifying the possible risks and analysing each risks to estimate the probability that it will occur and the impact that it will do if it does occur and then developing a contingency plan to manage those risks.

The main problems we could face with are that dealing with stakeholders. In our case, one of our stakeholders is the city government, in particular some members of the municipal council responsible for mobility and environment, and they have a strong influence on the acceptance or refusal of our software product. Potential risks can include a change in the city government or a shift in the local government priorities for different reasons. To face this risks one solution could is to give an active role to the stakeholders in the development of the project, especially in the requirements analysis part and in the design phase. To do this many meetings with stakeholders are planned and also some periodical reviews and discussion. In the RASD schedule also a negotiation part is inserted, because we could face with the risks of putting together the desires of stakeholders with market choice.

Another risks always related to political risks are the laws related to security and their future changes. Of course our product must deal with this and changes in the laws could influence the requirements or the design phase. To face with this risk we could keep monitored the discussions related to this laws and, if some changes are proposed, prepare another plan before the laws being approved.

Together with the political and financial risks, the other main problems we could encounter are the project risks. In particular issues arising from people management. First of all we have to consider that people may get ill. There are some periods in which this would not be a big problem but other, near to milestones or meetings, could cause delays and changes in the project schedule. We have to consider also the possibility of people quitting the company. To face with this problem we have allocated, whenever possible, different resources to the same task so that a single person is not in charge of a specific task.

Another risk related to project risk is the fact of overestimate the ability of a specific resource. This could be avoided hiring selected people with experience. Of course another solution is to add people to the project, though this wouldn’t have to be the first solution, as the tasks are extremely specific.

Finally, as far as project risk is concerned, another problem could be the fact that the project schedule will not be respected due to further issues that may arise and not preventively scheduled. To avoid this in the schedule have been considered time for revision and check.

As for business risk, the main problem that may arise concern market risks and sales risks, that is, building a product that no one really wants or that the sales force doesn’t understand how to sell. Another risk of this kind is the fact that the customer wouldn’t accept the system. To avoid this it’s necessary to encourage sales company to meet with the software developer in order to well plan the market strategy. In this way we can win the support of the majority of the customer.

Finally, as far as technical risks is concerned, we have to consider first of all that our software depend many on external components and devices such as the database, the Google Map service, the car with monitor and sensors and the bank that manages the payments. In fact, for instance, the database could not be able to process as many transactions per second as expected, the Google Map service could change (together with theirs API) and the car components could not have an appropriate hardware. To deal with this problems we have to replace potentially defective components with components of known reliability and we have to design the code to be as portable as possible and with a great modularity and independence between components, exploiting the information hiding principle as much as possible.

## **EFFORT SPENT**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Time spent [hours]** | |  |
| **Activities** | **Emanuele Chilà** | **Giorgio Lazzarinetti** | **Total** |
| Introduction | 1 | 1 | 2 |
| Project cost and effort estimation | 12 | 3 | 15 |
| Schedule | 1 | 6 | 7 |
| Resource assignment | 1 | 3 | 4 |
| Risk Management | 1 | 3 | 4 |
| Overall document formatting | 1 | 1 | 2 |
|  | **Total ITPD** | | 34 |

## 

## 8**. REFERENCES**

The used tool used to create the ITPD are:

* Merlin Project: to write the schedule and the resource allocation diagram;
* Microsoft Word 2011: to write the document;