

***“PowerEnjoy”***

*Project Management Document*

**Version 1.0** *(11/12/2016)*

Giorgio Marzorati (876546)

Aniel Rossi (877018)

Andrea Vaghi (877710)

INDEX OF CONTENTS

INTRODUCTION 3

Purpose 3

Scope 3

Reference Documents 5

Document Structure 5

PROJECT SIZE, COST AND EFFORT ESTIMATION 6

Size estimation: function points 6

Internal Logic Files (ILFs) 6

External Logic Files (ELFs) 7

External Inputs (EIs) 7

External Inquiries (EQs) 7

External Outputs (EOs) 7

Overall estimation 8

COST AND EFFORT ESTIMATION: COCOMO II 8

Scale Drivers 8

Cost Drivers 8

Schedule estimation 9

Effort equation 9

Schedule estimation 9

SCHEDULE 10

RESOURCE ALLOCATION 10

RISK MANAGEMENT 11

EFFORT SPENT 12

Giorgio Marzorati 12

Aniel Rossi 12

Andrea Vaghi 12

CHANGELOG 13

# **INTRODUCTION**

## Purpose & Scope

In this document, named Project Plan Document (PPD), we make a deep analysis about the complexity of the project in order to try to make the most real and feasible estimation of the costs and the effort involved in the entire project. The main focus of this analysis is to estimate the required budget needed, the schedule of the activities and the allocation of the resources in order to cover all the duties.

In the first part of the document we use two techniques of size estimation (Function Points and COCOMO II) that help us to have an idea respectively on the final number of lines of code(SLOC) of the software and on the relation between the cost and the effort required to complete all the identified activities coherently with our planning.

In the second part we propose a schedule of the identified activities distributed along the period of time selected for the realization of the entire project, from the analysis of requirements to the final deployment and start up.

In the third part we assign each activity to one or more members of the team, based on the effort that we want to concentrate in them and the criticity that they represent respect to the entire project.

In the last part we make an evaluation of the risks in which our project could be involved. This analysis takes into account three main categories of risks:

* Project risks
* Technical risks
* Business risks

After the evaluation, we propose for each identified risk a mitigation strategy in order to handle the situation when the possibility of that risk becomes reality.

**Definitions, Acronyms, Abbreviations**

• FP: Function Points.

• ILF: Internal logic file

• ELF: External logic file.

• EI: External Input.

• EO: External Output.

• EQ: External Inquiries.

• DBMS: Database Management System.

• API: Application Programming Interface.

• ETA: Estimated Time of Arrival.

• UI: User Interface.

• GPS: Global Positioning System.

## Reference Documents

• PowerEnjoy Requirement Analysis and Specification Document: RASD.pdf

• The Project Plan Example documents: Project Planning Example Document.pdf and Project Management Basics + Advanced.pdf

• The Function Points complexity evaluation tables.

• The COCOMO II Model Definition Manual (version 2.1, 1995 – 2000 Center for Software Engineering, USC).

# PROJECT SIZE, COST AND EFFORT ESTIMATION

In the next section we will at first present a valuation of the expected size of the PowerEnjoy project (expressed in SLOC), then an estimate of the cost and the required effort, using the following approaches:

* Function Points for the size estimation, based on each functionality that the business logic of the application has to offer and the corresponding lines of code to be written in order to correctly implement it;
* COCOMO II approach for the cost and the required effort, taking into account that we find ourselves into a Post-Architecture case of analysis

## Size estimation: function points

### Internal Logic Files (ILFs)

In this paragraph we will describe the Internal Logic Files on which the PowerEnjoy data structure is based on in order to be able to implement all the functionalities it offers. We will focus in particular in the description of their complexity, to justify the amount of FPs we decided to apply to every single ILF.

