Politecnico di Milano

PowerEnjoy Service - Project Plan Document

January 20, 2017

Version 1.1

Authors:

- Domenico FAVARO (Mat. 837995)
- Matheus FIM (Mat. 876069)
- Caio ZULIANI (Mat. 877266)

Prof. Elisabetta DI NITTO

Contents

1 Introduction

1.1 Revision History

This section records all revisions to the Document.

Version	Date	Authors	Summary
1.1	22/01/16	Domenico Favaro, Caio Zuliani, Matheus Fim	Initial Release

1.2 Purpose and Scope

The Project Plan Document (PPD) is fundamental to organize the production of the System. The main purpose of the Document is to study the complexity of the System, analize the difficulties that can be present in any stage of its development and deployment and give an overall estimation of the effort and cost that can be needed up to the final release of the product. The analysis will be done in terms of size, time and budget needed. This is fundamental to present our stakeholders a preview before the start of production in case they need to do any project changing decision before hand.
The second part of the Document helps the organization of our production by giving time estimations for each step and managing the workload of our resources accordingly. This serves our production team to make any project management decision but also our development team to know how they can organize themselves to follow the schedule and if by any reason the schedule can't be followed, they can notify the production. This is fundamental to keep track of the project developing at every stage of production.
Finally we present possible risks that can arise during the development and for each risk we have identified a contingency plan to avoid or minimize the possibility of harming the production.

1.3 Definitions and Abbreviations

- **RASD:** Reqirements And Specifications Document.
- **DD:** Design Document.
- **ITPD:** Integration Test Plan Document.
- **PPD:** Project Plan Document
- **App:** Application, refering to Web or Mobile App.
- **LOC:** Lines of Code, is a software metric used to measure the size of a computer program by counting the number of lines in the text of the program's source code.

- **FSM:** Functional Size Measurement, technique for measuring software in terms of the functionality it delivers.
- **FP:** Function Points. Unit of measurement used to express the amount of functionality for an information system. We'll be using them to compute a FSM of software.
- **VAF:** Value Adjustment Factor, used to adjust the Function Points estimation to the context of the project.
- **UFP:** Unadjusted Function Points, raw measurement of the total FP before the VAF is applied.
- **ILF:** Internal Logic Files, FP Function Type.
- **EIF:** External Interface Files, FP Function Type.
- **EI:** External Input, FP Function Type.
- **EO:** External Output, FP Function Type.
- **EQ:** External Inquiries, FP Function Type.
- **COCOMO:** Constructive Cost Model is a procedural software cost estimation model that we'll be using for this document.

For other concepts concerning the Service definition look in the **Glossary** section of the RASD, DD and ITPD.

1.4 Reference Documents

- Specification Document: Assignments AA 2016-2017.pdf
- PowerEnjoy Requirements And Specifications Document (RASD)
- PowerEnjoy Design Document (DD)
- PowerEnjoy Integration Test Plan Document (ITPD)
- Example Document - Project planning example document.pdf
- Project Planning Tools Documentation:
  - Function Points Table
    - `http://www.qsm.com/resources/function-point-languages-table`
  - COCOMO II - Manual
    - COCOMO Model Definition Manual
  - Project Management Software: Jira/Trello
    - `https://www.atlassian.com/software/jira`
    - `https://trello.com/`

2 Project Size, Cost and Effort Estimation

This section deals with the estimation of the expected size, cost and effort needed to develop the PowerEnjoy Service. For the size estimation we'll use the Function Points approach, once calculated we'll have an approximated size in LOC, and this measurement will be used later with COCOMO to predict the cost and amount of effort that will be required to develop the program.

2.1 Size Estimation: Function Points

The Function Mode approach is based in giving to each function needed in the code of the program a Category and a Complexity, once this is done, using Function Point Tables, a number of UFP is given to them. The VAF is then applied to the total of UFP. After the Total FP are calculated, it can then be translated into LOC. The Categories for FP are 5, and each has its own rating table for complexity according to the number of Data Elements present:

- Internal Logic Files (ILF):

  Translate to the sets of data used and managed by the application.

- External Interface Files (EIF):

  Similar to ILF set of data used by the application but generated and maintained by other applications.

  Their complexity table is the same as ILF.

Record Types	**1-5**	**6-19**	**> 19**
0-1	Low	Low	Avg
2-3	Low	Avg	High
> 3	Avg	High	High

- External Inputs (EI):

  Defined as operations to receive and manage data coming from the external environment.

File Types	**1-4**	**5-15**	**>15**
0-1	Low	Low	Avg
2	Low	Avg	High
> 2	Avg	High	High

- External Outputs (EO):

  They are elementary operations that generates data for the external environment. It usually includes managing data from logic files

- External Inquiries (EQ):

  Operation that involves both input and output, managing the data themselves without significant intervation from logic files.

  Their complexity table is the same as EO.

