POLITECNICO
MILANO 1863

Software engineering 2 Project

Project Plan Document

PowerEnJoy

Perugini Alex	876359
Re Marco	873564
Scotti Vincenzo	875505

Table of Contents

1 Introduction

1.1 Purpose

The main purpose of this document is to provide an organization for the whole project phase, it first evaluates the expected complexity of the PowerEnJoy system and is intended to provide a work plan for the development of the system. This evaluations are based on previously released documents, in particular on the Design Document.
All these informations can be used as a reference point to define the budget of the whole system, the resources allocation and the schedule of the activities.

1.2 Scope

The scope of this document is to provide a specific plan to be followed developing the system, in particular we calculate the cost estimation of the system and the effort estimation in order to have the approximative budget and to know how many man/month of work are needed. The document also presents a scheduling of the activities and the people involved in that activities and an analysis of the needed resources and of all the risks that could occur during the development of the application.

1.3 Glossary

1.3.1 Glossary

Charging Area/Station	Subset of safe area where is possible to plug the car to a power grid to charge its battery
Client	User of PowerEnJoy registered to the system with normal log in elements
Driver	User sitting in the car that has the control of the vehicle, can be both a client or an operator
Driving Licence	Category B licence needed to drive the cars provided by PowerEnJoy
Money Saving Option	Functionality provided through the navigation system that enlights the nearest charging area to the client destination with free plugs
Operator	Employee of the company that provides the PowerEnJoy service, he is in charge of managing the cars according to the instructions he receives from the system
Passenger	Person inside the car that is not on the driver seat
Payment Method	The payment methods are all the methods accepted by the external payment system that the system will

	interface with
Power Grid	Grid located at each power station, it has plugs to charge the PowerEnJoy cars
Punishment	Fee on the charge for the ride
Ride	Entire operation of driving the car from the moment the engine is ignited to the moment the car is left parked in a safe area
Reward	Discount on the charge for the ride
Safe Area	Area defined by a set of GPS positions in which is possible to park the PowerEnJoy cars
Special login elements	Email and password gave to operators to authenticate as employes during the worktime

1.3.2 Acronyms

GPS	Global Positioning System
RASD	Requirement Analysis and Specification Document
DD	Design Document
ITPD	Integration Test Plan Document
DB	Data Base
DBMS	Data Base Management System
RDBMS	Relational Data Base Management System
UX	User Experience
BCE	Boundary Control Entity
ER	Entity Relationship
JEE	Java Enterprise Edition
FP	Function Points
LOC	Lines of Code
SLOC	Source Lines of Code
KSLOC	1000 Source Lines of Code
ILF	Internal Logic Files
EIF	External Interface Files

IDE	Integrated Development Environment

1.3.3 Abbreviations

App	Application, in this case refers to the software installed on smartphones
COCOMO	COnstructive COst MOdel

1.4 Reference Documents

1. Project description: Assignments AA 2016-2017.pdf
2. RASDPowerEnJoy.pdf
3. DDPowerEnJoy.pdf
4. ITPDPowerEnJoy.pdf
5. External document about COCOMO II
   http://csse.usc.edu/csse/research/COCOMOII/cocomo2000.0/CII_modelman2000.0.pdf
6. Example documents:
   - Second example of usage of FP and COCOMO for Assignment 5.pdf
   - Example of usage of FP and COCOMO for Assignment 5.pdf
   - Project planning example document.pdf

1.5 Document Structure

The document is organized as follows:

- Section 1: Introduction, provides a general description of how the estimations regarding the product realization is going to be done.
- Section 2: Size estimation, contains the procedure used to estimate the LOC based on the FP technique.
- Section 3: Effort estimation, contains the procedure used to estimate the man per hour effort necessary to develop the software. The value will be derived using the COCOMO II procedure applied to the values obtained by the FP evaluation.
- Section 4: Schedule, provides the description of the involved activities and how they're going to be performed.
- Section 5: Resources, provides the description of the available human resources and how they're going to be allocated in order to fulfill to the project realization.
- Section 6: Risks management, contains a report of the potential problems that can occur in the development and the related evaluation of a solution.
- Section 7: Used tools, contains the list of the tools used to realize the PPD and the scope they were used for.

Section 8: Hours of work, report of work time for each member.

2 Size estimation

This section provides the detailed procedure to evaluate the expected size of the PoweEnJoy service. The evaluation will be based on the FP technique applied to the structure identified in the DD.

2.1 F.P. Estimation

The Function Points technique provides an estimation of the size of the project expressed in Lines of Code. The procedure will, at first, derive the unadjusted values for the FP and then the Lines of Code will be derived using a language-dependent factor to normalize the value from the first estimate.
Here follows the weights used for the estimation:

	Low	**Average**	**High**
ILF	3	4	6
EIF	4	5	7
External Inputs	3	4	6
External Outputs	7	10	15
External Inquiries	5	7	10

2.1.1 Internal Logic Files

As presented by the ER schema of the internal DB for the application, the ILF we are going to handle are those about User, Car, Notification, Reservation, Ride and Safe area - Charging statio.

ILF	**Complexity**	**FP**
User	Low	7
Car	Average	10
Notification	High	15
Reservation	Average	10
Ride	Low	7
Safe area - Charging station	Low	7
Total		56

- User: the system stores just a few strings representing personal data for each client (email, password…) plus the coordinates resulting in low evaluation of FP.
- Car: the system stores battery level, state and position (that are continuously updated) plus some constant values (for example the windscreen code). Due to the continuous updates the evaluation of complexity is average.
- Notification: the estimated complexity results in an high value because even if the data representing the entry are simple, the procedure to generate the notification is quite complex (it involves many components, as shown by the sequence diagram in the DD).
- Reservation: differently from the Notification, the Reservation data are created only when requested by the client and doesn't need an internal control and evaluation of many data as it occurs with the Notification. Anyway after its creation a simple time control to check the expiration is needed, so the complexity is average.
- Ride: due to the simple raw data and the easy management (the ride is only created when it starts and updated few times) the complexity is low.
- Safe area - Charging station: are made of simple data stored and both rarely updated and accessed so are considered simple.

2.1.2 External Interface FIles

The only data to be managed externally are the ones regarding Payment and Driving License.

EIF	Complexity	FP
Payment	Low	5
License	Low	5
Total		10

- Payment and License: both complexity can be considered low since they are simple checks (license check) of data and standard requests (payment informations) with external systems.

2.1.3 External Inputs

The system interacts with the user to allow him to Log in - Log out, Registrate - Update Informations and Reservation.

