Politecnico di Milano

Software Engineering 2

PROJECT PLAN

PowerEnjoy

Authors:

Simone Bruzzechesse

Luca Franceschetti

Gian Giacomo Gatti

INDEX

1 INTRODUCTION

In this document, we will analyse the effort and the cost of the realization of our PowerEnjoy system, in order to achieve that we will estimate the size of our project. First, we will use the Function Points' approach and then COCOMO model and moreover we will identify project tasks, their schedule and also the resource allocated. To show the last thing we will use a Gantt Chart and in the last part of the document we will report a brief list of the possible risks that affected our project plan and we will specify their relevance and a way to recover from them.

2 FUNCTIONAL POINT APPROACH

The Functional Point approach is a technique that is used to evaluate the effort needed for the design and implementation of an application. This technique evaluates the complexity of an application analysing the functionalities of the application itself. We use the functional point approach according to the different functionalities that can be derived from the RASD document and from the Design Document. In particular, the technique that we analyse groups the functionalities of the application in:

- **Internal Logic File**: it represents a set of homogeneous data handled by the system and in our system, this turns out to be all the data structure stored in the system's database.
- **External Interface File**: it represents a set of homogeneous data used by the application but generated and handled by external application and so that needs an interface between the first and the second to exchange them. In our system, these are the interfaces with external services.
- **External Input**: elementary operation to elaborate data coming from the external environment.

- **External Output**: elementary operation that generates data for the external environment.
- **External Inquiry**: elementary operation that involves simple input and output. operations.

The following table outline the number of Functional Point based on functionality and relative complexity:

Function Point	Complexity		
	Simple	Average	Complex
Internal Logic File	7	10	15
External Interface File	5	7	10
External Input	3	4	6
External Output	4	5	7
External Inquiry	3	4	6

2.1 INTERNAL LOGICAL FILE (ILF)

The PowerEnjoy system must store information related to:

1. **Users**: the users that have access to the system:
    a. These tables have a several number of attributes;
    b. They are probably associated to a high number of reservation and trips so there could be a lot of association within the tables;
    c. The system must provide security for the information contained in the user record;

    Overall the weight of the user information is: **Medium;**

2. **Cars**: all the cars belonging to the system:
    a. These tables contain the information related to a car;
    b. They are associate to a high number of reservation and trip;

    Overall the weight of the car information is: **Simple;**

3. **Reservation**: the reservation that are made by the users:

   a. These tables are associated both to a user and to a car of the system and also, they will probably be associated to a trip;
   b. These tables contain the information related to reservation;

   Overall the weight of the reservation information is: **Simple**;

4. **Trip**: the trips that are made by the users of the system:
   a. These tables contain several attributes;
   b. They are associated to a reservation and the corresponding user and car;
   c. They change information several times due to the fact that the trip can be in different state;

   Overall the weight of the trip information is: **Complex**;

5. **Safe Areas and Charging Station**: the first are the safe areas of the system while the second are the charging station that belong to the system:
   a. These tables are both simple since they have only a few attributes;

   Overall the weight of the both information is: **Simple**;

We summarize the information related to the ILF using the table below:

Data	**Weight Type**	**Weight Associated**
Users	Medium	10
Cars	Simple	7
Reservation	Simple	7
Trip	Complex	15
Safe Areas	Simple	7
Charging Station	Simple	7
TOTAL		**53**

2.2 EXTERNAL INTERFACE FILE (EIF)

The Power Enjoy system must interact with some external system in order to provide its functionalities, in detail:

1. **Maps Service**: the service that can inform the system about the current position of the users and the cars of the system and also to the position in the map of the safe areas, since the information that the system need from external one are heterogenous we need to implement a complex interface. So, the weight associated will be: **Complex**;
2. **Notification Service**: this service allows the system to send information to the user such as the payment information of the last trip. Since the task of this service are only a few one we will need a simple interface that interact with it. So, the weight associated will be: **Simple**;
3. **Payment Service**: this service manages the payments of the system, since this task is very frequent in many applications its interface will be easy to implement so its weight will be: **Simple**;

We summarize the information related to the EIF using the table below:

Data	Weight Type	Weight Associated
Maps Service	Complex	10
Notification Service	Simple	7
Payment Service	Simple	7
TOTAL		**24**

2.3 EXTERNAL INPUT (EI)

The external inputs of the system are all the possible way of interaction between the user and the application.