ILF Rating:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Data Elements** | | |
| **Record Elements** | 1-19 | 20-50 | 51+ |
| 1 | Low | Low | Avg |
| 2-5 | Low | Avg | High |
| 6+ | Avg | High | High |

1) Client

Attributes:

* Driving License id
* Email
* PIN
* Name
* Surname
* Payment Method

### External Interface Files (EIFs)

In this section we will explain the interaction between PowerEnjoy and some sets of data provided by other independent applications/providers. In particular our system relies on two main external services, the *Mapping Service* and the *Payment Service Provider*. The information retrieved through the interaction between the system and these components sometimes need a certain amount of analysis and processing in order to be useful in our system. There may be situations in which the complexity of some of those interactions needs to be considered in a proper way, taking into account the dimension of the data exchanged and the amount of business logic required to process it. The list of the interactions is the following:

**Mapping Service:**

* City Map Retrieval:

This information is requested for the displaying of the city map on the user’s mobile application. The operation of retrieving is quite simple, because it requires just the user position or an address, but the amount of data returned is high and needs some logic to be forwarded to the client. Thus, we decided to assign a high complexity value

* Address - Position Conversion Result

The Mapping Service is able to convert an address provided in String format to a pair of coordinates that will be used to update a single table on the database for each request, so the complexity of this EIF is low

* Money Saving Mode Data (secondo me va high):

Money saving mode data is a collection of information that the user gets when activates that mode (ETA, Distance to the destination, Information about the path and so on). Although this data are computed by the mapping service, we need some logic operation in order to provide to the external service the information it requires (two pairs of coordinates, one for the user actual position and one for the Special Parking Area). Specifically, the choice of the second one is based on the final destination of the user, the availability of Special Parking Areas and the distribution of cars, so to jump to a conclusion we need several database queries, distance computation and comparisons. Taking into account these considerations, we decided to assign a medium (high) grade of complexity

* Distance Retrieval:

The distance between two pairs of coordinates is a simple data expressed with a double precision floating point value, computed by the Mapping Service. As inputs, that external component needs only the coordinates representing the start point and the end point. Since we only need to provide this kind of information (that we mainly store in the database) and we don’t have to apply any logic to the returning data (maybe just some formatting operation), we classify this operation with a low complexity

**Payment Service Provider:**

* Transaction Acknowledgement:

The only information we receive from the Payment Service Provider is an acknowledgment for the outcome of a user payment for a ride or a fee for an expired reservation. Since we store the payments in a single ILF, all we have to do with the data incoming from the Payment Service Provider is an update on a single field of a table on the database, we assume that the complexity of this operation can be evaluated as low

Overall point assignment:

|  |  |  |
| --- | --- | --- |
| **EIF** | **Complexity** | **FP** |
| City Map Retrieval | High | 10 |
| Address/Position Convertion | Low | 5 |
| Money Saving Mode Data | High | 10 |
| Distance Retrieval | Low | 5 |
| Transaction Ack | Low | 5 |
| *Total* | | 35 |

### External Inputs (EIs)

The assistance team handles every help request, as well as cars malfunctions and recharge issues. This information is provided to the assistance team by the central server (except for client help requests made directly with phone call), and the team has also the responsibility of storing and managing data retrieved by clients about the specific issue. In order to achieve that goal, assistance team’s terminals are equipped with a dedicated software for central server interaction.

The last component the central server interacts with is the Database system, with which communicate synchronously to store and retrieve the whole managed data.

### External Inquiries (EQs)

The assistance team handles every help request, as well as cars malfunctions and recharge issues. This information is provided to the assistance team by the central server (except for client help requests made directly with phone call), and the team has also the responsibility of storing and managing data retrieved by clients about the specific issue. In order to achieve that goal, assistance team’s terminals are equipped with a dedicated software for central server interaction.

The last component the central server interacts with is the Database system, with which communicate synchronously to store and retrieve the whole managed data.

### External Outputs (EOs)

The assistance team handles every help request, as well as cars malfunctions and recharge issues. This information is provided to the assistance team by the central server (except for client help requests made directly with phone call), and the team has also the responsibility of storing and managing data retrieved by clients about the specific issue. In order to achieve that goal, assistance team’s terminals are equipped with a dedicated software for central server interaction.

The last component the central server interacts with is the Database system, with which communicate synchronously to store and retrieve the whole managed data.

### Overall estimation

The assistance team handles every help request, as well as cars malfunctions and recharge issues. This information is provided to the assistance team by the central server (except for client help requests made directly with phone call), and the team has also the responsibility of storing and managing data retrieved by clients about the specific issue. In order to achieve that goal, assistance team’s terminals are equipped with a dedicated software for central server interaction.