External Inputs	Complexity	FP
Log in - Log out	Low	2x3
Registration - Update Informations	Average	2x4
Reservation Creation	Low	3
Unlock Car	Low	3
Total		20

- Log in - Log out: those are simple operations, so both weights are low.
- Registration - Update Informations: even if they are both simple operations in the data registration, they may involve the use of an external system, the evaluated weight is average.
- Reservation Creation: this operation requires only to create a new entry for the reservation and start its timer, so it can be considered of low complexity.
- Unlock Car: this operation requires only to do simple checks on a code or a position and to send the unlock car signal, so it can be considered of low complexity.

2.1.4 External Outputs

The system interacts with the user to make him aware of Notification, Reservation Expiration and to Display Actual Charge.

External Outputs	Complexity	FP
Notification	High	7
Map Refresh	High	7
Reservation Expiration	Average	5
Display Actual Charge	Low	4
Total		23

- Notification: since the procedure is onerous (the most complex algorithm is the one to find the nearest free operator) the overall complexity is high.
- Map Refresh: refreshing the map when a client is looking for a car to reserve can be considered an high complexity operation because the system continuously sends data to the client and has to check if the information of the map is changed.
- Reservation Expiration: involves different components to perform many simple operations (send the payment, change the car state, set the reservation as expired and notify the client) reaching an average complexity. The complexity is not high in this case because only simple operations are involved, there isn't a complex algorithm as in the notification case.
- Display Actual Charge: it's just the update of a value shown on a screen so it has a low complexity.

2.1.5 External Inquiries

The system allows user to require information as Read Movements History, Locate Cars and Use Safe Money Option (to require the navigation informations).

External Inquiries	Complexity	FP
Read Movements History	Low	3
Locate Cars	Average	5
Use Safe Money Option	High	6
Total		14

- Read Movements History: it's just a simple query and has a low complexity.
- Locate Cars: although the algorithm to identify the nearest cars cannot be considered simple, the overall complexity cannot be considered high because there are more difficult algorithms in the system, so the resulting complexity is estimated as average.

- Use Safe Money Option: this operation can be considered of high complexity due to the fact that it deals with a continuos updated data set (like car distribution).

2.2 Evaluation of estimation

The total produced FP are 123, the following table resumes the found values.

ILF	56
EIF	10
External Input	20
External Output	23
External Inquiries	14
Total FP	123

Given the total number of FP and since the intended language to develop the system is JEE we are able to estimate the total lines of code, based on an $AVC = 46$ (for the chosen language) that varies between an $AVC_{high} = 67$ and an $AVC_{low} = 15$.
Here follow the formula and the found values:

$$LOC = AVC * FP = 46 * 123 = 5658$$

$$LOC_{high} = 8241$$

$$LOC_{low} = 1845$$

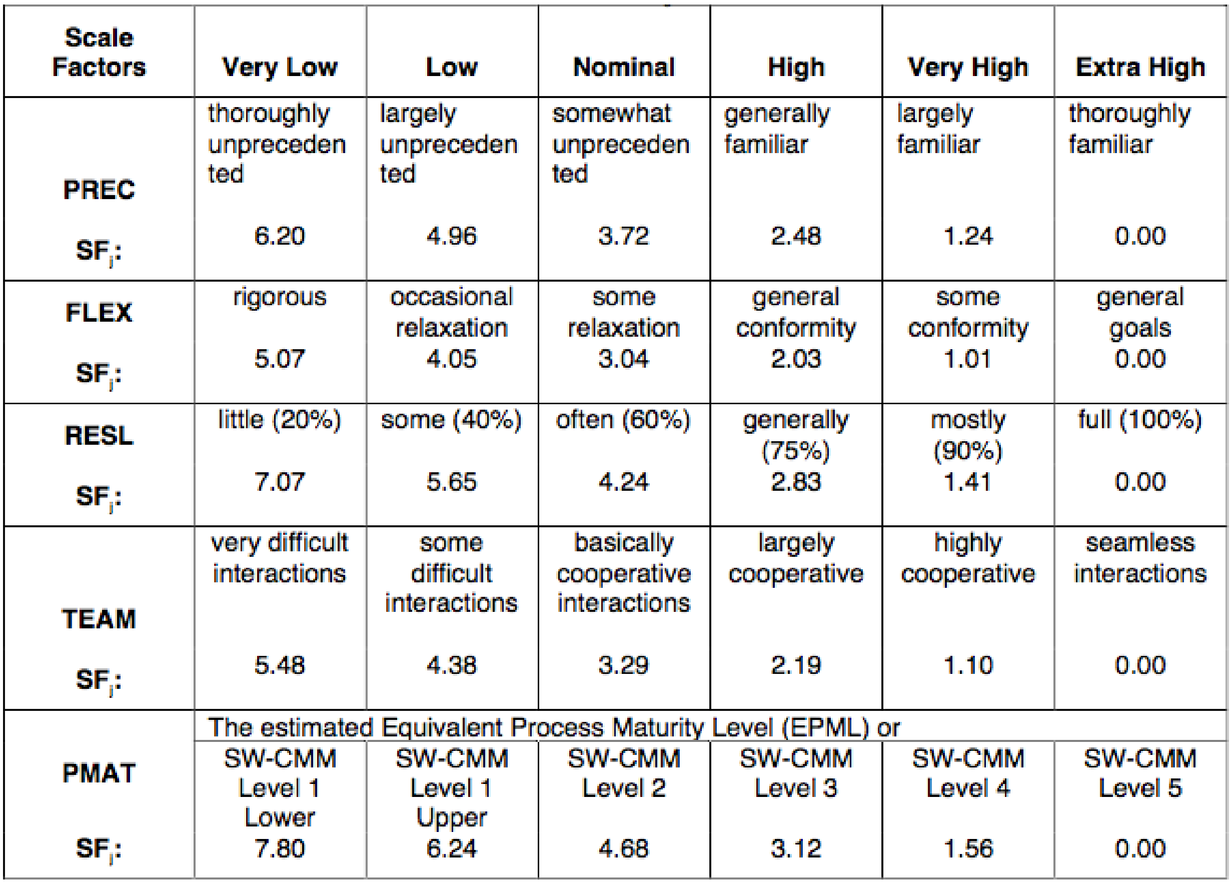
3 Effort estimation

3.1 COCOMO II

This section provides an estimation of the effort needed to develop the system based on the COCOMO II approach. The analysis is done considering that we are in a post-architecture phase, since the architecture of the system is widely presented in the previously released Design Document.
COCOMO II provides

3.1.1 Scale Factors

The COCOMO II approach is based at first on the estimation of the five scale factors.