File Types	**1-5**	**6-19**	**>19**
0-1	Low	Low	Avg
2-3	Low	Avg	High
> 3	Avg	High	High

To measure the complexity in FP, weights are given to each Function Type using the following table:

Function Type	**Low**	**Avg**	**High**
ILF	7	10	15
EIF	5	7	10
EI	3	4	6
EO	4	5	7
EQ	3	4	6

2.1.1 Internal Logic Files (ILF)

The PowerEnjoy System is organized to store a number of information to provide its functionalities. This information corresponds to the Entities that compose the System and will be manage by the Entity Beans in the Controllers *(see DD)*. The ILF will then correspond to managing the information of these entities, which are:

- Cars: Cars will be stored in the system DB and replicated by persistent entities in the JEE2 environment for managing their status. Their information include CarID, Location, Status, BatteryStatus, if it's PluggedIn and Number of Passengers.

  Car element count is 6 Fields.

- Users: Users tuples will be added by the System in the DB as well for the Registration feature, their attributes include UserID, if it's LoggedIn, UserName, Password, Email, DriverLicense, PaymentInfo and Location.

  User count is 8 Fields.

- CRM: Although not directly added by the System, the loggedIn status of the CRMs is important for the Chat feature. It stores their CRMID, Password, Name, Email and if it's LoggedIn.

  CRM count is 5 Fields.

- Reservations: Will link Users with Cars and will have a status to know wether the reservation is active or not. Their attributes are ReservationID, UserID, CarID, Status, ReservationTime, PickUp-Time.

  Reservation elements count is 6 and references 2 Record Types.

- Rides: Once the Reservation has been confirmed a Ride is created, it refers to the Reservation and it tracks the use of the Car. It has RideID, ReservationID, CarID, Status, StartTime, EndTime, LocationPath.

  Ride count is 7 and references 2 Record Types.

- Payments: When the Ride is finished it generates the Payment for the User to pay, it tracks the input made by the Car, Rides and Users and include also eventual Discounts and Extra Fees. It has PaymentID, RideID, TotalFee, PassengerDiscount, BatteryDis-count, PluggedInDiscount, LowBatteryFee, NoPlugNearFee.

  Payment has 8 fields, references 1 Record Type.

- User Reports: Tracks any fault in the Service by Users or Cars and are handled by CRMs. They have the following fields: UserRepor-tID, UserID, CRMID, CarID, Description.

  User Report has 5 fields, references 3 Record Types.

- Locations: Although not managed by the System, the record of safe parking and plug in areas has to be kept and accessed by the LocationHelper. They have basic Info like LocationID, Latitude, Longitude, Type.

  Locations have 4 fields.

Using the Complexity Table for the ILF we obtain the following.

ILF	Complexity	FP
Car Data	Low	7
User Data	Low	7
CRM Data	Low	7
Reservations	Avg	10
Rides	Avg	10
Payments	Low	7
User Reports	Low	7
Locations	Low	7

The Total UFP for ILF is **62**.

2.1.2 External Interface Files (EIF)

Our only external data source comes from Google Maps API, that offers the map service. It's used by the Location Helper and by the User and CRM Interface Components. It'll be accessed to retrieve the map interface find directions and show the location of the User and the Cars, as well as to retrieve walking and driving distances. The response is given in JSON and it provides numerous fields, however it doesn't reference any record types so we count only the number of parameters the response gives us.

EIF	Complexity	FP
API Map Retrieval	Avg	10
Direction Retrieval	Low	7

This gives us that the UFP for EIF are**17**.

2.1.3 External Inputs (EI)

The Sytem relies on many interactions provided for the User to allow them the functionalities of the service, others involve or are shared by the CRM, we'll present all of them and give a complexity estimation for each one. They consist on a function activated by the User Input.

Shared:

- Login: Controls that the parameters are correct and logs in the User. Low Complexity.
- Logout: Signs out the connected User. Low Complexity.
- Reserve Car: Creates a new Reservation. It involves User, Car and Reservation. Complexity Average.

- Cancel Reservation: Cancels already created Reservation, interacts just with User and Reservation. Complexity Low.

Users:

- Register: Creates a new User and inserts his/her data in the system, has many fields but concers only the User. Complexity is Low.
- Confirm Reservation: Unlocks the Car and starts a new Ride. Average Complexity.
- End Ride: Finishes current Ride and creates a new Payment. Complexity is Average.

CRM:

- Create User Report: Links User, CRM and Car in case of any issue. Average Complexity.
- Lock Car: Signals the Car to Lock. Low Complexity.
- Unlock Car: Low Complexity.

EI	**Complexity**	**FP**
Login	Low	3 x2
Logout	Low	3 x2
Register	Low	3
Reserve Car	Avg	4 x2
Cancel Reservation	Low	3 x2
Confirm Reservation	Avg	4
End Ride	Avg	4
Create User Report	Avg	4
Lock Car	Low	3
Unlock Car	Low	3

This gives us a Total UFP for the EI of **47**.

2.1.4 External Outputs (EO)

PowerEnjoy External Outputs are limited to sending Emails in case of User Registration to send the password, End of Ride to send the Payment and Time Limit over the Reservation to notify the User.

- Email Password: Involves just the User, Low Complexity.
- Email Payment: Sends Payment to User, Low Complexity.
- Send Late Reservation Notification: Notifies status of Reservation to User, Low Complexity.

EO	**Complexity**	**FP**
Notify Password	Low	4
Notify Payment	Low	4
Notify Reservation	Low	4

The Total UFP for the EO is **12**.

