Politecnico di Milano

A.A. 2016-2017

Software Engineering 2:

PowerEnJoy

Project **P**lan

Lorenzo Frigerio (mat. 879203)

Martina Perfetto (mat. 808869)

Ivan Vigorito (mat. 879204)

22th January 2017

Version 1.0

Contents

1. Introduction

This document aims to evaluate the effective effort needed to entirely develop PowerEnJoy project

in all its features, and after that give a hypothesis on how to schedule the development. The

evaluation starts with a Function Point Analysis, which will provide a rough estimate of the

SLOC (Source Lines of Code). Then with SLOC value, we will proceed with a COCOMO II analysis

to calculate correspondent Effort and Duration.

The second part of the document will explain the project schedule through tasks identification

and allocation to team members.

1.1. Definitions, Acronyms, Abbreviations

User / registered user: he/she is the client of the service; he/she is able to rent a car in order to travel around the city. He is associated with:

-Name

-Surname

-Other personal information

-Method of payment

-Number of driving license and expiration date

-Password

Employee/operator: is the one that help users in case of emergency and has the responsibility of managing cars in case of malfunction. Users can call them by

using the telephone exchange of the application.

Method of payment: is inserted by the user during the registration phase but can be updated over the time. Only one method is active at once and payment are concluded using services offered by the different companies holding the credit card. An invoice containing all the charges collected is generated monthly.

Car: sometimes referred as vehicle is the means of transport rented by users. It contains a set of sensors that analyse the number of passengers presents on the car, control the charge of the battery and detect when a door is closed. Moreover, it includes a module that transmit this information to the system using the Internet.

Available car: car that is not in use at the moment by any user, has at least 20% of charge and is not reserved by anyone.

Reservation: made by a user that wants to use a car. Has a duration of one hour maximum and is associated with a unique car. Once the user asks to unlock the car the car becomes associated to the user until he decides to end the ride.

Charge: amount of money that users have to pay due to the use of the service. It is immediately calculated by the system after a ride but money is transferred only at the end of the month.

Penalty: fee derived from a bad behaving of the user such as a damage on the car or a fine for exceeding speed limits. The fee will be notified to the user and included in the monthly invoice.

GPS navigation device: system that equip each car and that is able to calculate the exact position of the car and display to the user the route to follow. Its display is also used to show the current fee of the ride and the status of the battery.

Special Safe Area: special parking areas that contain plugs that allow cars to be recharged. They are provided with sensors that detect the number of spots that are currently used and communicate the number to the system. They are also called power grid stations.

Safe Area: Space included in boundaries that determine where users can park a car. It covers entire metropolitan cities in order to facilitate users to find a park

and they may also contain power grid stations. Users cannot terminate a ride while outside from a safe area.

In this document we use ‘Safe Area’ to identify both Safe Area and Special Safe Area.

Ride/Rental: it last from when the user picks up the car until when the system stops charging the user. It includes a possible set of temporary stops and the total path travelled by the user.

Park: is when a user leaves the car and wants to end the rental. At this point the system stops charging the user.

Stop: is when a user leaves the car but wants to resume the ride in the future. The car will be locked by the system that, however, will continue to charge the user for the ride.

1.2. Reference Documents

- PowerEnJoy RASD

- Function Points in line of code: http://www.qsm.com/resources/function-point-languages-table

- The COCOMO II Model definition manual http://csse.usc.edu/csse/research/COCOMOII/cocomo2000.0/CII_modelman2000.0.pdf

2. Project Size and Cost evaluation

2.1. Function Point Analysis

In order to calculate the Function Points we start to spot the features of the project and classify them in five categories: External Inputs, External Outputs, External Inquiries, Internal Logic Files and External Logic Files. Beginning from these features we calculate the overall amount on Function Points and then the expected size (Line of Code) required.