Scale Factors	Very Low	Low	Nominal	High	Very High	Extra High
PREC SF_j:	thoroughly unprecedented 6.20	largely unprecedented 4.96	somewhat unprecedented 3.72	generally familiar 2.48	largely familiar 1.24	thoroughly familiar 0.00
FLEX SF_j:	rigorous 5.07	occasional relaxation 4.05	some relaxation 3.04	general conformity 2.03	some conformity 1.01	general goals 0.00
RESL SF_j:	little (20%) 7.07	some (40%) 5.65	often (60%) 4.24	generally (75%) 2.83	mostly (90%) 1.41	full (100%) 0.00
TEAM SF_j:	very difficult interactions 5.48	some difficult interactions 4.38	basically cooperative interactions 3.29	largely cooperative 2.19	highly cooperative 1.10	seamless interactions 0.00
PMAT	The estimated Equivalent Process Maturity Level (EPML) or					
SF_j:	SW-CMM Level 1 Lower 7.80	SW-CMM Level 1 Upper 6.24	SW-CMM Level 2 4.68	SW-CMM Level 3 3.12	SW-CMM Level 4 1.56	SW-CMM Level 5 0.00

That are:

- Precedentedness: the team has already developed non similar java projects but has never experienced web application and large scale projects, so the scale factor is considered low.
- Development Flexibility: the assignments for this project lists some required functionalities (therefore there are some constraints) but only at an high level without going in the detail, so the scale factor is nominal.
- Architecture / Risk Resolution: since this document provides a complete analysis of risks and how to solve them this value can be set to high

- Team Cohesion: two out of three members of the group already worked together in another project and in general the cohesion of all the members is very good, so this value is very high
- Process Maturity: according to the five CMMI maturity levels, we can consider the maturity level as defined (level 3), so the value is nominal

Scale factor	**Level**	**Value**
Precedentedness	Low	4.96
Development Flexibility	Nominal	3.04
Architecture / Risk Resolution	High	2.83
Team Cohesion	Very high	1.10
Process Maturity	Nominal	4.68
Total		**16,61**

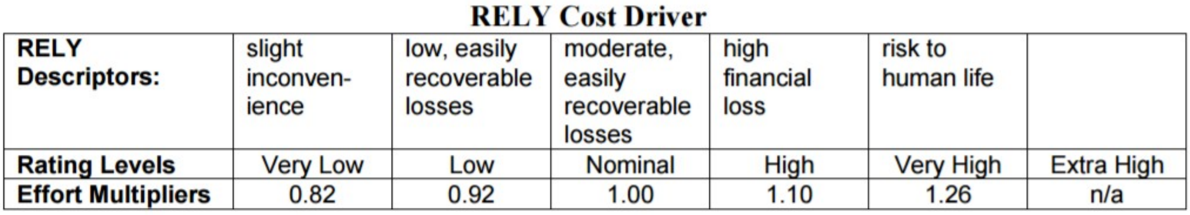
$$E = B + 0.01 * \sum_{j=1}^{5} SF_j \qquad with\, B = 0.91$$

$$E = 1.0761$$

3.1.2 Cost Driver

Next step in the effort estimation procedure is to analyse to derive the EM term in the main formula.

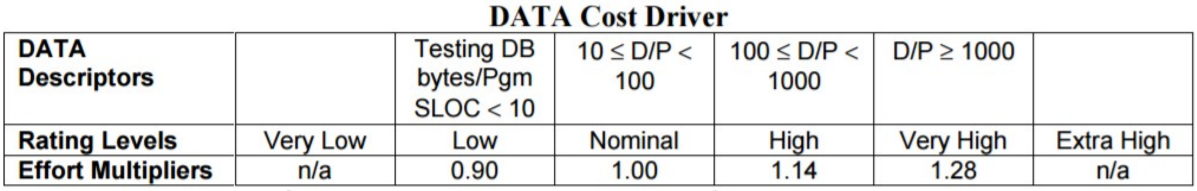
- **Required software reliability (RELY)**: the failure of the software could lead to malfunctions in the use of the service, in fact it could be impossible to have available all the functionalities offered but these functionalities don't influence human life or financial losses, so the value is nominal.



RELY Cost Driver

RELY Descriptors:	slight inconvenience	low, easily recoverable losses	moderate, easily recoverable losses	high financial loss	risk to human life	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	0.82	0.92	1.00	1.10	1.26	n/a

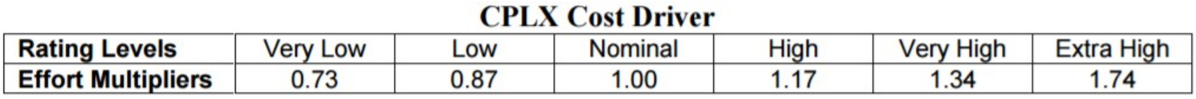
- **Data base size (DATA)**: according to other considerations made before in this document about the complexity of data structures, the size of the DB can be reasonably estimated as order of GB. This size is distributed over approximately 5000 LOC so the D/P factor is estimated between 100 and 1000, so the value is high.



DATA Cost Driver

DATA Descriptors		Testing DB bytes/Pgm SLOC < 10	10 ≤ D/P < 100	100 ≤ D/P < 1000	D/P ≥ 1000	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	n/a	0.90	1.00	1.14	1.28	n/a

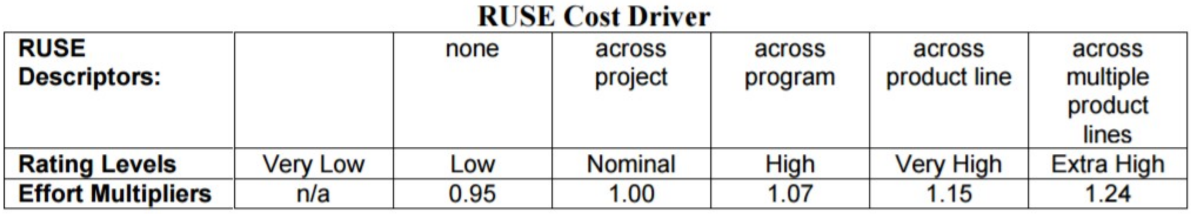
- **Product complexity (CPLX)**: value set to high according to COCOMO II rating scale and all general considerations made previously in this document.



CPLX Cost Driver

Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	0.73	0.87	1.00	1.17	1.34	1.74

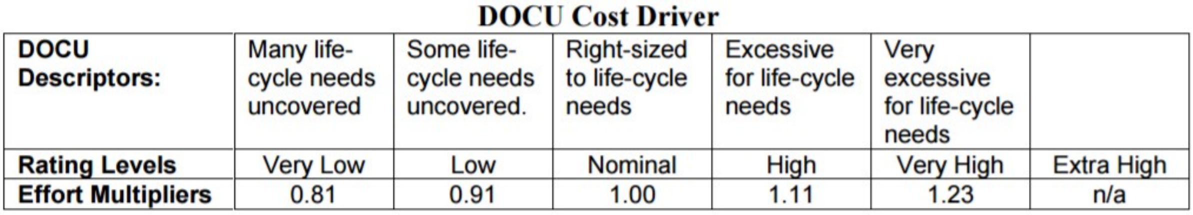
- **Reusability (RUSE)**: the reusability of modules and components are to the project itself and not specifically designed to be reused in other situations, so the value is nominal.