- **Login/Logout**: these are simple operations related only to the user of the system, the only difficulty of this kind of operation is related to the Login since also the securityManager is involved in the operation as its role is to check whether the user can log in into the system according to the user and password that are inserted.
- **Registration**: this is a simple operation which again, as the login/logout is related only to the user, it will add a new User into the database and all his personal information and payment info are stored.
- **Manage User Account**: this is a simple operation which involves only the personal data and the payment information of the user of the application.
- **Make Reservation**: this operation is considered complex since it requires to store some information into the database (such as the user who reserve and the car that is reserved but also the initial time of the reservation) and also it must consider when the timer expires to charge the user who reserve the car.
- **Delete Reservation**: this operation is considered simple since it only requires to retrieve the list of reservations of the system and check that the reservation is not expired and then delete that from the system.
- **Start Trip**: this operation is considered of average complexity since it must check whether there is a previous reservation between the user and the car and after this check it must obtain a lot of information such as the initial time and position and also the number of passengers inside the car.
- **Enable Money Saving Option**: this is a complex operation of the system since as we said before in the RASD and the DD documents it involves a complex evaluation of the destination and also the path between the initial position of the user and the final position that is given.

- **Stop Trip**: this operation is considered complex since it involves a lot of operation and check as previously said in the DD and the RASD document. The most important ones are the check on the final position, the evaluation of the bill associated to the trip and also the association of the discounts to the trip itself.

We summarize the information related to the external input using the table below:

External Input	**Complexity**	**FP**
Login	Simple	3
Logout	Simple	3
Registration	Simple	3
Manage User Account	Simple	3
Make Reservation	Complex	6
Delete Reservation	Simple	3
Start Trip	Average	4
MoneySavingOption	Complex	6
Stop Trip	Complex	6
TOTAL		**37**

2.4 EXTERNAL OUTPUT (EO)

The external outputs of the system are described below:

- **Final Destination** (according to the money saving option) **and path**: this is an average complexity information that is send from the system to the car of the user who select the money saving option.
- **Invoices:** the system generates involves which are simple information related to the last trip of the user and that is send to the last one using the notification manager and also shown on the car's tablet.

We summarize the information related to the external output using the table below:

External Output	Complexity	FP
Final Destination	Average	5
Invoices	Simple	4
TOTAL		**9**

2.5 External Inquiry (EQ)

Since the application scope is mainly to manage the reservation and the trip associated to the user, it doesn't have too much inquiry. The only ones are listed below:

- **User Profile**: the user profile can be shown if the user asks to manage or to see it;
- **Car Status:** during the navigation of the user in the map, he can select a car and ask for the car status that is displayed by the system on the user application.

We summarize the information related to the external output using the table below:

External Inquiry	Complexity	FP
User Profile	Simple	3
Car Status	Simple	3
TOTAL		**6**

2.6 Final Results

The final results can be evaluated according to the different weights determined for each element, using the following equation we can derive the total number of FPs and so we can provide an indication of the size of the system in functional points terms:

$$UFP = ILF + EIF + EI + EO + EQ = 53 + 24 + 37 + 9 + 6 = \mathbf{129}$$

Using the FPs above we can also obtain an estimation of the lines of code (LOC) of the system using the following formula:

$$LOC = AVC \times UFP$$

where AVC is a language dependent factor and we use the AVC specific for J2EE (46). So the final estimation is:

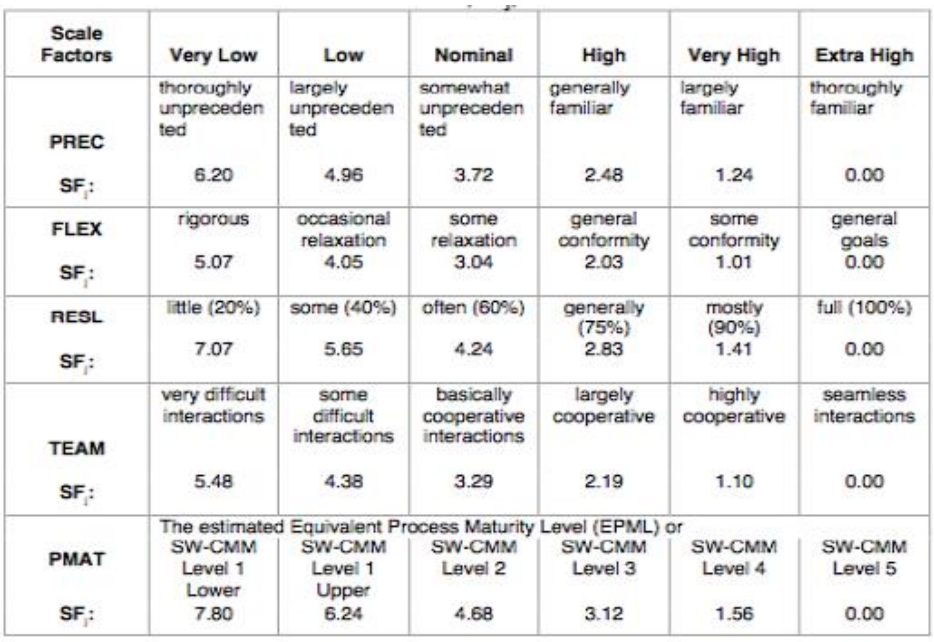
$$LOC = 46 \times 129 = \mathbf{5934}$$

3 COCOMO

3.1 Exponent

3.1.1 Scale Factors

In the table below we show the scale factors used to evaluate the exponent:



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

Then we evaluate the scale factor in our system according to the precedence values:

- **Precedentedness**: It refers to the fact that the system that we are realizing is similar to other ones that were developed before. Since the framework used in the system is new this value will be low.
- **Development flexibility**: it refers to the degree of flexibility during the development process, since there were specific instructions in the main features but nothing specific is said about technologies we can assume that the values is high.

- **Risk resolution**: this value will be high if we have a good risk management plan, clear definition of budget and schedule, focus on architectural definition. Since the risk management plan is good and the role and the schedule is well-define the value will be very high.
- **Team cohesion**: this value will be high if all the members of the group are able to work in a team and share the same vision, since we are able to work in group and after the first reunion to share a common idea of the project we can assume that the value will be high. Not very high since for most of the time we work singularly.
- **Process maturity**: refers to a well-known method for assessing the maturity of a software organization, since the goals are achieved these values will be set to high.

Scale Factor	Factor	Value
Precedentness	LOW	4.96
Development Flexibility	HIGH	2.03
Risk Resolution	VERY HIGH	1.41
Team Cohesion	HIGH	2.19
Process Maturity	HIGH	3.12
Total:		**13.71**