2.1.5 External Inquiries (EQ)

Similar to EI but sends a data response instead for the User. Basically a query of our system. PowerEnjoy EQ are:

- Search Nearby Cars: Gives a List of close by avaliable Cars from the User Location. Has Low Complexity.
- Money Saving Option: Calculates the best Re-Charging station according to a standard Deviation Algorithm (*see DD*) and returns it to the User so a discount for the ride can be applied. Is of Average Complexity.
- Find Location: Gives a Path to arrive to the given Location.
- Find Cars: Returns all desired Cars for the CRM.
- Find Users: Returns all Users for the CRM.
- Find Reservations: Returns desired Reservations for the CRM.
- Find Rides: Returns all Rides for the CRM.
- Find Payments: Returns given Payments for the CRM.
- Find User Reports: Returns the User Reports for the CRM.

All the Finding Functions are considered to have Low Complexity.

EQ	**Complexity**	**FP**
Search Nearby Cars	Low	3
Money Saving Option	Avg	4
Find Location	Low	3
Find Cars	Low	3
Find Users	Low	3
Find Reservations	Low	3
Find Rides	Low	3
Find Payments	Low	3
Find User Reports	Low	3

This gives us a Total UFP for the EQ of **28**.

2.1.6 Overall Estimation

The following table shows the estimation results for each Function Type:

Function Type	**Value**
ILF	62
EIF	17
EI	47
EO	12
EQ	28

The added Total of UFP is **166**.

To have an idea of the size of our project in the code we translate the total of UFP to LOC using the Function Points Languages Table. Since we'll develop in the J2EE platform the measures are $Median = 49, High = 67$.

$$LowBoundLOC = 166 * 49 = \mathbf{8134}$$

$$HighBoundLOC = 166 * 67 = \mathbf{11122}$$

We can expect a software ranging from the size of 8200 to 11150 LOC.

2.2 Cost and Effort Estimation: COCOMO II

In the estimation of the cost and effort needed in the development of the PoweEnjoy Service it will be employed the COCOMO II.

2.2.1 Scale Drivers

In the evaluation of the Scale Drivers it was used the following table provided by the COCOMO II approach.

Scale Factors	**Very Low**	**Low**	**Nominal**	**High**	**Ver**
PREC SFj	thoroughly unprecedented 6.20	largely unprecedented 4.96	somewhat unprecedented 3.72	generally familiar 2.48	larg 1.42
FLEX SFj	rigorous 5.07	occasional relaxation 4.05	some relaxation 3.04	general conformity 2.03	som 1.01
RESL SFj	little (20%) 7.07	some (40%) 7.07	often (60%) 4.24	generally (75%) 2.83	mos 1.41
TEAM SFj	very difficult interactions 5.48	some difficult interactions 4.38	basically cooperative interactions 3.29	largely cooperative 2.19	high coop
PMAT SFj	SW-CMM Level 1 Lower 7.80	SW-CMM Level 1 Upper 6.24	SW-CMM Level 2 4.86	SW-CMM Level 3 3.12	SW- 3 1.5

Analysing each point that compose the Scale Drivers:

- Precedentedness: It's about if the team/organization has a previous experience with this type of project. As the team don't have a large professional experience the item will be evaluate as low.
- Development flexibility: It's about the degree of flexibility in the development process. In this project there were several specifications for the system when becomes of functionalities thus the item will be evaluate as low.
- Risk resolution: It's about the risk analysis made. It was carried out several risk evaluations during the differents parts of the project so the item will be evaluated as high.
- Team cohesion: It's the evaluation of how well the development team knows each other and work together. As the team has achieved a excellent level of work since the division of the workload until the respect to the deadlines the item will be evaluated as very high.
- Process maturity: It's about the process maturity of the organization/team that is know as CMM. Considering that it's the team first project of this kind especially involving Java EE the item will be evaluated as Level 2. Moreover it's important to emphasize the proactive of the group even if it's not characteristic of the evaluation of Level 2 of CMM.

Thus the evaluation is:

Scale Driver	**Factor**	**Value**
Precedentedness (PREC)	Low	4.96
Risk resolution (RESL)	High	2.83
Team cohesion (TEAM)	Very high	1.10
Process maturity (PMAT)	Level 2	4.68
Total		13.57

2.2.2 Cost Drivers

- Required Software Reliability (RELY): Being the measure of the "extent to which the software must perform its intended function over a period of time" the RELY will be set as high because as the main business is renting the cars through the app, an interruption of the service means directly financial losses. Moreover customers will expected a reliable service and any interruption can cause unnecessary instabilities.

Required Software Reliability (RELY)						
RELY Descriptors	Slight Inconvenience	Low, easily recoverable losses	Moderate, easily recoverable losses	High financial loss	Risk to human life	
Rating	Very Low	Low	Nominal	High	Very High	Extra High
Effort multipliers	0.82	0.92	1.00	1.10	1.26	n/a

- Database size (DATA): It's possible to estimate the database size according to the COCOMO II with the D/P ratio(DB bytes/LOC). Considering such variables as the tables, fields and their respective growing it was estimated a 4 GB database. As the LOC is the range of 8200-11150 it is obtained a DP of the range of 386 to 524. Thus the DATA is evaluated as high.