RUSE Cost Driver

RUSE Descriptors:		none	across project	across program	across product line	across multiple product lines
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	n/a	0.95	1.00	1.07	1.15	1.24

- **Documentation match to life-cycle needs (DOCU)**: the documentation of this project has to cover the needs of a life-cycle, a poor documentation could bring to extra costs in the future, so this value is set to nominal.

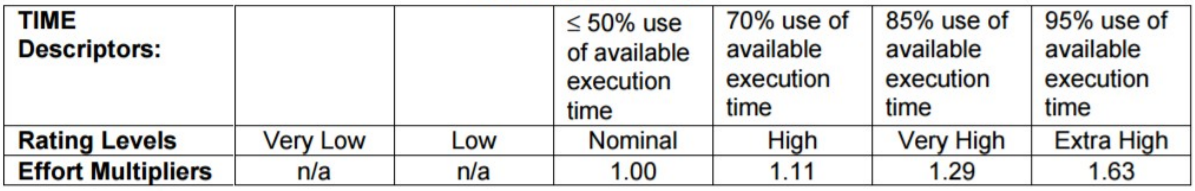


DOCU Cost Driver

DOCU Descriptors:	Many life-cycle needs uncovered	Some life-cycle needs uncovered.	Right-sized to life-cycle needs	Excessive for life-cycle needs	Very excessive for life-cycle needs	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	0.81	0.91	1.00	1.11	1.23	n/a

- **Execution time constraint (TIME)**: the expected amount of CPU used by the system with respect to the hardware capabilities is about 70%, because in the ITPD the software has been precisely analysed in order to have proper hardware, so the value is set to high.

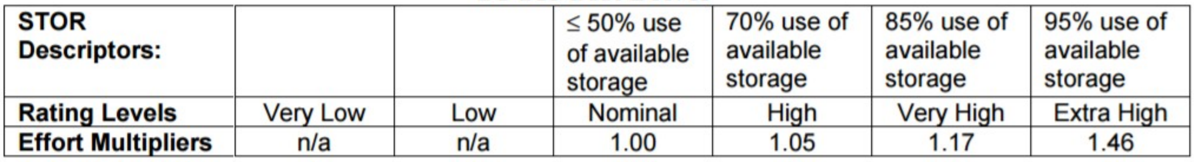
TIME Cost Driver



TIME Descriptors:			≤ 50% use of available execution time	70% use of available execution time	85% use of available execution time	95% use of available execution time
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	n/a	n/a	1.00	1.11	1.29	1.63

- **Storage constraints (STOR)**: according to what is said in for the previous value and the relatively low cost of TB of storage this value is set to nominal.

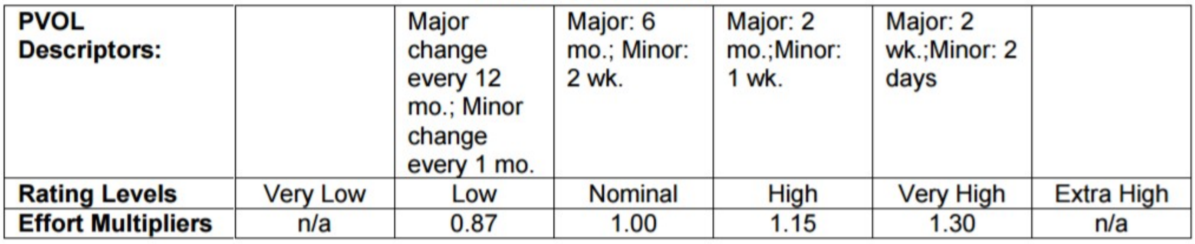
STOR Cost Driver



STOR Descriptors:			≤ 50% use of available storage	70% use of available storage	85% use of available storage	95% use of available storage
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	n/a	n/a	1.00	1.05	1.17	1.46

- **Platform Volatility (PVOL)**: the platform used are not expected to change very often, for example JEE platform has a major release once every one or two years. We can assume that other chosen platforms are stable as JEE is, so this value is set to low.

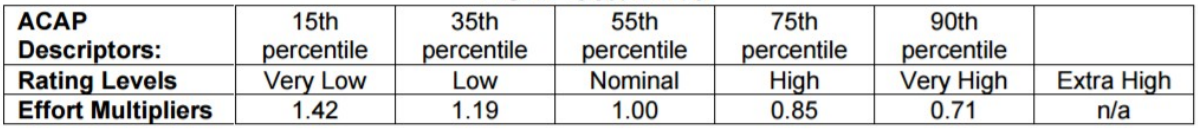
PVOL Cost Driver



PVOL Descriptors:		Major change every 12 mo.; Minor change every 1 mo.	Major: 6 mo.; Minor: 2 wk.	Major: 2 mo.;Minor: 1 wk.	Major: 2 wk.;Minor: 2 days	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	n/a	0.87	1.00	1.15	1.30	n/a

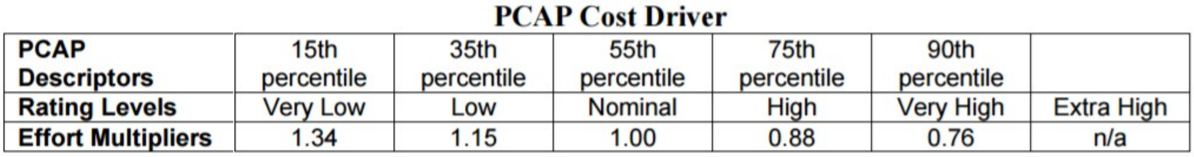
- **Analyst capability (ACAP)**: the analysis of the requirements has been done in a detailed and specific way in the RASD considering many possible scenarios so this value is set to high.

ACAP Cost Driver



ACAP Descriptors:	15th percentile	35th percentile	55th percentile	75th percentile	90th percentile	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	1.42	1.19	1.00	0.85	0.71	n/a

- **Programmer capability (PCAP)**: the estimation of this parameter could be set to nominal because the members of the group have good programming skills but, given that this is the first time that they develop a web application, it’s possible to encounter various kind of problems due to inexperience.

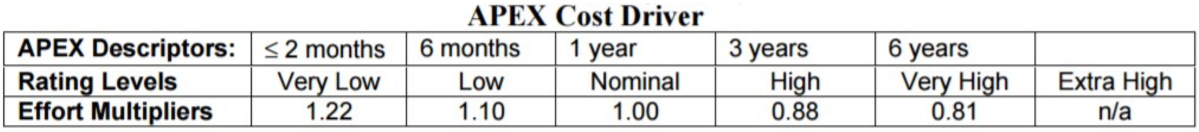


PCAP Cost Driver

PCAP Descriptors	15th percentile	35th percentile	55th percentile	75th percentile	90th percentile	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	1.34	1.15	1.00	0.88	0.76	n/a

- **Personnel continuity (PCON)**: the analysis of this driver is not relevant in this case.