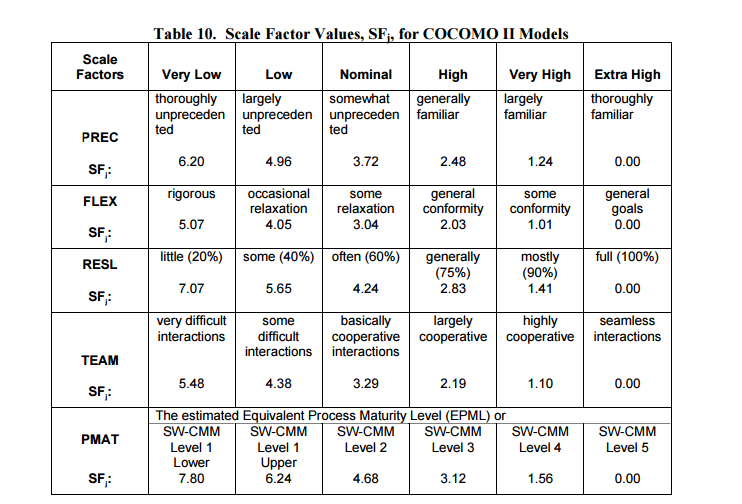
The last component the central server interacts with is the Database system, with which communicate synchronously to store and retrieve the whole managed data.

## COST AND EFFORT ESTIMATION: COCOMO II

In this section we’re going to use the second estimation method mentioned in the introduction, COCOMO II. The purpose is to make an analysis of the costs and the effort needed in order to be able to build the project in terms of time and money.

### Scale Drivers

The first thing to do is to evaluate some scale factors, making a sort of highlighting of some of the main properties of the project and the team. In order do that we refer to the following official COCOMO II table:



A brief description for each scale driver:

• Precedentness: it reflects the previous experience of our team with the development of large scale projects. In our case the value is low, our team made together only one another large scale project except this.

• Development flexibility: it reflects the degree of flexibility in the development process with respect to the external specification and requirements. Since we don’t have any particular constraint (especially technical) specified in the specifications the flexibility is very high.

• Risk resolution: reflects the level of awareness and reactiveness with respect to risks. The risk analysis we performed is quite extensive, evaluating both internal and external risks, dividing them also in categories. The value is set to high.

• Team cohesion: it’s an indicator of how well the team members know each other and work together in a cooperative way. Here the value is high due to the fact that the we know each other very well but we live in different towns, so sometimes it can be tricky to find the time to meet each other.

• Process maturity: due to the fact that we work on a well defined development process this value is set to nominal.

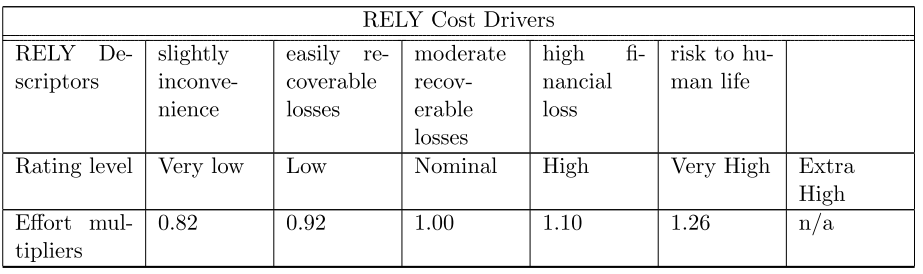
The results of our evaluation is the following:

|  |  |  |
| --- | --- | --- |
| **Scale Driver** | **Factor** | **Values** |
| Precedentness | Nominal | 3.72 |
| Development  Flexibility | Very High | 1.01 |
| Risk Resolution | High | 2.83 |
| Team Cohesion | High | 2.19 |
| Process Maturity | Nominal | 4.68 |
| *Total* | | 14.43 |

### Cost Drivers

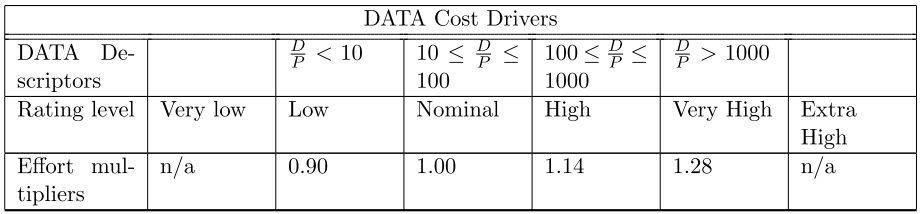
*Reliability*:

A possible shutdown of the server in a particular amount of time leads only to the loss of the payments of the rides that are happening in that moment and the impossibility to reserve or identify a car. In the case of missed payments we can easily recover the rides that weren’t payed from the db. In our case the value is set to low.