Database size (DATA)						
DATA Descriptors		D/P<10	10 < D/P <100	100 < D/P <1000	D/P >1000	
Rating	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): In accordance with the COCOMO II standards for the evaluation the CPLX, whose is composed by five parameters, the rating level in our system was evalueted as very high. It was concluded that mostly because of the Computational Operations component due the fact that of our system has more complexes numerical analyses such as the partial differential equations used in the Standard Deviation Algorithm of the Money-Saving option.

Product complexity (CPLX)						
Rating	Very Low	Low	Nominal	High	Very High	Extra High
Effort multipliers	0.73	0.87	1.00	1.17	1.34	1.74

- Required reusability (RUSE): The main requirements of the project don't established the reuse of the code as a goal so the evaluation of this cost driver will be as nominal.

Required reusability (RUSE)						
RUSE Descriptons		None	Across project	Across program	Across product line	Across multiple product lines
Rating	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 DOCU evaluate how adapted the project's documentation is in relation with its life-cycle needs. As the project documentation was up to date with the life-cycle needs the evaluation will be set as nominal.

Documentation match to life-cycle needs (DOCU)						
DOCU Descriptons	Many life-cycle needs uncovered	Some life-cycle needs uncovered	Right-sized to life-cycle needs	Excessive for lifecycle needs	Very excessive for life-cycle needs	
Rating	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): Being the measure of the use of the CPU processing power, regarding the total capacity available it was estimated that as very high due the fact that the software has multiple functions such as keeping track of the rides, cars and users requests. Furthermore the system need to be available in a full regime because of it's real time nature.

Execution time constraint (TIME)						
TIME Descriptons			<= 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	Very Low	Low	Nominal	High	Very High	Extra High
Effort multipliers	n/a	n/a	1.00	1.11	1.29	1.63

- Storage constraint (STOR): The STOR evaluates the main storage constraint that is imposed in one software. As this constraint is considered as of easily solution nowadays, this cost driver was set as nominal.

Storage constraint (STOR)						
STOR Descriptons			<= 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	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 PVOL measure the volatility in terms of updates in the system per time. As the system is structured in 3 tiers that are Client Tier(Web App, Mobile App, Car), Server Tier (Java EE Web and Application Server), EIS Tier (DataBase Server) the PVOL will be evaluated as nominal because it's not expected major changes when becomes of the Server and EIS tier. But it is expected that the Client Tier will required changes due the volatility in the smartphone market where there are several new occurrences such as system updates and launching of new devices in a short period of time.

Platform Volatility (PVOL)						
PVOL Descriptons		Major change every 12 month, minor change every 1 month.	Major change every 6 month, minor change every 2 weeks.	Major change every 2 month, minor change every 1 week.	Major change every 2 weeks, minor change every 2 days.	
Rating	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 ACAP measure the level of ability, efficiency and communication skills of an Analyst. It will be evaluated as high due the fact that the team has developed an extensive work of analyses of how to bring into the software the functions that were required by the stakeholder.

Analyst Capability (ACAP)						
ACAP Descriptons	15th percentile	35th percentile	55th percentile	75th percentile	95th percentile	
Rating	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 PCAP is based on the measure of the capacity of the programmers as a team estimating their ability, cooperation and communication. As the project was not implemented we estimated this evaluation having as a parameter the work developed until now and with the personal abilities of the components of the team. Therefore the PCAP will be set as high.

Programmer Capability (PCAP)						
PCAP Descriptons	15th percentile	35th percentile	55th percentile	75th percentile	90th percentile	
Rating	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 personnel continuity factor (PCON) measures the annual personnel turnover. As in the stipulated time of the project the group did no have turnovers the PCON will be evaluated as very high.

Personnel Continuity (PCON)						
PCON Descriptons	48% per year	24% per year	12% per year	6% per year	3% per year	
Rating	Very Low	Low	Nominal	High	Very High	Extra High
Effort multipliers	1.29	1.12	1.00	0.90	0.81	n/a

- Application Experience (APEX) :The APEX evalueted the experience of the team with the application. As it is the first application of the king that the team develop it will be evaluated as low .

Application Experience (APEX)						
APEX Descriptons	<= 2 months	6 months	1 year	3 year	6 year	
Rating	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) : Being the rating of the experience of the team with the platform, in the case Java EE, the PLEX will be evaluated as low although the team has had a previous experience with Java especially the SE, databases.

Platform Experience (PLEX)						
PLEX Descriptons	<= 2 months	6 months	1 year	3 year	6 year	
Rating	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 LTEX measure the experience of the team with the language and tools used for the development of the project. Being the first time of the group working with Java EE the evaluation will be low. Although the members of the group have some previous experience with java and databases.