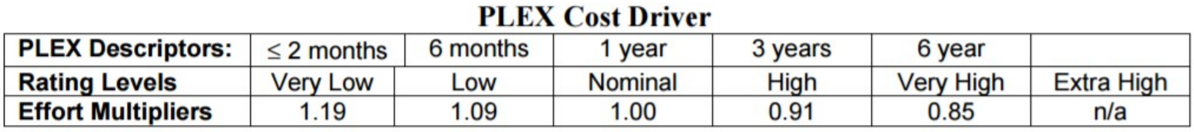
- **Application experience (APEX)**: the team has experience in developing Java application but not in JEE application so this parameter is set to low.



APEX Cost Driver

APEX Descriptors:	≤ 2 months	6 months	1 year	3 years	6 years	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	1.22	1.10	1.00	0.88	0.81	n/a

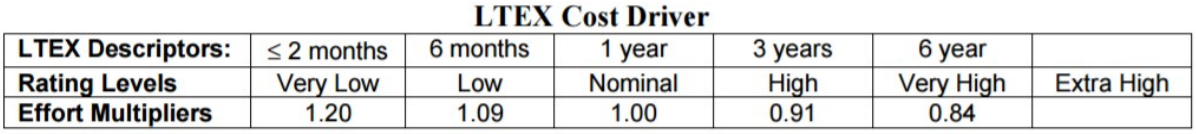
- **Platform experience (PLEX)**: the team doesn’t have experience with JEE platform but have some in server side application developing and in databases so this value is set to low.



PLEX Cost Driver

PLEX Descriptors:	≤ 2 months	6 months	1 year	3 years	6 year	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	1.19	1.09	1.00	0.91	0.85	n/a

- **Language and tool experience (LTEX)**: the team knows well the Java language but never used tools like NetBeans and the testing tools listed in the ITPD document. The IDE is similar to other ones used before (Eclipse and IntelliJ) so the value of this driver is set to nominal.

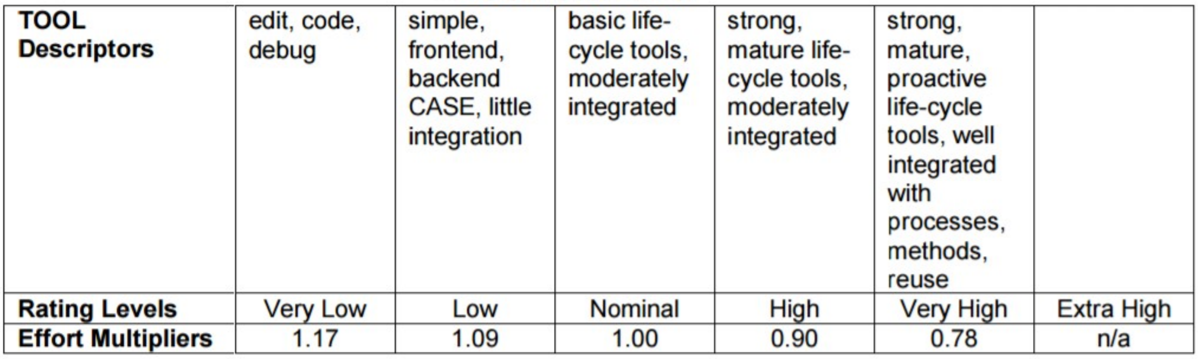


LTEX Cost Driver

LTEX Descriptors:	≤ 2 months	6 months	1 year	3 years	6 year	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	1.20	1.09	1.00	0.91	0.84	

- **Use of software tools (TOOL)**: the platform and the environment chosen for this project are strong, mature life-cycles and well integrated so this value is set to high.

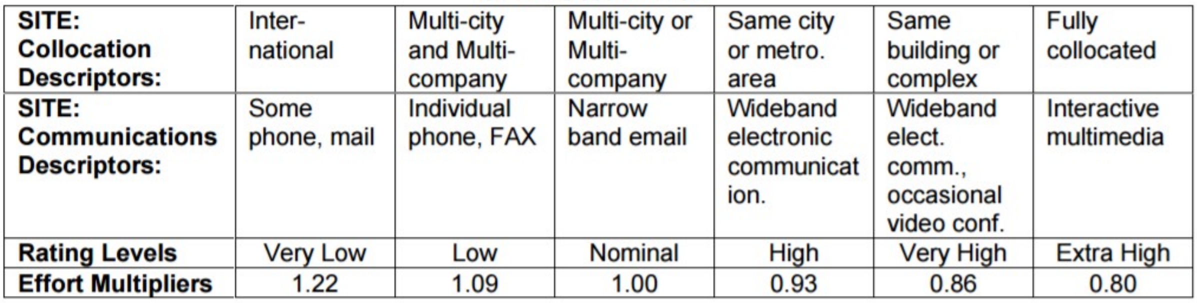
TOOL Cost Driver



TOOL Descriptors	edit, code, debug	simple, frontend, backend CASE, little integration	basic life-cycle tools, moderately integrated	strong, mature life-cycle tools, moderately integrated	strong, mature, proactive life-cycle tools, well integrated with processes, methods, reuse	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	1.17	1.09	1.00	0.90	0.78	n/a

- **Multisite development (SITE)**: the team is all located into the same metro area so this parameter is set to high.

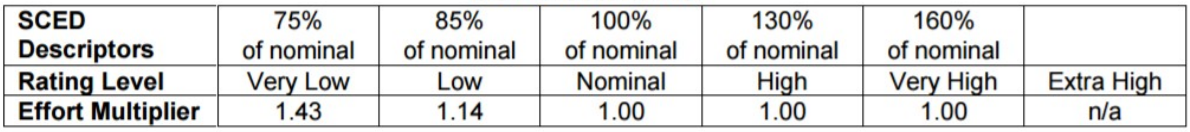
SITE Cost Driver



SITE: Collocation Descriptors:	Inter-national	Multi-city and Multi-company	Multi-city or Multi-company	Same city or metro. area	Same building or complex	Fully collocated
SITE: Communications Descriptors:	Some phone, mail	Individual phone, FAX	Narrow band email	Wideband electronic communication.	Wideband elect. comm., occasional video conf.	Interactive multimedia
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	1.22	1.09	1.00	0.93	0.86	0.80

- **Required development schedule (SCED)**: the schedule doesn't result to be stretched-out or compressed during the whole project so this value is set to nominal.