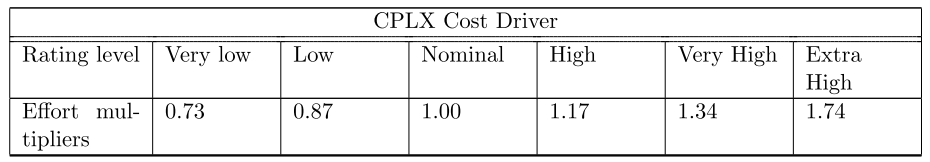
*Database Size*:

This measure considers the effective size of our database. We don’t have the ultimate answer, but our estimation given the tables and fields we have is to reach a 3GB database. Since it is distributed over 10.000-15.000 SLOC, the ratio D/P (measured as testing DB bytes/program SLOC) is between 209 and 314, resulting in the DATA cost driver being high.



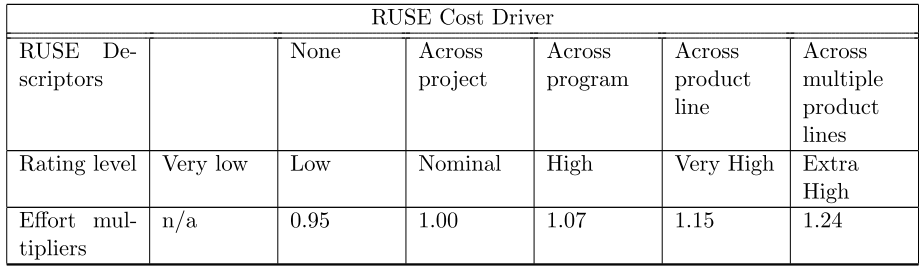
*Product Complexity*:

Normal according to the COCOMO II Standard.



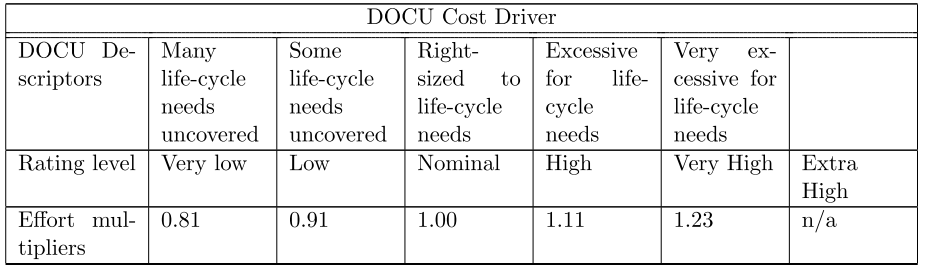
*Required Reusability*:

This cost driver accounts for the additional effort needed to construct components intended for reuse on current or future projects. In our case, the reusability requirements are limited in scope to the project itself, so the RUSE cost driver is set to nominal.



*Documentation match to life-cycle needs*:

This parameter describes the relationship between the documentation and the application requirements. The rating scale goes from Very Low (many life-cycle needs uncovered) to Very High (very excessive for life-cycle needs). In our case the value is Normal, due to the fact that in the documentation are covered the main life-cycle needs, so the document descriptors can be considered right-sized to life-cycle needs.