Language and Tool Experience (LTEX)						
LTEX Descriptons	<= 2 months	6 months	1 year	3 year	6 year	
Rating	Very Low	Low	Nominal	High	Very High	Extra High
Effort multipliers	1.20	1.09	1.00	0.91	0.84	n/a

- Use of Software Tools (TOOL):The TOOL try to scale the use of software tools during the developing of software. Due the availability a strong platform for the development of the project the TOOL will be evalueted as high

Use of Software Tools (TOOL)						
TOOL Descriptons	Edit, code, debug	Simple, frontend, backend CASE, little integration	Basic lifecycle tools, moderately integrated.	Strong, mature lifecycle tools, moderately integrated	Strong, mature, proactive lifecycle tools, well integrated with processes, methods, reuse.	
Rating	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 SITE try to evaluete the effects of the multisite development in a project. The two main factor of this cost driver are site collocation and communication support. The team is located in the same city and depends upon the communication via social media. Thus the SITE evaluation will be high.

Multisite development (SITE)						
SITE Collocation	International	Multi-city and multi-company	Multi-city or multi-company	Same city or metro area	Same building or complex	Fully collocated
SITE Communications	Some phone, mail.	Individual phone, FAX.	Narrowband email.	Wideband electronic communication.	Wideband electronic communication.	Interactive multimedia.
Rating	Very Low	Low	Nominal	High	Very High	Extra High
Effort multipliers	1.22	1.09	1.00	0.93	0.86	n/a

- Required development schedule (SCED): Being the rate of schedule stretch-out or acceleration regarding a nominal amount of effort in the development of a project the SCED will be evaluated as nominal due the fact that the efforts in the development of the project were well distributed among the members of the group given the corresponding allocated time.

Required development schedule (SCED)						
SCED Descriptons	75% of nominal	85% of nominal	100% of nominal	130% of nominal	160% of nominal	
Rating	Very Low	Low	Nominal	High	Very High	Extra High
Effort multipliers	1.43	1.14	1.00	1.00	1.00	n/a

Therefore the results of the previous analyzes are expressed in the following table:

Cost Drive	**Factor**	**Value**
Required Software Reliability (RELY)	High	1.10
Database size (DATA)	High	1.14
Product complexity (CPLX)	Very High	1.34
Required reusability (RUSE)	Nominal	1.00
Documentation match to life-cycle needs (DOCU)	Nominal	1.00
Execution time constraint (TIME)	Very High	1.29
Storage constraint (STOR)	Nominal	1.00
Platform Volatility (PVOL)	Nominal	1.00
Analyst Capability (ACAP)	High	0.85
Programmer Capability (PCAP)	High	0.88
Personnel Continuity (PCON)	Very High	0.81
Application Experience (APEX)	Low	1.10
Platform Experience (PLEX)	Low	1.09
Language and Tool Experience (LTEX)	Low	1.09
Use of Software Tools (TOOL)	High	0.90
Multisite development (SITE)	High	0.93
Required development schedule (SCED)	Nominal	1.00
Total		1.4366

2.2.3 Effort Equation

In order to obtain the effort estimation with units Person-Months(PM) it will be need the following formulas:

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

$$E = B + 0.01 * \sum_{i=1}^{4} SF[i]$$

where the parameters are:

A=2.94
B=0.91
EAF = Product of all cost drivers = 1.4336
E = Exponent derived from the scale drivers=0.91+0.01*13.57=1.0457
LowBoundKSLOC=8.134
HighBoundKSLOC=11.122

Thus for the lower bound of the KSLOC was obtained: and for the

$$Effort = 2.94 * 1.4336 * 8.134^{1.0457} = 37.73PM \approx 38PM$$

high bound of the KSLOC:

$$Effort = 2.94 * 1.4336 * 11.122^{1.0457} = 52.33PM \approx 52PM$$

2.2.4 Schedule Estimation

In terms of the schedule the following formulas will be need:

$$Duration = 3.67 * Effort^{F}$$

$$F = 0.28 + 0.2 * (E - B) = 0.28 + 0.2(1.0457 - 0.91) = 0.30714$$

The schedule considering the lower boud is:

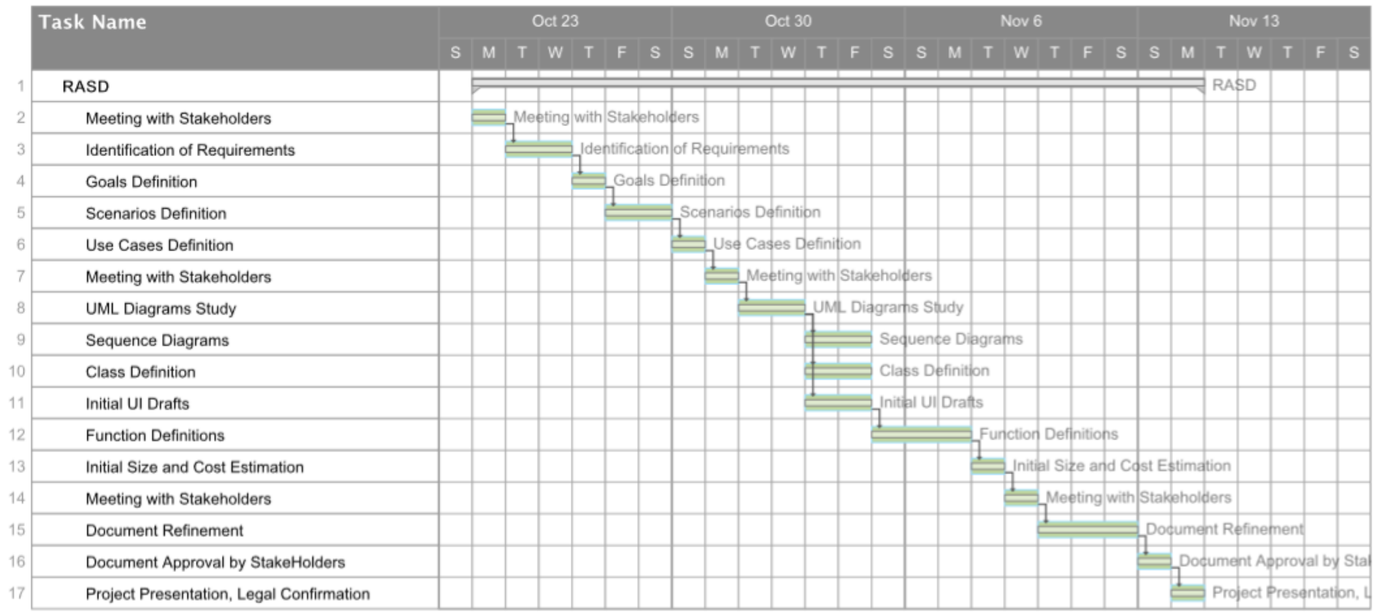
$$Duration = 3.67 * 37.73^{0.30714} = 11.19months$$