These tables are used to evaluate complexity:

For Internal Logical Files and External Interface Files

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

For External Output and External Inquiry

	Data Elements		
File Types	**1 - 5**	**6 - 19**	**20+**
0 or 1	Low	Low	Avg.
2 - 3	Low	Avg.	High
4+	Avg.	High	High

For External Input

	Data Elements		
File Types	**1 - 4**	**5 - 15**	**16+**
0 or 1	Low	Low	Avg.
2 - 3	Low	Avg.	High
3+	Avg.	High	High

Function Type	Complexity-Weight Low	Average	High
Internal Logical Files	7	10	15
External Interfaces Files	5	7	10
External Inputs	3	4	6
External Outputs	4	5	7
External Inquiries	3	4	6

2.1.1. Internal Logic Files (ILFs)

PowerEnJoy is based on a significant amount of stored information that allows to manage, in addition to the traditional administration of users, the cars and the cities in which the system is deployed. However, users still represents a consistent part of the application, in fact they contain many personal information and are also linked to the chosen payment methods so they are quite complex to manage. Moreover, there are the rides and the reservations which are composed of many attributes such as the date of beginning and end but also the costs. Safe Area and Special Safe Area are easier to administer since the simply represents boundaries of cities and the set of charging stations present there. Finally, cars and employees have a low complexity since they are not composed of lots of fields, they only need to store the status of the cars and some basic information of the employees working for the system.

Internal Logic Files	Complexity	Function Points
User	Average	10
Ride	Average	10
Reservation	Average	10
Safe Area	Low	7
Special Safe Area	Average	10
Payment Method	Low	7
Car	Average	10
Employee	Low	7
Total		71

2.1.2. External Interface Files (ELFs)

The external services used by PowerEnJoy are focused on the notification system and the payment system. As regard the notifications the application is dependent on the information returned by the external SMS e mail services. Whereas, payments are based on external APIs and are more complex to manage since they require a safe environment and many efforts to prevent external attack to the system.

External Interace Files	Complexity	Function Points
Mail	Low	5
Sms	Low	5
Payment Information Retrival	Average	7
Total		17

2.1.3. External Inputs (EIs)

PowerEnJoy requires the implementation of many input requests coming from different actors of the system.

- User: through the mobile app and the web application can perform the basic requests such as login/logout and the registration; all of them requires only the User Management so they contribute 3 FPs only. Moreover, they are responsible of the requests of unlock the car when nearby which require the allocation of a new ride and perform the request to the car on unlocking. Finally, users can make requests for reserve an available car, the operation requires the creation of a new reservation while checking that the user has no other active reservations.
- Car: The car module can send to the system some requests. For instance, it can require the calculation of the route to follow, High complexity since requires the managing of Safe Areas, Special Safe Areas but also Maps. In addition, the begin and the end of the ride when the engine is ignited or the car is closed with no passengers inside. Both requires the system to begin or end to calculate the cost

of the ride. Lastly, cars collect the information retrieved from sensor and send them to the system which is in charge of calculating the final cost based also on these inputs

- Employee: He is responsible for the requests of updating the status of a car when under maintenance or a problem has occurred. Since it requires only the update of a single element, this function is considered with low complexity.

External Inputs	Complexity	Function Points
Login/Logout/Register	Low	3x3
Car Reservation	Average	4
Unlock / Lock car	Average	4x2
Calculate route	High	6
Update user's information	Low	3
Update car status	Low	3
Begin/End the ride	Average	4x2
Send ride information from car to the system	Low	3
Total		44

2.1.4. External Inquiries (EQs)

Regarding inquiries, the system has to manage some requests or retrieving information coming from users and employees. The system is able to return the personal information of a client to a user that wants to check his data but also to an employee. In addition, it can return the set of available car through a simple query that return all the car with status available without further computations, the operation is more complex than the previous since requires both cars and safe areas to retrieve an answer. Employees can also check the status of the cars and its related rides in order to support users in case of problems.

External Inquiries	Complexity	Function Points
Retrieve user information	Low	3
Retrieve rides of users	Average	4
Retrieve car info	Average	4
Retrieve available cars	Average	4
Total		15

2.1.5. External Outputs (EOs)

The system produces some external outputs addressed mainly to the car but also to notify the user in case of necessity. In fact, the system is designed to notify the user the calculated invoices, based on Rides and Users and requires many data elements. In regard to the cars, the system send to them the calculated cost of the ride based only on the information of the Ride.