3.1.2 Evaluation of the Exponent E

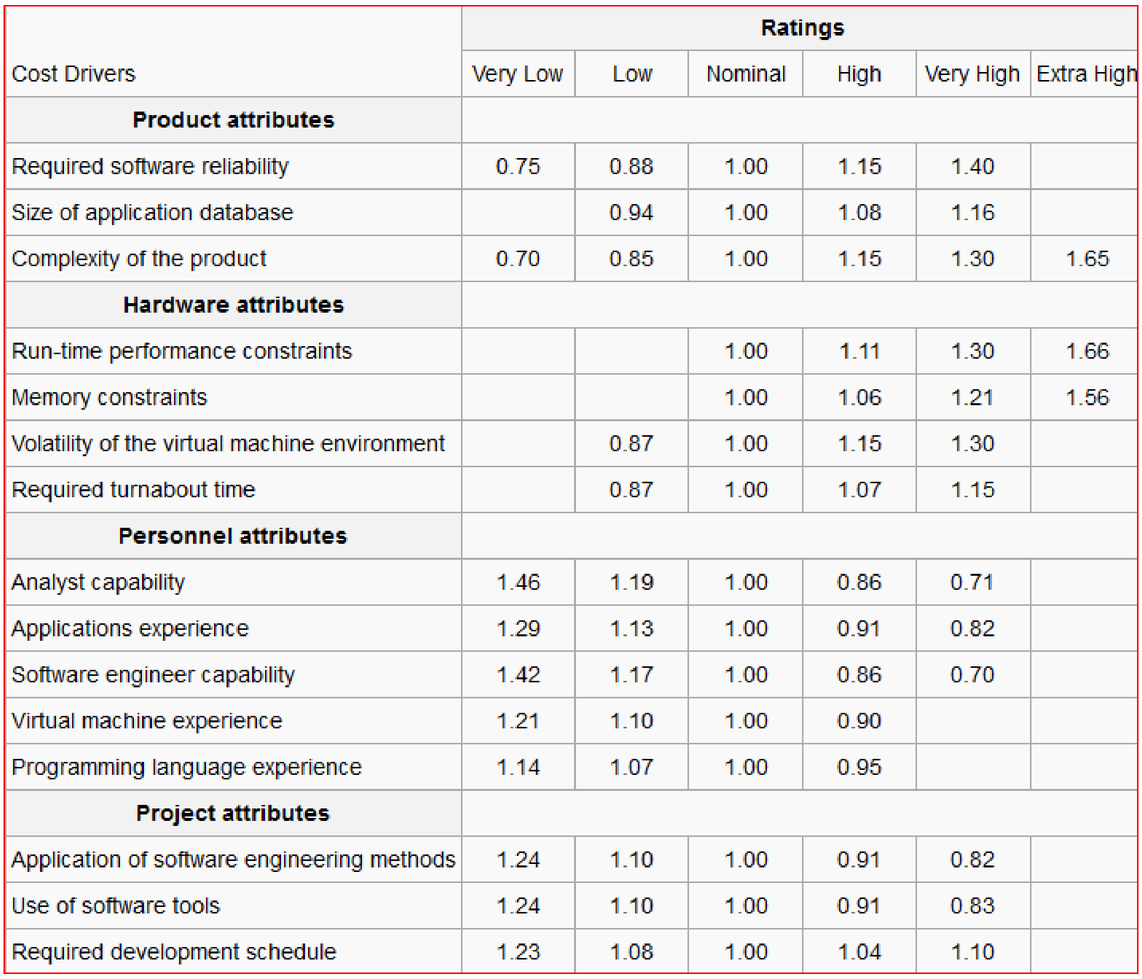
The evaluation of the Exponent is based on the following equation:

$$E = B + 0.01 \times \sum_{1 \le j \le 5} SF_j \qquad , where\ B = 0.91$$

So, the value of E is: 0.91 + 0.01 * 13.71 = **1.047.**

3.2 EAF

In the figure we report the cost drivers:



Cost Drivers	Ratings					
	Very Low	Low	Nominal	High	Very High	Extra High
Product attributes						
Required software reliability	0.75	0.88	1.00	1.15	1.40	
Size of application database		0.94	1.00	1.08	1.16	
Complexity of the product	0.70	0.85	1.00	1.15	1.30	1.65
Hardware attributes						
Run-time performance constraints			1.00	1.11	1.30	1.66
Memory constraints			1.00	1.06	1.21	1.56
Volatility of the virtual machine environment		0.87	1.00	1.15	1.30	
Required turnabout time		0.87	1.00	1.07	1.15	
Personnel attributes						
Analyst capability	1.46	1.19	1.00	0.86	0.71	
Applications experience	1.29	1.13	1.00	0.91	0.82	
Software engineer capability	1.42	1.17	1.00	0.86	0.70	
Virtual machine experience	1.21	1.10	1.00	0.90		
Programming language experience	1.14	1.07	1.00	0.95		
Project attributes						
Application of software engineering methods	1.24	1.10	1.00	0.91	0.82	
Use of software tools	1.24	1.10	1.00	0.91	0.83	
Required development schedule	1.23	1.08	1.00	1.04	1.10	

We give our evaluation to every cost driver according to our system.

PRODUCT ATTRIBUTES

- **Required Software Reliability**: this measure is very high since a system failure means loss of the actual reservation and the status of the trip so it could mean a big loss also in economic terms.
- **Data Base Size**: since the size of the DB is quite big because of the number of Reservation, Trip and SafeAreas this number will be high.
- **Product Complexity**: Set to high according to the new COCOMO II CPLEX rating scale.

HARDWARE ATTRIBUTES

- **Execution Time Constraint**: this value is not too much important in our application so it can be put as low.
- **Main Storage Constraint**: since this parameter represents the degree of main storage constraint we can put it as very low.
- **Platform Volatility**: since we use J2EE platform and general operating system and browser we can assume that the platform doesn't change too frequent so we can assume this value as nominal.
- **Required turnabout time**: this value is set to nominal since the deliveries were frequent and the members of the group were also involved in other projects.