and the schedule considering the higher bound is:

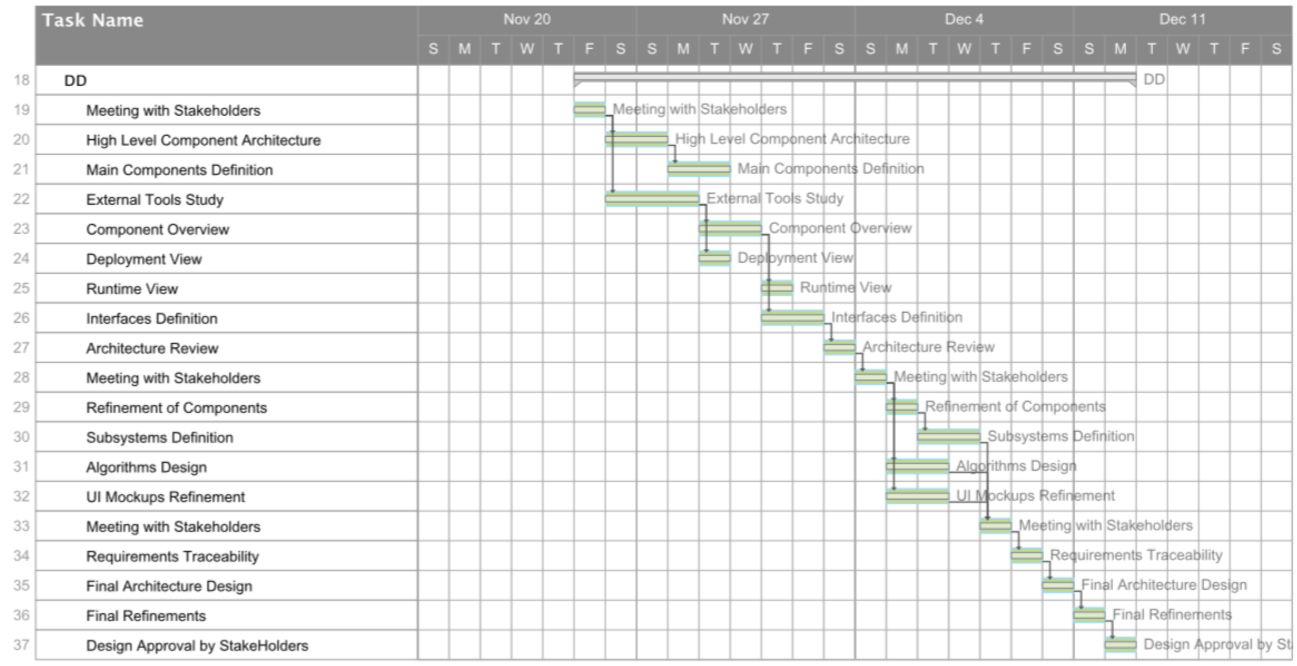
$$Duration = 3.67 * 52.33^{0.30714} = 12.37months$$