External Outputs	Complexity	Function Points
Update cost of the ride	Low	4
Update Maps	Average	5
Send invoices notifications	Average	5
Total		14

2.1.6. Overall estimation

Internal Logic Files	71
External Logic Files	17
External Inputs	44
External Inquiries	15
External Outputs	14
Total	161

Assuming that PowerEnJoy will be developed using Java Enterprise Edition (46 SLOC/FP) as a platform the total SLOC are calculated as:

SLOC = 161 * 46 = 7406

2.2. Cost and effort estimation: COCOMO II

2.2.1. Scale Drivers

In order to, evaluate the values of the scale drivers, we refer to the following official COCOMO II table:

Scale Factors (*Wi*)	Very Low	Low	Nominal	High	Very High	Extra High
PREC	thoroughly unprecedented	largely unprecedented	somewhat unprecedented	generally familiar	largely familiar	throughly familiar
FLEX	rigorous	occasional relaxation	some relaxation	general conformity	some conformity	general goals
RESL[a]	little (20%)	some (40%)	often (60%)	generally (75%)	mostly (90%)	full (100%)
TEAM	very difficult interactions	some difficult interactions	basically cooperative interactions	largely cooperative	highly cooperative	seamless interactions
PMAT	Weighted average of "Yes" answers to CMM Maturity Questionnaire					

- Precedentedness: It indicates the experience of the team with similar projects in the past. Since the team has low experience with related software systems and some innovative algorithms are needed for the managing of the cars the indicator is set to low.
- Development flexibility: It shows the level of flexibility allowed during the development process. Since the pre-established requirements are quite strict but there are no specific requirements about the architecture from external specifications, the value is set to nominal.
- Risk resolution: It indicates the team ability to react and manage risks. PowEnJoy suffers from the typical risks of a client/server application and also rely on many stakeholders involved in the process, such as the municipalities. However, most of the risks can be easily spotted and immediate reactions can be performed so the indicator is set to high
- Team cohesion: It is an indicator that reflects the ability of the team to cooperate together during the development of the system. Since the team has high experience in operating as a team and a shared vision is clear among the components the indicator is set to high.
- Process Maturity: The evaluation of this parameter is based on the KPAs

rating levels. Since the Requirements Management is frequently assessed but there is not a clear training program for individuals the EPML is estimated around 2 so a Nominal rating is given to PMAT. In fact the positive evaluation coming from the software and the requirements planning are balanced by a poor definition of the organization processes.

The results of our evaluation is the following:

Scale Driver	**Factor**	**Value**
Precedentedness(PREC)	Low	4.96
Development flexibility(FLEX)	Nominal	3.04
Risk resolution (RESL)	High	2.83
Team cohesion(TEAM)	High	2.19
Process maturity (PMAT)	Level 2	4.68
Total		17.70

2.2.2. Cost Drivers

- *Personnel capability (PERS)*

This indicator summarizes the information about the team that will work on the project and its performances. Is it clear that the main strength is the personnel continuity of the team that can help in the allocation of tasks and the internal cooperation. On the other side the team suffers from some lack especially in the programming part.

ACAP

The personnel working on the requirements and the design has great ability in the cooperation part and an ability that can be classified as an average.

PCAP

As regard the programming phase, people working on it are considered to be part of the 35^{th} percentile since the team has no significant capabilities in team programming especially if programming efficiency is considered.

PCON

Since the system will be developed by a team of three people that is not expected

to change during the development the annual personal turnover is set to high.

Sum of ACAP, PCAP, PCON Ratings	2 + 2 + 4 = 8 (Low)	Low
Combined ACAP and PCAP Percentile	45%	Low
Annual Personnel Turnover	5%	Very High
Total		1.0

- *Product reliability and complexity (RCPX)*

This indicator summarizes the level of complexity and the required level of reliability required by the system. Since the system has no specific requirements for an high level of reliability we considered the emphasis on reliability as basic (Nominal). Similarly, the product requires the implementation and the managing of the typical functionalities presents in many applications, such as user management and the administration of their requests. However, since the integration with cars may incur in some problems the product complexity is set to complex (High). Finally, the database in the mature phase of the product will have a medium amount of tuples, so the database size is set to moderate (Nominal).