PERSONNEL ATTRIBUTES

- **Analyst Capability**: we can assume this value as high since a big amount of time was spent in the design document and the requirement analisys.
- **Application Experience**: this parameter is evaluated according to the past experience of the members of the group with web projects and the J2EE framework so in this case we can set this parameter to nominal since we never develop such projects but we know the architecture.
- **Programmer Capability**: this parameter can be set as high since the members of the group have already experience with the programming language used.
- **Platform Experience**: since our average knowledges about platforms such as: databases, ui and application-server development are good so this parameter is set to high.
- **Language and Tool Experience**: this parameter is set to high since we already knew Java and already used NetBeans.

PROJECT ATTRIBUTES

- **Application of software engineering methods**: this value is set to high since all the members of the group already know some engineering method due to the previous experience with software projects.

- **Use of software tools**: this parameter is set to very high because of the previous experiences.
- **Required development schedule**: since our efforts is distributed well over the time but the implementation required high efforts we set this parameter to high.

Then in the table we show our evaluation of the cost driver according to our system:

Cost Driver	**Factor**	**Value**
Required Software Reliability	VERY HIGH	1.40
Data Base Size	HIGH	1.08
Product Complexity	HIGH	1.15
Execution Time Constraint	LOW	N.A.
Main Storage Constraint	VERY LOW	N.A.
Platform Volatility	NOMINAL	1.00
Required turnabout time	NOMINAL	1.00
Analyst Capability	HIGH	0.86
Application Experience	NOMINAL	1.00
Programmer Capability	HIGH	0.86
Platform Experience	HIGH	0.90
Language and Tool Experience	HIGH	0.95
Application of software engineering methods	HIGH	0.91
Usage of Software Tools	VERY HIGH	0.83
Required development schedule	HIGH	1.04
EAF:		**0.864**

In the end, we evaluate the EAF as:

$$EAF = scale\ driver_1 \times scale\ driver_2 \times \ldots \times scale\ driver_{15}$$

So **EAF = 0.864**

3.3 Effort

Overall, we resume the value evaluated before and use them in order to obtain an estimation of the effort that is needed for the development of the system (express in Person-Months PM).

$$Effort = A \times EAF \times KSLOC^{E}$$

Where:

- A → **2.94**
- EAF → product of all the cost drivers, equal to: **0.864**;
- E → exponent derived from Scale Drivers: **1.047**;
- KSLOC → estimated lines of code using the FP analysis: **5.934**

With these parameters, we can compute the Effort value, that is equal to:

$$Effort = 2.94 \times 0.864 \times 5.934^{1.047} = 16.39\ PM$$

3.4 Duration and staffing

For the schedule estimation, we are going to use the following formula:

$$Duration = 3.67 \times Effort^{F}$$

Where:

- F → 0.28 + 0.2 * (E – B) = 0.3074
- Effort → 16.39

Follows then:

$$Duration = 3.67 \times 16.39^{0.3074} = \mathbf{8.67\ months}$$

Then we can estimate the number of people needed to complete the project:

$$Members = \frac{Effort}{Duration} = \frac{16.39}{8.67} = 1.89 \sim \mathbf{2\ Person}$$

4 TASKS AND SCHEDULES

The tasks of the PowerEnjoy system are the following:

T1. Requirements Analysis and Specification

T2. Design Document

T3. Integration Test Plan

T4. Project Plan

T5. Code Inspection

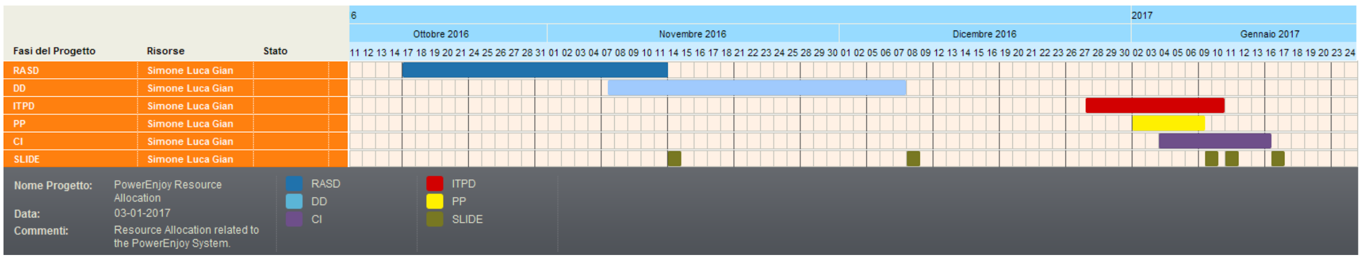
T6. Project Slide Presentation

The schedule of the previous tasks is the following:

Task	Submission Deadline
RASD	13/11/2016
DD	11/12/2016
ITPD	15/01/2017
PP	22/01/2017
CI	05/02/2017
Slides	(to be schedules)

5 Resources Allocation

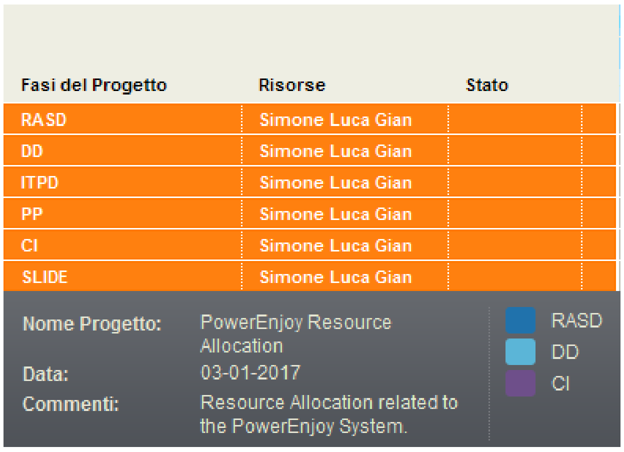
We use a Gantt chart to show the allocation of resources to the different tasks, in particular the members of the group are the resources of the system. The beginning and the end of the block correspond to the beginning and at the end of the activity which involved all the members of the group as shown in the figure.



Fasi del Progetto	Risorse	Stato
RASD	Simone Luca Gian	
DD	Simone Luca Gian	
ITPD	Simone Luca Gian	
PP	Simone Luca Gian	
CI	Simone Luca Gian	
SLIDE	Simone Luca Gian	

Nome Progetto: PowerEnjoy Resource Allocation
Data: 03-01-2017
Commenti: Resource Allocation related to the PowerEnjoy System.

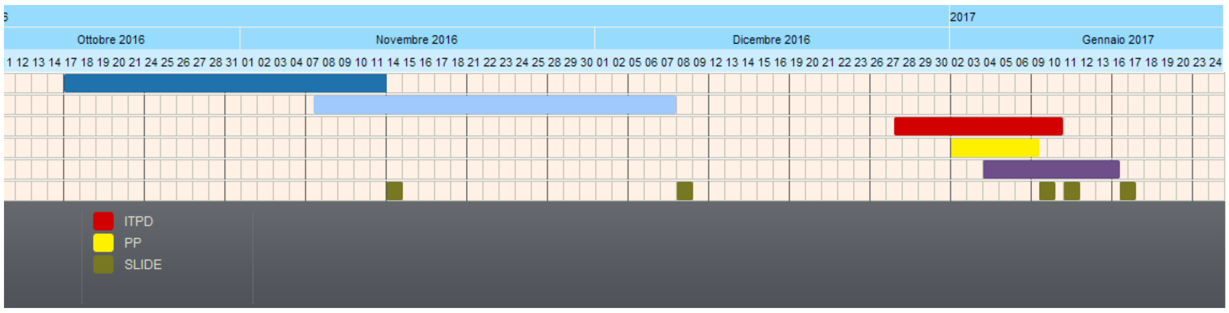
🟦	RASD
🟦	DD
🟪	CI
🟥	ITPD
🟨	PP
🟫	SLIDE



Fasi del Progetto	Risorse	Stato
RASD	Simone Luca Gian	
DD	Simone Luca Gian	
ITPD	Simone Luca Gian	
PP	Simone Luca Gian	
CI	Simone Luca Gian	
SLIDE	Simone Luca Gian	