SCED Cost Driver



SCED Descriptors	75% of nominal	85% of nominal	100% of nominal	130% of nominal	160% of nominal	
Rating Level	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multiplier	1.43	1.14	1.00	1.00	1.00	n/a

Now is possible to sum up all the drivers taken into account into this table:

Cost Driver	**Level**	**Value**
Required software reliability (RELY)	Nominal	1.00
Data base size (DATA)	High	1.14
Product complexity (CPLX)	High	1.17
Reusability (RUSE)	Nominal	1.00

Documentation match to life-cycle needs (DOCU)	Nominal	1.00
Execution time constraint (TIME)	High	1.11
Storage constraints (STOR)	Nominal	1.00
Platform volatility (PVOL)	Low	0.87
Analyst capability (ACAP)	High	0.85
Programmer capability (PCAP)	Nominal	1.00
Personnel continuity (PCON)	-	-
Application experience (APEX)	Low	1.10
Platform experience (PLEX)	Low	1.09
Language and tool experience (LTEX)	Nominal	1.00
Use of software tools (TOOLS)	High	0.90
Multisite development (SITE)	High	0.93
Required development schedule (SCED)	Nominal	1.00
Total		1.09874399

3.1.3 Effort Equation

$$PM = A * Size^{E} * \prod_{i=1}^{16} EM_i$$

$$A = 2.94$$

$$5658/1000 = 5.658\,KSLOC$$

$$E = 1.0761$$

$$PM = 20.8537631$$

4 Schedule

In this chapter are described the tasks necessary to complete the PowerEnJoy project and their relative scheduling.
It is important to underline that even if this document has been realized after the real project start, the scheduling process was already ongoing and so the parts regarding the RASD and the DD, that was both necessary for this document completion, are part of the schedule shown in this section.
Another additional and important information regarding this section is that the schedule reported is subject to changes and so it's not definitive.
Moreover from the duration formula $Duration = 3.67 * Effort^{F}$, where $F = 0.28 + 0.2 * (E - B)$ and $B = 0.91$ so we derive:

$$F = 0.28 + 0.2 * (1.0716 - 0.91) = 0.31232$$

$$Duration = 3.67 * 20.8537631^{0.31232} = 9.477\,months$$

4.1 Tasks and milestones identification

4.1.1 Milestones

Milestone	Date
M1- Project start	16/10/2016
M2 - RASD meeting with stakeholders	25/10/2016
M3 - RASD submission deadline	13/11/2016
M4 - RASD presentation and review	16/11/2016
M5 - DD meeting with stakeholders	30/11/2016
M6 - DD submission deadline	11/12/2016
M7 - DD presentation and review	14/12/2016
M8 - ITPD meeting with stakeholders	21/12/2016
M9 - ITPD submission deadline	15/01/2016
M10 - ITPD presentation and review	22/01/2017
M11 - Development meeting with stakeholders (first)	29/01/2017
M12 - Development meeting with stakeholders (second)	30/04/2017
M13 - Development deadline	12/07/2017
M14 - Final meeting with stakeholders	18/07/2017

M15 - Deployment deadline	25/07/2017
M16 - Project end	01/09/2017

4.1.2 Tasks

RASD		
Activity	**Duration** (days)	**Dependencies**
T1 - Identify product characteristics	7	M1
T2 - Identify and describe requirements	7	T1
T3 - Alloy model realization	14	T2
T4 - Meeting with stakeholders	2	M2
T5 - Document revision	5	M4

DD		
Activity	**Duration (**days)	**Dependencies**
T6 - Design architecture	7	T3
T7 - Design main algorithms	7	T6
T8 - Design UI	7	T6
T9 - Requirements traceability	7	T6
T10 - Meeting with stakeholders	2	M5
T11 - Document revision	5	M7

ITPD		
Activity	**Duration** (days)	**Dependencies**
T12 - Define integration strategy	14	T7, T8, T9
T13 - Identify and describe tools and data	14	T12

T14 - Meeting with stakeholders	2	M8
T15 - Document revision	5	M10

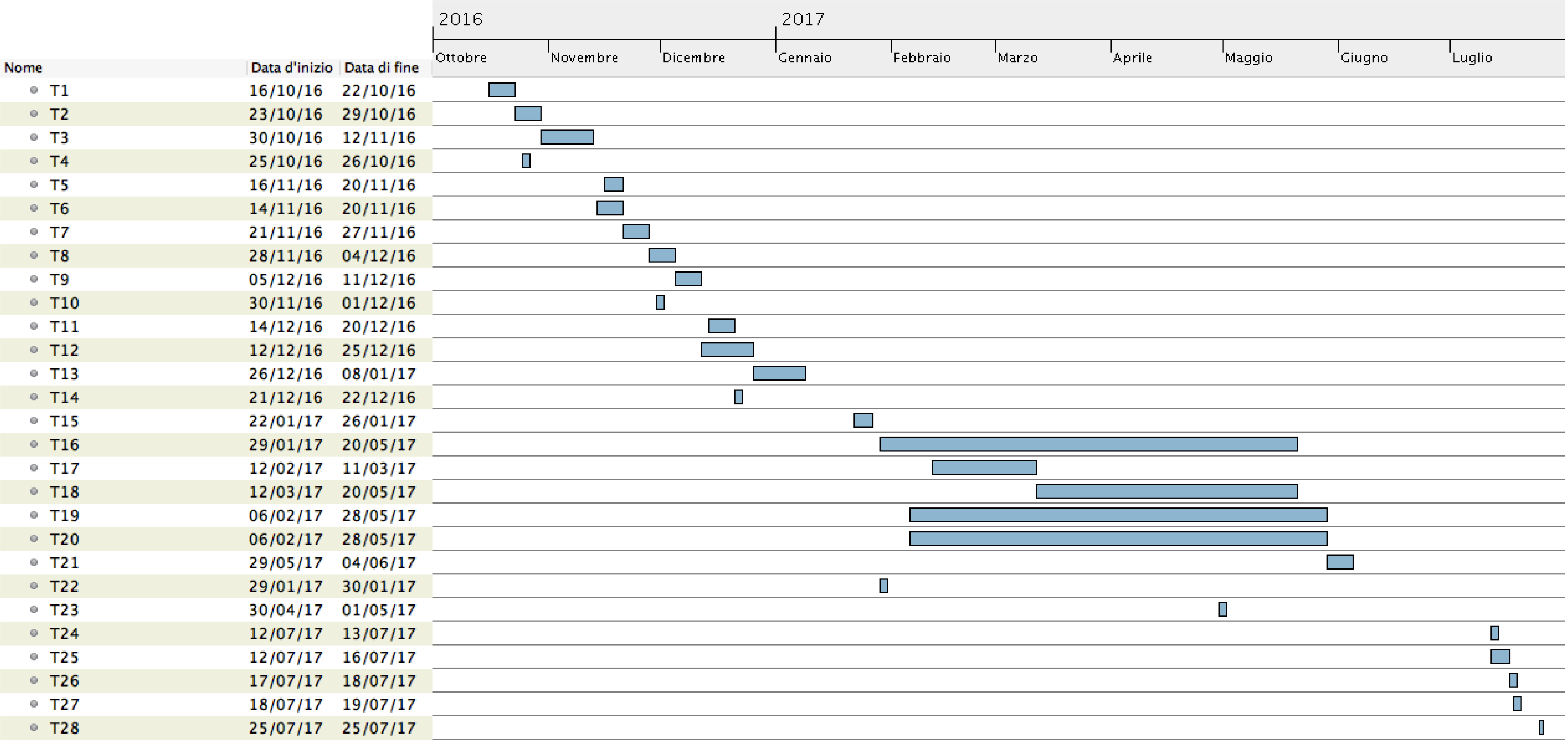
Development		
Activity	**Duration** (days)	**Dependencies**
T16 - Components development	112	M9
T17 - Unit testing	28	T16 (start)
T18 - Integration testing	70	T17
T19 - Document code	112	T16 (start)
T20 - Code inspection	112	T16 (start)
T21 - Write manual	7	T16, T18, T19, T20
T22 - Meeting with stakeholders (first)	2	M11
T23 - Meeting with stakeholders (second)	2	M12