RELY

Since a failure in the system can cause great inconveniences for the users but not dramatic financial loss the reliability of the system is set to Nominal. In fact, no human life is in risk in case of a problem.

DATA

To analyze the dimension of the DB we used as a metric the ratio between the

size of the DataBase, expressed in Bytes, and the program size, expressed in SLOC. Even if the size of the database is highly dependent from the number of user, in the mature phase is estimated to become significant so the indicator is set to High.

CPLX

The value is set to Nominal since the system is based on a traditional approach of technologies and implementation, however requires some efforts in the managing of the cars.

DOCU

As regard the amount of documentation the level is not significant so the level is set to Low.

Sum of RELY, DATA, CPLX, DOCU Ratings	3+4+3+2 = 12 (Nominal)	Nominal
Emphasis on reliability, documentation	Basic	Nominal
Product complexity	Moderate	Nominal
Database size	Moderate	Nominal
Total		1.00

- *Required Reuse (RUSE)*

Since the system is oriented to the realization of a specific product it does not require an high level of reusability so the indicator is set to Low.

RUSE	None	Low
Total		0.95

- *Plaform Difficulty (PDIF)*

PowerEnJoy is based on a central business logic that once implemented do not requires many major changes. However, some changes are required in regard to the mobile applications which have to follow the updates of the relative Operating System so the system is considered Stable (Nominal).

In regard to time and storage constraints the application overall is low consuming so the constraint can be considered Low.

TIME

PowerEnJoy is based on a service used by many users contemporary with a quite significant amount of request especially from cars. However, the distribution of requests during the day vary significantly so the values is set to Nominal.

Sum of TIME, STOR, and PVOL ratings	4 + 3 + 2 = 9 (Nominal)	Low
Time and storage constraint	<50%	Nominal
Platform volatility	Stable	Nominal
Total		1.00

- *Personnel Experience (PREX)*

This Early Design cost driver combines the three Post-Architecture cost drivers application experience (AEXP), platform experience (PEXP), and language and tool experience (LTEX). They are ranged in 5 categories: very Low, Low, Nominal, High, very High and extra High. As regard to the overall experience expressed in month the value individuated is around 9 month so a Low value.

AEXP

The team, has very low experience with this type of application so a very low value is selected.

PEXP

Our team has low experience with Java Enterprise Edition. Despite that, in the

last years we practiced with the use of DataBase (SQL), Java language, HTML and Web Servers. For this reason, we will not expect to find big difficulties using JEE. A Nominal value is selected.

LTEX

As said for "Platform experience" we are non-expert in JEE, but we experienced some other related things. So, we have the knowledge of the programming environment and there shouldn't be any kind of hurdle in learning these new Tools. A Nominal value is selected.

Sum of AEXP, PEXP, and LTEX ratings	1 + 3 + 3 = 7 (LOW)	Low
Applications, Platform, Language and Tool Experience	9 month	Low
Total		1.12

- *Facilities (FCIL)*

This early design cost driver combines the two Post-Architecture of TOOL and SITE. Both of them are ranged in 5 categories: very Low, Low, Nominal, High, very High and extra High.

TOOL

The usage of software tools to code, edit or management. This project utilized only the basic tools, but the environment is well integrated and professional. A high value is selected.

SITE

The project was developed entirely using a lot of interactive multimedia provided

by internet, in order to be consistent in all the project phases. A very high evaluation can be assigned for this value.

Sum of TOOL and SITE ratings	4 + 5 = 9 (Very High)	Very High
TOOL support	Good	High
Multisite conditions	Strong support of simple M/S devel	Very High
Total		0.73

- *Schedule (SCED)*

This value measures the schedule constrains imposed on the project team. Considering the number of hours spent on this project, the deadlines and also the efforts, this parameter is set to high. The starting phase and the developing of documentation required a lot of time.