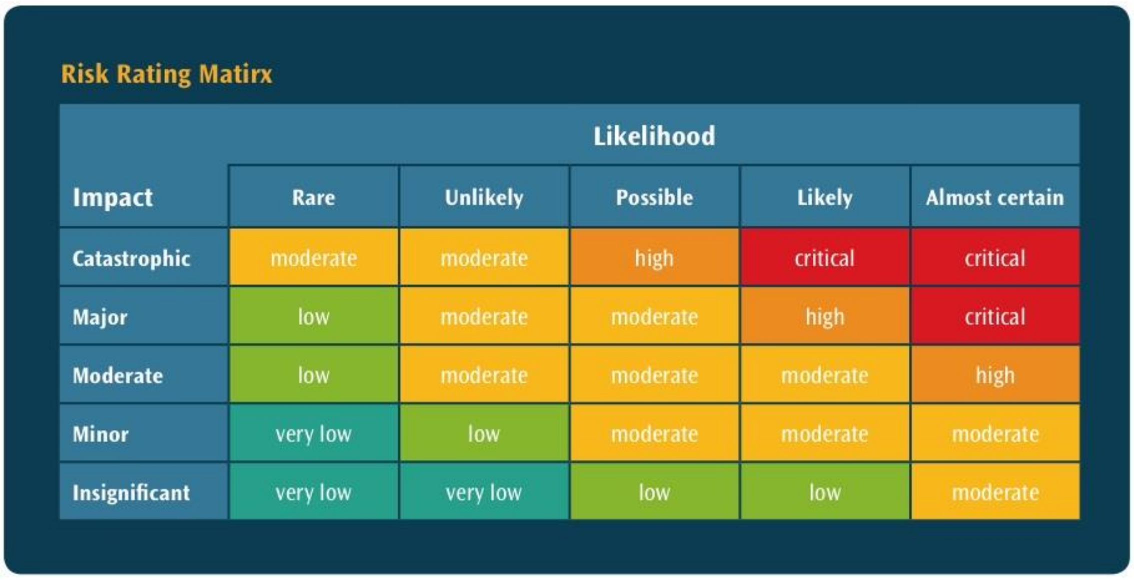
Nome Progetto: PowerEnjoy Resource Allocation
Data: 03-01-2017
Commenti: Resource Allocation related to the PowerEnjoy System.

🟦 RASD
🟦 DD
🟪 CI



6 PROJECT RISKS

In the development of a project, handling the risks is a very important task. The first thing to do is to identify all the most relevant risks that can occur and identify the possibilities that the risk can occur and also the impact if it happens. Then it is important to determine the recovery actions in order to solve the risks. It is important that in order to define the risks rating, we refer to the table below where, for every factor, we give an evaluation for impacts and likelihood.



Risk Rating Matirx

	Likelihood				
Impact	**Rare**	**Unlikely**	**Possible**	**Likely**	**Almost certain**
Catastrophic	moderate	moderate	high	critical	critical
Major	low	moderate	moderate	high	critical
Moderate	low	moderate	moderate	moderate	high
Minor	very low	low	moderate	moderate	moderate
Insignificant	very low	very low	low	low	moderate

Then we analyse the major risk of our system that are listed below:

Risk	Likelihood	Impact	Recovery Action
Lack of experience with technologies that are used	Likely	Moderate	It is important to take time to learn tools and technologies that are involved in the system in order to avoid loss of time in the advanced part of the project.
Unrealistic Schedules	Possible	Major	Since the schedules are already determined it is important to organize time in the best way in order to not miss a deadline.
Developing wrong functions	Possible	Major	Take care of specifying the functionalities that are developed using the right documentation and check with the client if they correspond to the wanted ones.
Test Failure	Likely	Moderate	Use continues test inspection during the coding phase using stub and driver to test the single class of the system.
Data Loss in the server DB	Rare	Catastrop hic	Use backup frequently to avoid complete loss of data related to the application
Bad External Components	Unlikely	Major	Check documentation and other example of interfaces with external services and check if they can work or not in order to use only services which are already working in the right way
Lack of communication	Possibly	Major	Communicate to each member of the team every important change in the system behaviour and plan meeting to discuss what have been done during between the meetings.

We will adopt a proactive strategy, that tries to avoid risk instead of trying to solve them after they are already become a problem. So, the strategy is to identify all the potential threat and analyse them, also taking in consideration the probability of happening and the potential damage.

7 OTHER INFO

7.1 HOURS OF WORK

We report approximately how many hours each member has worked on this document.

- Simone Bruzzechesse: 10 hours
- Luca Franceschetti: 10 hours
- Gian Giacomo Gatti: 10 hours

7.2 USED TOOLS

1. Word
2. Gant chart editor: “https://www.tomsplanner.it/”