3 Project Plan Schedule

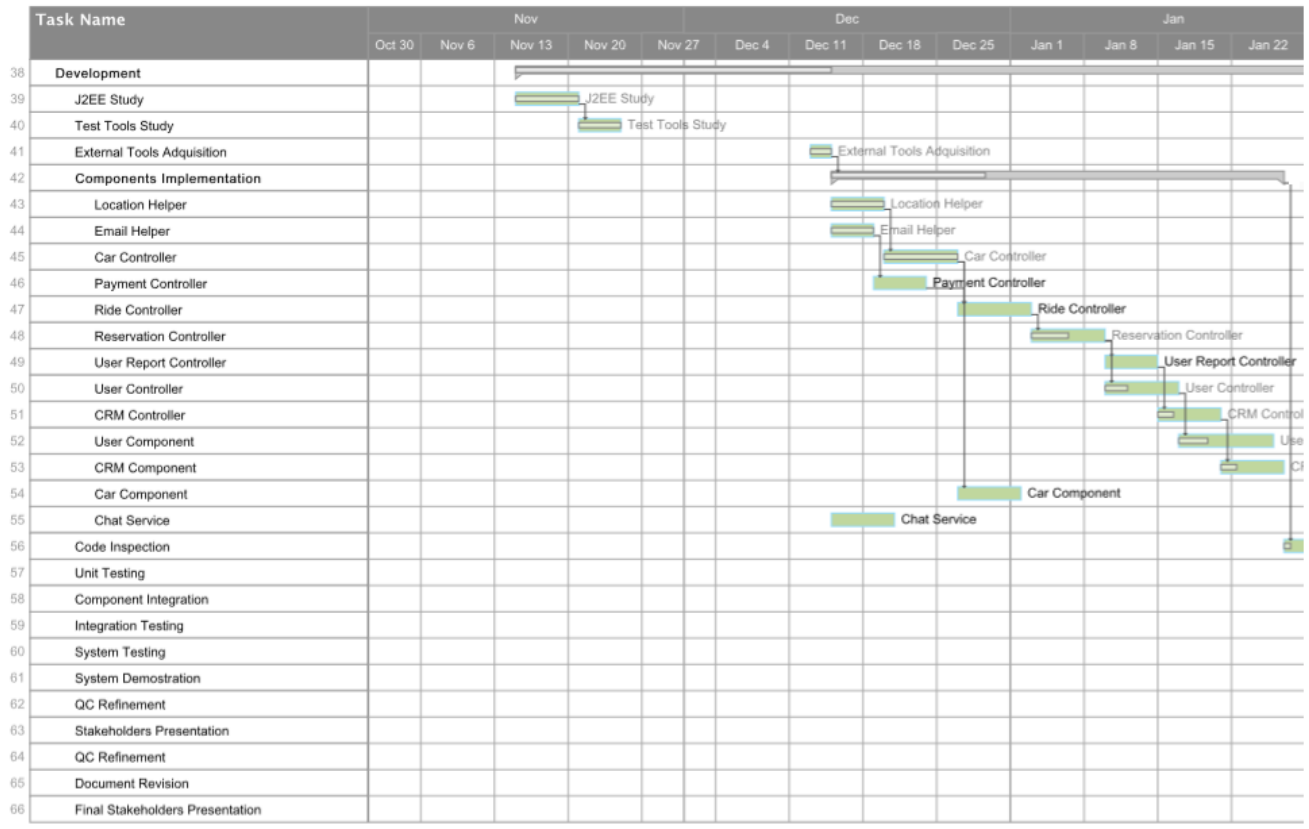
This section illustrate a high level view of the organization of the project schedule, showing the main phases of the production process. Should it be implemented, more detailed schedules for each of the phases and each of the sprints organized inside them will be defined during the project for better internal organization.
Due to the size of the Schedule Graph we separated in 4 different sections, RASD, DD, Development and Deployment.