SCED	130%	High
Total		1.00

Overall the EAF are:

Personnel capability (PERS)	Low	1.00
Product reliability and complexity	Nominal	1.00
Required Reuse (RUSE)	Low	0.95
Plaform Difficulty (PDIF)	Nominal	1.00

Personnel experience (PREX)	Low	1.12
Facilities (FCIL)	Very High	0.73
Schedule (SCED)	Nominal	1.00
Total		0.7767

2.2.3. Effort equation

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

$$Effort = A^* EAF * KSLOC^E$$

A = 2.94

EAF = product of all cost drivers

E = exponent derived from the scale drivers. It is calculated as:

$B + 0.01 * \sum SF[i] = B + 0.01 * 17.70 = 0.91 + 0.177 = 1.087$

$$Effort = A * EAF * KSLOC^E = 2.94 * 0.7767 * 7.4^{(1.087)} = 20.11 \text{ PM}$$

2.2.4 Schedule estimation

In order to calculate the duration expressed in months we use the formula:

$$Duration = 3.67 * Effort^f$$

$F = D + 0.2 * 0.01 * \sum SF[i] = D + 0.2 * (E - B) = 0.28 + 0.2 * (1.087 - 0.91) = 0.3154$

$$Duration = 3.67 * 20^{0.31} = 9.28 \text{ month}$$

3. Project Scheduling

3.1. Tasks Identification

In this paragraph, we are going to illustrate all the tasks of the project.
We will stop developing after the project plan, but in this document, we consider all the missing steps, too.
This is just a guideline since the project could be modified or we could introduce new requirements and functionalities.

There are the main tasks:
[T1]: Project Plan
[T2]: Requirements Analysis and Specification Document (RASD)
[T3]: Design Document
[T4]: Integration Test Plan
[T5]: Project Implementation
[T6]: Unit Test
[T7]: Integration Test
[T8]: Deliver and test a Beta Release
[T9]: Final Release

In the table below, we present the task interdependencies:

Task	Start	Deadline	Dependencies
T1	17/10/2016	23/10/2016	--
T2	17/10/2016	13/11/2016	--
T3	14/11/2016	11/12/2017	T1
T4	12/12/2016	15/01/2017	T2
T5	16/01/2017	16/05/2017	T2,T4
T6	17/05/2017	23/05/2017	T5,T4
T7	24/05/2017	30/05/2017	T5,T6
T8	31/05/2017	06/06/2017	T5,T6,T7
T9	07/06/2017	20/06/2017	T8

The entire duration is about 9 months according to COCOMO II analysis.
Some deadlines were already fixed and the others are estimated on previous experience.

3.2. Tasks Allocation

For each component of the group, we indicate his development work.

In this tables, there are some tasks that required more than one person or more than one week.

Project Plan (17/10/16-23/10/16)	
	1st week
Ivan	Introduction Project size Risk Management
Lorenzo	Introduction Project size Cost and effort estimation
Martina	Introduction Project size Task allocation

RASD (17/10/16-13/11/16)				
	1st week	2nd week	3rd week	4th week
Ivan	Description of the problem Proposed system	Functional Requirements Non-functional requirements Actor: Registered user Actor: Employee	Domain model	Minor fix
Lorenzo	Goals Domain properties Assumptions	Functional Requirements Actor: Unregistered user Actor: Car	Glossary Tools Domain model	Minor fix
Martina	Possible scenarios Actors	Non-functional requirements Actor: Unregistered user Actor: Car Actor: Registered user Actor: Employee	Alloy Code Generated world	Minor fix

Design Document (14/11/16-11/12/16)				
	1st week	2nd week	3rd week	4th week
Ivan	Scope Overview References High level components	Algorithm design Deployment view Selected architectural styles and patterns User interface Component interfaces	Design overview User experience Runtime view	Further preview of UI
Lorenzo	Scope Overview Glossary	Algorithm design Deployment view	Component interfaces Runtime view	Requirements traceability