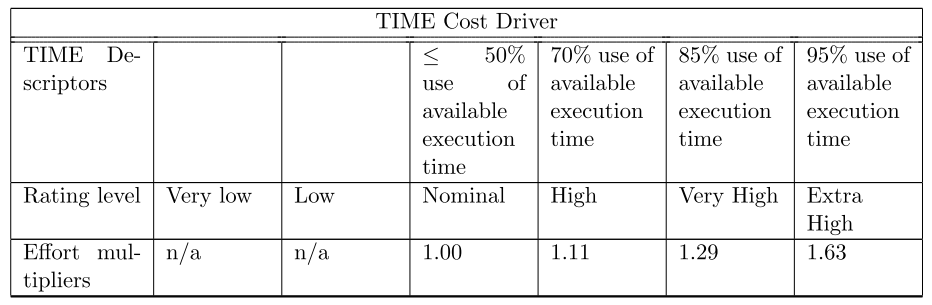


### Platform factors

These types of factors refer to the complexity of the hardware talking about the machine level.

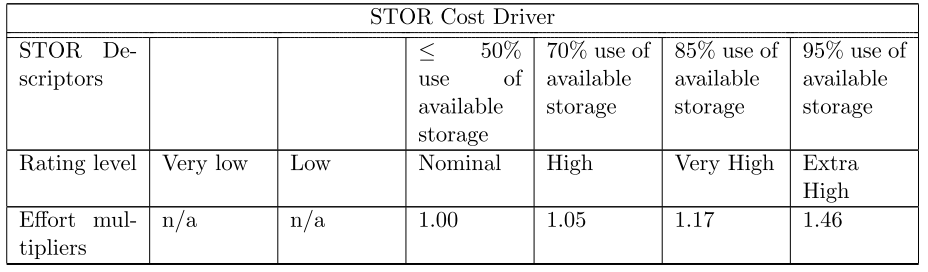
*Execution Time Constraint*

This is the measure of the execution time constraint imposed upon a software system. Since we do not have many accesses in the same moment (estimating a number of user of 10000), we are not going to handle too much parallel operations and these operations are going to be simple one. For these reasons, we choose the value high of the table corresponding to the rate 1.11. The rating express the percentage of the available execution time expected to be used by the system or subsystem consuming the execution time resource.



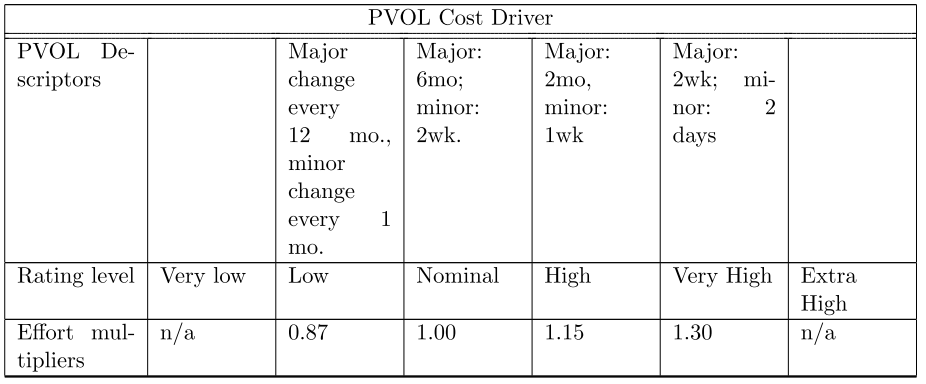
*Main storage constraint*

This rating represent the degree of the main storage constraint imposed on a software system or subsystem. Because of the low cost and the high capacity of the disk on the market, we get a nominal parameter with value 1.00.



*Platform volatility*

In this section with platform we are going to mean the complex of hardware and software (form OS to DBMS) that the software has to call to perform its tasks. Since we would like to be aligned with all the software products on which we rely on (car software, payment method, OS client), but there are in a small number of them we can imagine to have not so many changes during the time, for that reasons we get a normal parameter (1.00).

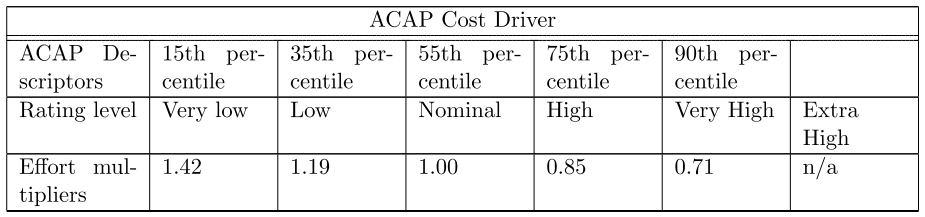


### Personnel factors

People factors have the strongest influence in determining the amount of effort required to develop a software product and they rate the development team’s capability and experience.