Deployment		
Activity	**Duration** (days)	**Dependencies**
T24 - System deployment	2	M13,T16, T18, T19, T20
T25 - Operators formation	5	T16, T18, T19, T20
T26 - Cars distribution	2	T24, T25
T27 - Final meeting with stakeholders	2	M14

Startup		
Activity	**Duration** (days)	**Dependencies**
T28 - System first start	1	M15

4.2 Schedule

Here follows the derived schedule. As shown by the graph there are some periods without any activity to be performed, this was done on purpose in case the actual execution doesn't meet the estimate time.



Nome	Data d'inizio	Data di fine
T1	16/10/16	22/10/16
T2	23/10/16	29/10/16
T3	30/10/16	12/11/16
T4	25/10/16	26/10/16
T5	16/11/16	20/11/16
T6	14/11/16	20/11/16
T7	21/11/16	27/11/16
T8	28/11/16	04/12/16
T9	05/12/16	11/12/16
T10	30/11/16	01/12/16
T11	14/12/16	20/12/16
T12	12/12/16	25/12/16
T13	26/12/16	08/01/17
T14	21/12/16	22/12/16
T15	22/01/17	26/01/17
T16	29/01/17	20/05/17
T17	12/02/17	11/03/17
T18	12/03/17	20/05/17
T19	06/02/17	28/05/17
T20	06/02/17	28/05/17
T21	29/05/17	04/06/17
T22	29/01/17	30/01/17
T23	30/04/17	01/05/17
T24	12/07/17	13/07/17
T25	12/07/17	16/07/17
T26	17/07/17	18/07/17
T27	18/07/17	19/07/17
T28	25/07/17	25/07/17

5 Resources

5.1 Available resources

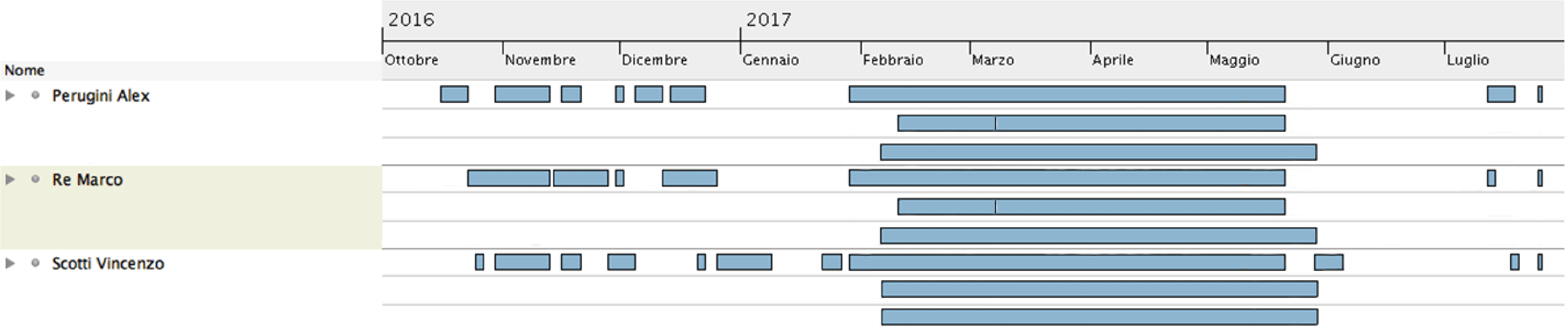
The human resources available for the project are the three members of the team, Alex Perugini, Marco Re and Vincenzo Scotti. As for the overall schedule,also the allocation of resources may be subject to changes in order to match all the deadlines.