	High level components	Component diagram: Web Service Selected architectural styles and patterns		
Martina	Purpose Scope Overview High level components	Algorithm design Deployment view Component diagram: User Management	User navigation flow E-R Diagram	Relational model

Integration Test Plan (12/12/16-15/01/17)					
	1st week	2nd week	3rd week	4th week	5th week
Ivan	Entry Criteria	Elements to be integrate Integration testing strategy	Business logic test	Subcomponents and subsystems Subsystem test Integration sequence	Fixing DD for consistency purpose
Lorenzo	Purpose and scope	Elements to be integrated	Business logic test Car interface test	Subcomponents and subsystems Subsystem test	Program stubs Tools and test equipment
Martina	Glossary	Elements to be integrated	Persistence module test	Subcomponents and subsystems integration sequence	Test data

Project Implementation (16/01/17-16/06/17)				
	1st month	2nd month	3rd month	4th month
Ivan	Web Tier	Business Tier	Persistence Module	External Components Interfaces
Lorenzo	Web Tier	Business Tier	Persistence Module	External Components Interfaces
Martina	Web Tier	Business Tier	Persistence Module	External Components Interfaces

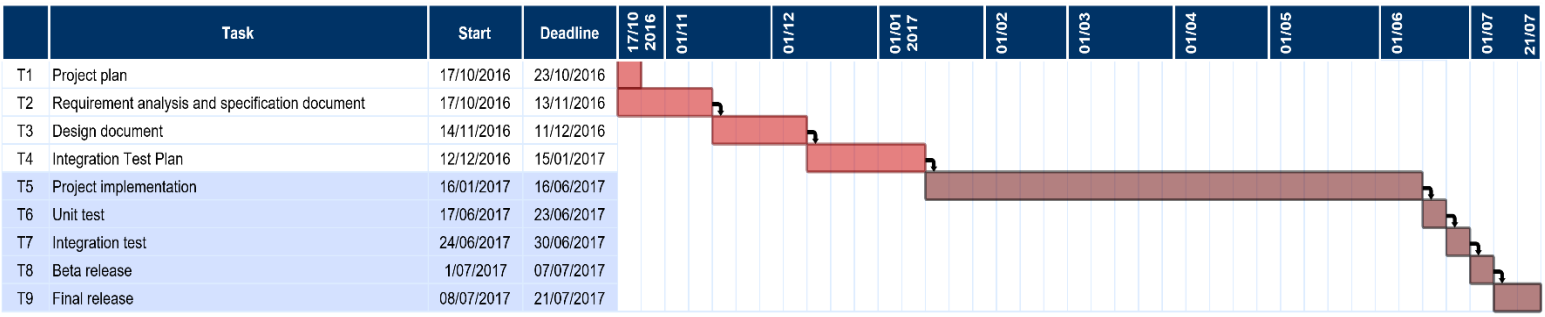
Unit Test (17/06/17-23/06/17)	
	1st week
Ivan	Unit Testing
Lorenzo	Unit Testing
Martina	Unit Testing

Integration Test (24/06/17-30/06/17)	
	1st week
Ivan	Integration Testing
Lorenzo	Integration Testing
Martina	Integration Testing

Beta Release (01/07/17-07/07/17)	
	1st week
Ivan	Beta Release Testing
Lorenzo	Beta Release Testing
Martina	Beta Release Testing

Final Release (08/07/17-21/07/17)		
	1st week	2nd week
Ivan	Analyzing Beta testing results and revisions/changes	Adjustments and final release
Lorenzo	Analyzing Beta testing results and revisions/changes	Adjustments and final release
Martina	Analyzing Beta testing results and revisions/changes	Adjustments and final release

Here below there is a Gantt graph that illustrate all the tasks, the order we followed to complete them and for each task the start and the deadline.



4. Risk Analysis

With the risk analysis we are going to explain the several issues which may affect the software and their functionalities. In order to define the different kind of risks which could occur, they have been ranged in 3 different categories: project risks, technical risks and business risks.