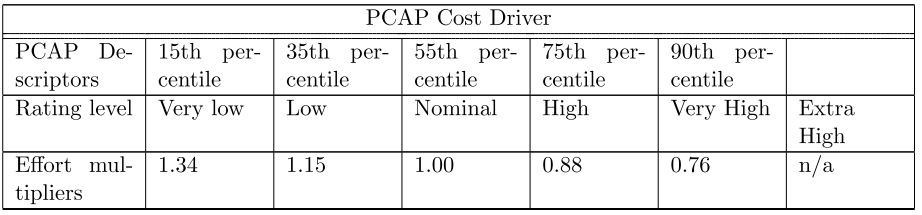
*Analyst capability*

This value represent the design ability and efficiency, the ability to communicate and cooperate of the analyst, personnel who work on requirements and high-level design and detailed design. Since we spent a lot of time analysing the requirements deeply and building a solid system during the requirement analysis and the design phase we could think that we are going to have less problem in the future phase for the development so we choose the high value with rate 0.85



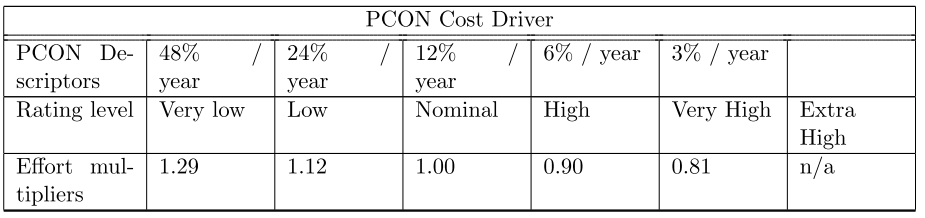
*Programmer capability*

Since with this value we represent the ability, the efficiency and the thoroughness of our team of programmers we get a high value (0.88) because we think we are a good and efficient team, since we had some previous experiences as a team.



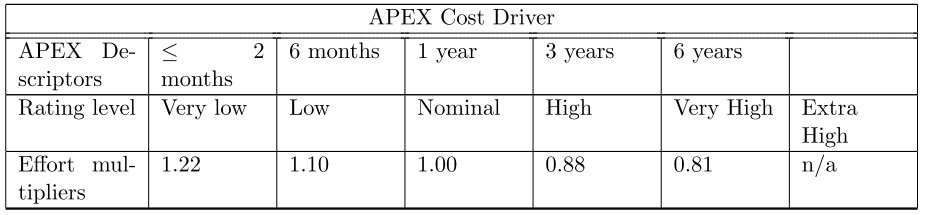
*Personnel continuity*

Considering our period of work and development of the project, we were almost all present during every meeting so we choose the high value 0.90.



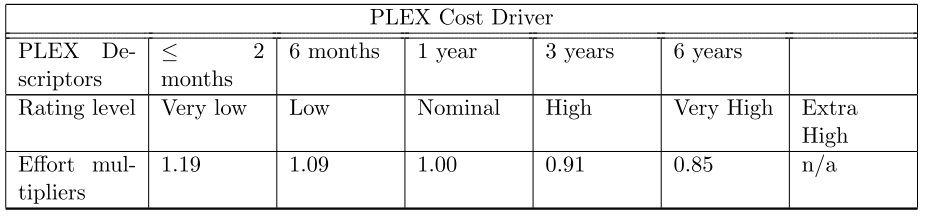
*Application experience*

This rating is dependent on the level of applications experience of the project team developing the software system or subsystem in term year of experience. We choose a normal value 1.00 because we did some project before this one.