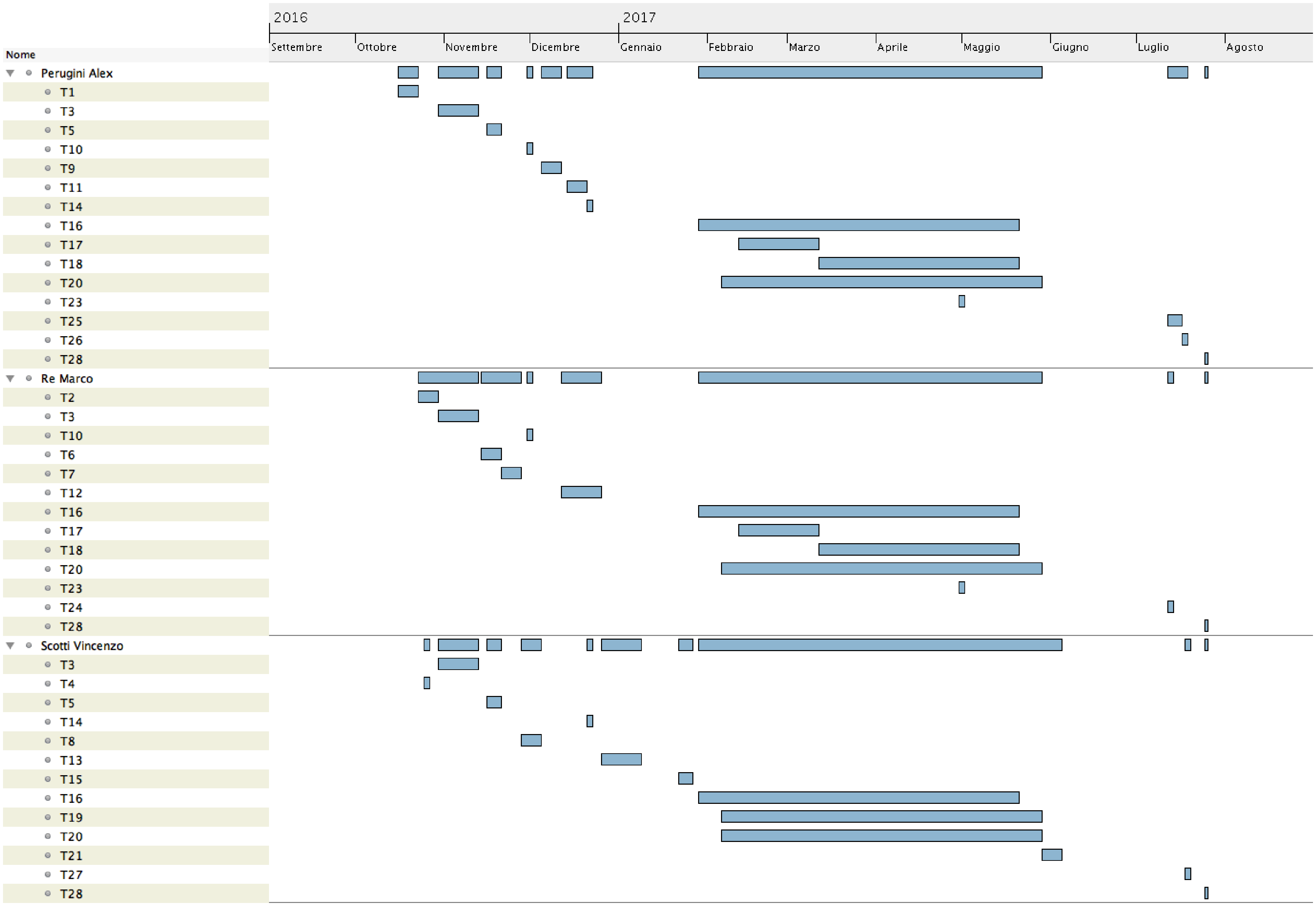
5.2 Resources allocation

The prevented schedule will be divided between the members of the team has shows the following table:

Member	**Task**
Perugini Alex	T1,T3, T5, T9,T10,T11, T14,T16, T17, T18, T20, T23, T25, T26, T28
Re Marco	T2, T3, T6, T7, T10, T12, T16, T17, T18, T20, T23, T24, T28
Scotti Vincenzo	T3, T4, T5, T8, T13, T14, T15, T16, T19, T20, T21, T27, T28

Here follows the resulting graph for the resources allocation:


2016
2017
Ottobre
Novembre
Dicembre
Gennaio
Febbraio
Marzo
Aprile
Maggio
Giugno
Luglio
Nome
Perugini Alex
Re Marco
Scotti Vincenzo


2016
2017
Settembre
Ottobre
Novembre
Dicembre
Gennaio
Febbraio
Marzo
Aprile
Maggio
Giugno
Luglio
Agosto
Nome
Perugini Alex
T1
T3
T5
T10
T9
T11
T14
T16
T17
T18
T20
T23
T25
T26
T28
Re Marco
T2
T3
T10
T6
T7
T12
T16
T17
T18
T20
T23
T24
T28
Scotti Vincenzo
T3
T4
T5
T14
T8
T13
T15
T16
T19
T20
T21
T27
T28

6 Risks management

In this section we are going to consider risks we may face while developing the PowerEnJoy application. We not only take into account technical risks but also ones related to business and to project planning.

First risks we are going to evaluate are political ones. The main issue concerning this category is that of a change in laws regarding the city viability for electric cars. While a debarment for electric cars to access some zones of the city is quite unlikely to happen, it wouldn’t represent a serious risk to face. In fact we could simply make all clients aware of the forbidden zones and of the fees they would pay. There should also be an update of the system to store those zones and according to stakeholders desires, we can also take into account to notify users if they’re going to drive through them.

One of the major risks that has to be faced is the possibility that requirements have been misunderstood or that stakeholders want to change them. It is well known that this is quite likely to happen and depending on the nature of the required changes consequences may be quite serious. The best way to avoid problems of this type is to make stakeholders conscious of the project as much as possible. This can be done through periodical reviews, meetings and demonstrations of prototypes of the product. In this way both developers and stakeholders would have an immediate feedback on the project, making the resolution of problems much more simple.

Another problem we have to go through is the possibility that the most used store like App Store, Google Play Store and Windows Store require specific features for the app to be deployed there. During the development of the application we have to be sure that all required features are satisfied, otherwise it would take time to make the app match them. The absence of the app on those store in fact would represent a big issue in its distribution, which is also strongly influenced by the presence of other similar apps in the stores. To make the app more accepted by market in presence of similar apps, we suggest considering to develop a very intuitive interface and a marketing campaign for the product.

There can also be problems linked to the project planning itself. We have to consider that it can be too optimistic and it can't take into account all the possible issues. This would produce an increase in the cost of the whole project and to avoid this, some extra time has been allocated to eventually adjust problems. Linked to this, there is also the possibility to have underestimated the time of development. Then, we can contemplate the idea to release a first version of the product at the established deadline and then, when it is fully completed, there is a second release.

Among the issues that can make the project late we have to consider team linked ones. Some people of the team may get ill in the same period which can be easily faced overlapping some activities and making each team member aware of others’ work. There may also be differences of opinion or skills in the same team, making the cooperation much more difficult than expected. In those cases we may also think to change some members or to split the group in two or more.

Regarding more technical issues we have to consider different aspects. First of all we have to avoid all risks linked to a loss of the code which would have disastrous consequences. This can simply done through the use of a version control system and an appropriate use of backups.

Then we have to think about the possibility that the server or the database are inadequate for the developed system, both due to an overestimate of usage and to an underestimate of

it. In the both cases we should allow for possibility to change them and get ones more adequate.
We also have to bear in mind that there may be issues regarding the external interfaces we have to deal with. In fact a policy has to be decided with the stakeholders about what to do if the payment system doesn’t work or if a driving license can’t be checked due to a malfunction of motorization gateway.

7. Used tools

- Google Drive: documents sharing
- Google Docs: word processor, concurrent work platform
- GitHub: control version
- GanttProject: schedule and resources allocation representation

8. Hours of work

Perugini Alex:

- 12/01/17 2h 30m
- 13/01/17 2h
- 14/01/17 2h
- 15/01/17 1h 30m

Re Marco:

- 12/01/17 2h 30m
- 13/01/17 2h
- 14/01/17 3h 30m

Scotti Vincenzo

- 12/01/17 2h 30m
- 13/01/17 2h
- 14/01/17 2h
- 15/01/17 1h