These categories should cover all the environment which are touched by our software. It is also provided a small table, in which it is described the probability of the issue might happen and the effects. A brief description related the way to overcome the possible issue, is explained under the table.

4.1. Project Risks

Effects	Probability	Risk
Serius	Moderate	Requirement Change
Catastrophic	Low	Lack of experience

Medium	Low	Staff absence during important dates

Requirement Change

Description: It is an unpredictable risk that could provoke the entire restructuration of the system. For instance, the staff misunderstands the customer's needs and requests.

Strategy: It can be managed using much reusable code as possible and modularity. Maybe it is also important keep updated the customer about the project progresses.

Lack of experience:

Description: during the implementation, it is possible to find a programmer unable to develop some components. That will cause a delay on the deliveries since the team member has to update its knowledge

Strategy: It is important to ensure that all the member selected for the project are able to reach up their tasks assigned.

Staff absence during important dates:

Description: It is possible that for illness or other reasons, any member of the group is not able to work or attend to some important dates

Strategy: All the teammates should now the progresses and tasks completed of all the infrastructure.

4.2. Technical Risks

Effects	Probability	Risk
Catastrophic	Low	DataBase performance
Serius	Moderate	Server failures
Catastrophic	Low	Lack of data
Serius	Moderate	Defective components
Catastrophic	Low	Change of external API

Database performance:

Description: The database cannot stand all the transactions due to exceeded number of users. That may occur because the requests to the server overcome the resources allocated for the infrastructure.

Strategy: The strategy to resolve this risk is to improve the database layer for instance buying a higher performance database from the supplier for supporting the older one.

Server failures:

Description: This kind of risk is unpredictable, and it causes the impossibility to reach the server functionalities and also the data.

Strategy: As said it is an unpredictable event, in order to avoid that the staff should analyse period lags and keep updated the software.

Lack of data:

Description: The lack of data, like the server failures is unpredictable. This issue may happen in any possible situation.

Strategy: In order to avoid these situations a database backup should be done frequently.

Defective components:

Description: Some components may be corrupted or break, they need to be repaired during the normal activities.

Strategy: replace damaged components with new ones. In these case an employee of the company will be in charge of this mansion.

Change of external API

Description: The vendors of an external service may change the API of an application which PowerEnJoy rely on.

Strategy: In this case a timely operation provided by the employees of PowerEnJoy in order to repair the software will be necessary. Nowadays the API are more or less all standardized.

4.3. Business Risks

Effects	Probability	Risk
Medium	Moderate	Change of cloud prices
Catastrophic	Low	Change of governament

		laws
Serius	Moderate	Competitors
Catastrophic	Low	Unused Product
Serius	Moderate	Underscore overall costs

Change of cloud prices:

Description: The cloud infrastructure which provides the services, may change their costs. This may occur without any possibility of forecasting. It would be better if this change happens during the initial stages, in which there is still the possibility to choose the supplier.

Strategy: In this case the costs are more if this change occurs during the production phase and not during the planning phase. A new business plan will choose the better solution for the company for saving money.

Change of government laws:

Description: In this case, the change of government laws may cause some troubles for the budget area. That because environmentally friendly projects all always supported by the government.

Strategy: In this case, the new laws take months to be approved. It will be necessary to move fast during this month to limit the impact of the new laws.

Competitors:

Description: The main business risk is the possibility that another company develops a similar application.

Strategy: This is unpredictable and unavoidable; the only possible strategy is to make the product the better as possible and making forecasting prevision about the competitors inside our arena.

Unused Product:

Description: the risk is that our product will not be used due to the existence of parallel services already commonly used.

Strategy: The strategy to avoid this risk is to implement specific functionalities that are exclusive of our system such as: low costs, environmentally friendly campaign and others. Gathering customer's opinion related to this projects, will be necessary in order to measure the demand.

Underscore overall costs:

Description: The business department may make a mistake in forecasting the overall costs. And there could be some fund issues.

Strategy: This is an unpredictable problem. In these cases, cutting down some other costs may help developing of the projects.

5. Hours of work

To redact this document, we spent 20 hours per person.