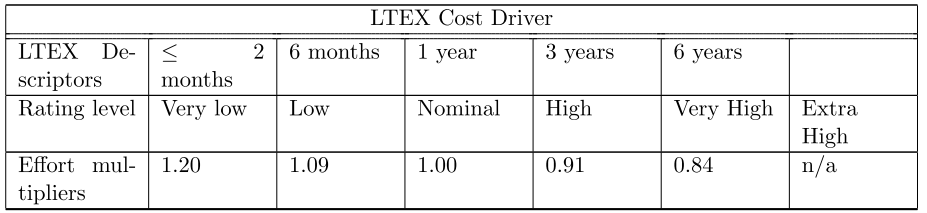
*Platform experience*

With this value, we consider the platform (graphic user interface, database, etc.) and knowledge experience. Because of our past project involving databases, graphic user interfaces, we would choose a normal value, but because of our non-deep knowledge of these platforms we prefer the low value (1.09)



*Language and tool experience*

This rating is dependent on the level of programming language and software tool experience of our team, including use of tool in requirements and design phases, configuration analysis, document extraction and so on. Because of our average knowledge, we get the normal value 1.00.

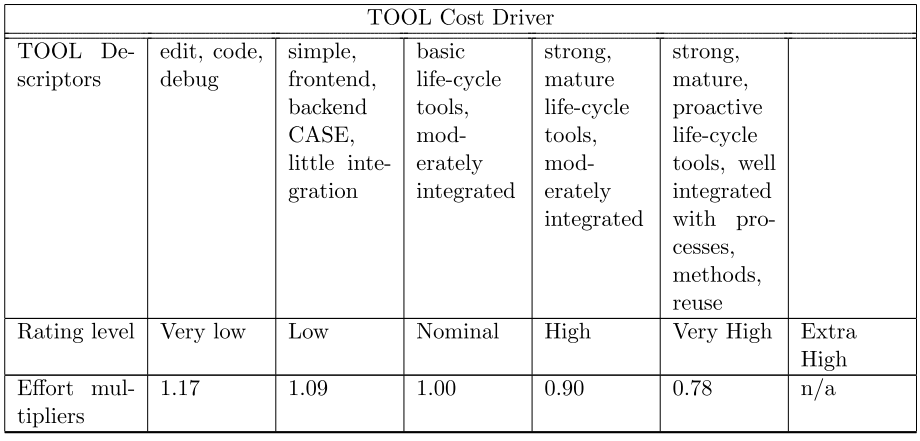


### Project Factors

Account for influences from the use of modern software tools, compression of the project schedule or location of development team.

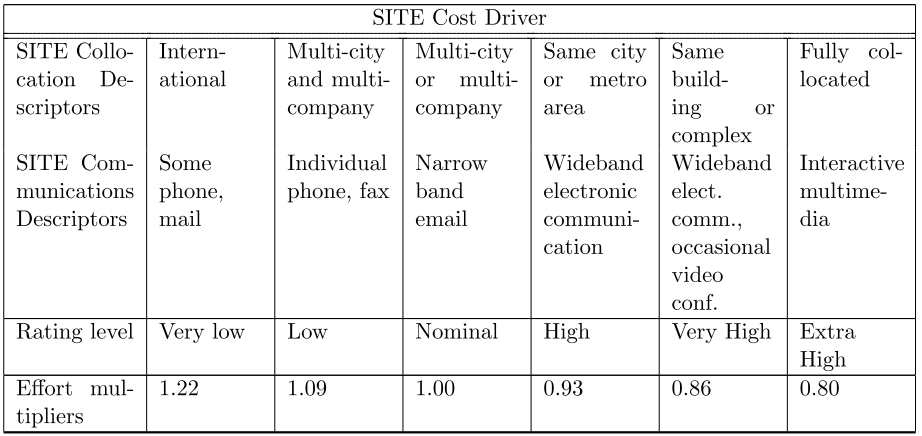
*Use of software tools*

It reflects the tool capability, maturity and integration. Because we are going to use solid tools and we are not going to use open source tools, which could create some problems, we choose the high value 0.90.



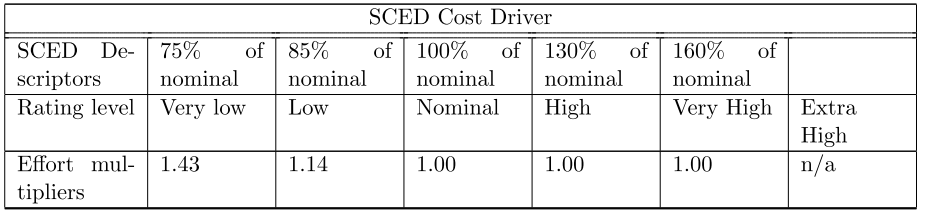
*Multisite development*

Since we built our system entirely during our meetings and we did not use multisite developments we get the very high value 0.86. This means that we are almost on the same building.



*Required development schedule*

This rating measure the constraint imposed on the project team developing the software in terms of percentage of schedule stretch-out or acceleration with respect to the nominal schedule. Because of our hard work during requirements and design phases we think that the right value is the high one 1.00.



## Schedule and resource allocation

During this chapter we will provide a general scheduling for the development of the project taking into account the period of time we had to develop this project, for that reason the period starts in October and finishes in March. Even if the implementation part of the project is not requested, we included the development, deployment and start-up phases in order to make it as complete as possible.

In this schema we also include the part of resource allocation dividing each tasks on the members of the team, adding some of them like meeting with service car software and with stakeholder. Those were added trying to reach the completeness and realistic for the scheduling part of the project, even if some of the task on implementation and deployment may need more time than the specified one.

# RISK MANAGEMENT

Concerns future happenings

Involves hange in mind, opinion, actions, places etc..

Involves choice

Brainstorm:

* Other application on the market (already existing)
* Loss of external software (e.g. Google Maps, SMS gateway, Payment Methods)
* People leaving (difficult in this case, but heavy impact if happening)
* Change in available car zone

2 Tables:

1)

|  |  |  |  |
| --- | --- | --- | --- |
| **Code** | **Risk** | **Probability** | **Effect** |
| PR1 | A key member of the team quits or is ill in a critical point of the project development | Average | Serious |
| PR2 | Some parts of the project require more time than expected (underestimating schedules) | Average | Serious |
| PR3 | The external car software component doesn’t work as expected or is not ready in time | Low | Catastrophic |
| PR4 | Loss of entire or part of the source code | Low | Catastrophic |
| TR1 | External commercial components change their configurations/protocols/API | Low | Catastrophic |
| TR2 | A new paymenth method starts getting used widely | Average | Serious |
| BR1 | Change in available car parking zone | Average | Serious |
| BR2 | The project don’t satisfy the expectation of the city amministration | Average | Catastrophic |
| BR3 | Budget overestimation | High | Serious |
| BR4 | Budget underestimation | Average | Catastrophic |
| BR5 | Missing agreement with the mobile data vendor | Average | Catastrophic |

2)

|  |  |
| --- | --- |
| **Code** | **Mitigation strategy** |
| PR1 | Split duties and responsibilities among the highest number of people. In the worst case we can consider a first beta release with less functionalities than the defined ones during the requirement analysis phase |
| PR2 | Schedule an extra time slots before the major activities or releases |
| PR3 | A bunch of meetings with the external sw company to define deadlines |
| PR4 | Use of a backup system to store versions of data |
| TR1 | Try to build components that are independent among them (modularity of the system) with a particular focus on interfaces implementation |
| TR2 | Keep sistematically under control the new methods of payment on the market |
| BR1 | Exploit the time between the proposal and the confirmation of the measure to update our system |
| BR2 | Make the administration part of the development process (meetings and presentations on RASD and DD) |
| BR3 | Propose an alternative way of using the residual budget (application support, future releases, new functionalities) |
| BR4 | First release with less functionalities than the defined ones during the requirement analysis phase |
| BR5 | Exploit the economy of scale in the market of mobile data providers |

# EFFORT SPENT

## Giorgio Marzorati

29/11/2016 - 3h

04/12/2016 - 3h

05/12/2016 - 2h

08/12/2016 - 3h

09/12/2016 - 4h

10/12/2016 - 3h

11/12/2016 - 2h

## Aniel Rossi

29/11/2016 - 3h

04/12/2016 - 3h

05/12/2016 - 2h

08/12/2016 - 3h

09/12/2016 - 5h

10/12/2016 - 3h

11/12/2016 - 1h

## Andrea Vaghi

29/11/2016 - 3h

01/12/2016 - 2h

02/12/2016 - 3h

04/12/2016 - 1h

05/12/2016 - 2h

06/12/2016 - 1.30h

07/12/2016 - 1h

08/12/2016 - 3h

09/12/2016 - 1h

10/12/2016 - 5h

11/12/2016 - 3h

# CHANGELOG

V1.0 - First release