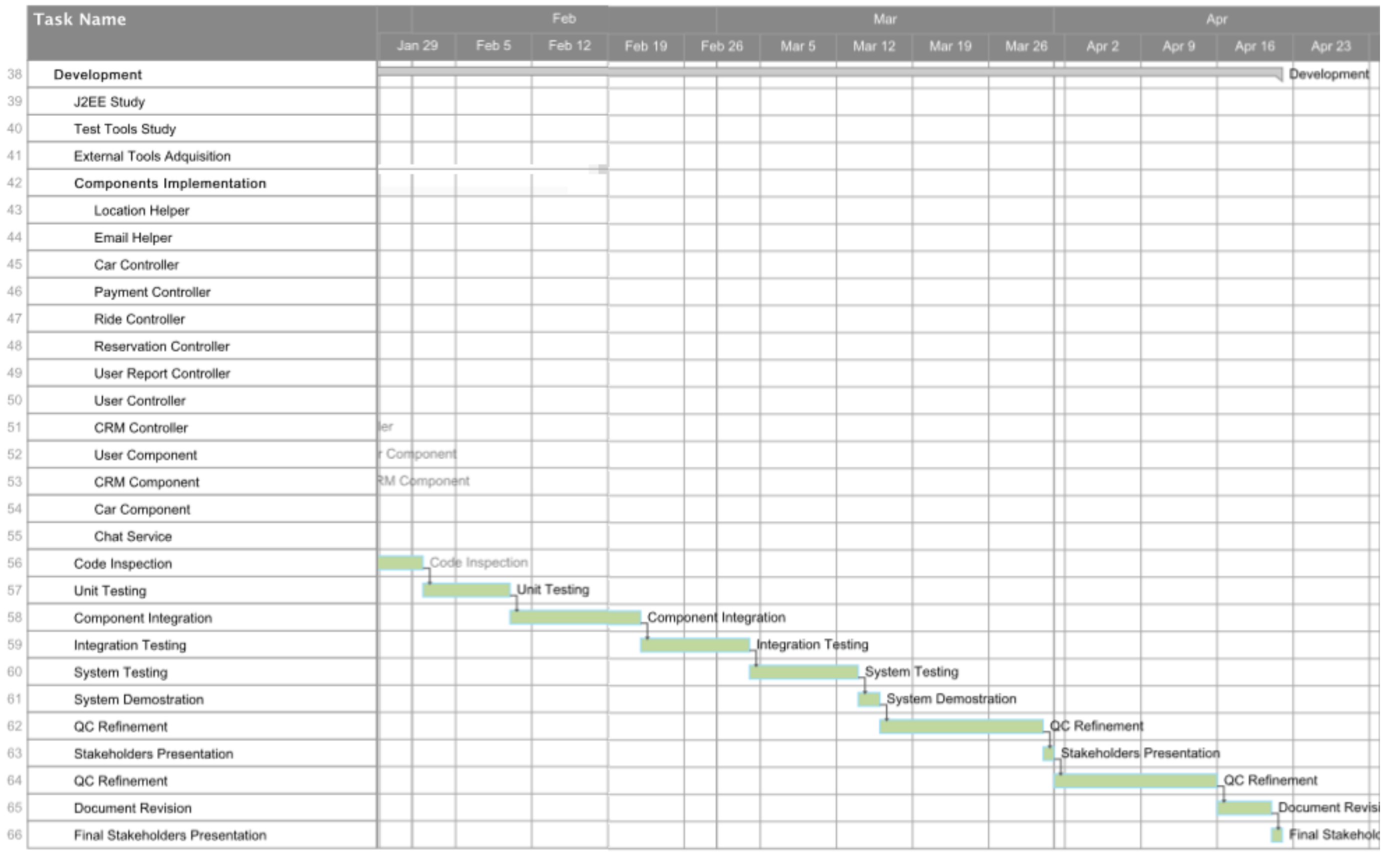
	Task Name
1	RASD
2	Meeting with Stakeholders
3	Identification of Requirements
4	Goals Definition
5	Scenarios Definition
6	Use Cases Definition
7	Meeting with Stakeholders
8	UML Diagrams Study
9	Sequence Diagrams
10	Class Definition
11	Initial UI Drafts
12	Function Definitions
13	Initial Size and Cost Estimation
14	Meeting with Stakeholders
15	Document Refinement
16	Document Approval by StakeHolders
17	Project Presentation, Legal Confirmation

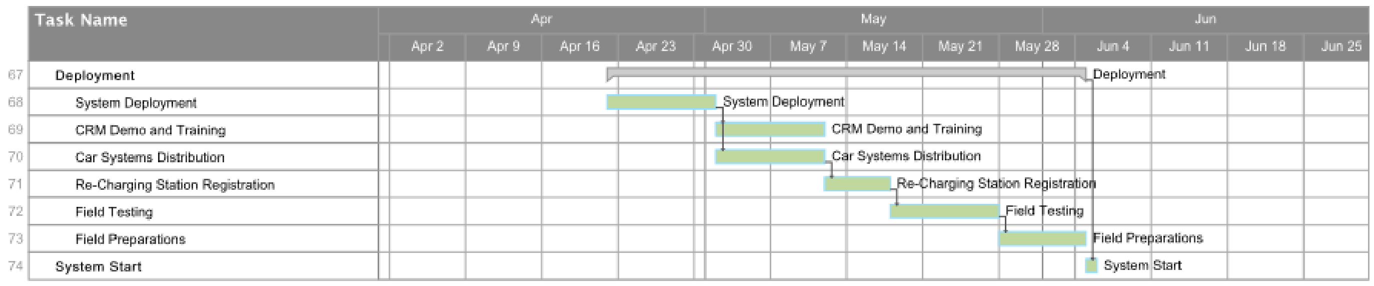


	Task Name
18	DD
19	Meeting with Stakeholders
20	High Level Component Architecture
21	Main Components Definition
22	External Tools Study
23	Component Overview
24	Deployment View
25	Runtime View
26	Interfaces Definition
27	Architecture Review
28	Meeting with Stakeholders
29	Refinement of Components
30	Subsystems Definition
31	Algorithms Design
32	UI Mockups Refinement
33	Meeting with Stakeholders
34	Requirements Traceability
35	Final Architecture Design
36	Final Refinements
37	Design Approval by StakeHolders



	Task Name
38	Development
39	J2EE Study
40	Test Tools Study
41	External Tools Adquisition
42	Components Implementation
43	Location Helper
44	Email Helper
45	Car Controller
46	Payment Controller
47	Ride Controller
48	Reservation Controller
49	User Report Controller
50	User Controller
51	CRM Controller
52	User Component
53	CRM Component
54	Car Component
55	Chat Service
56	Code Inspection
57	Unit Testing
58	Component Integration
59	Integration Testing
60	System Testing
61	System Demostration
62	QC Refinement
63	Stakeholders Presentation
64	QC Refinement
65	Document Revision
66	Final Stakeholders Presentation


Task Name
Feb
Mar
Apr
Jan 29
Feb 5
Feb 12
Feb 19
Feb 26
Mar 5
Mar 12
Mar 19
Mar 26
Apr 2
Apr 9
Apr 16
Apr 23
38 Development
39 J2EE Study
40 Test Tools Study
41 External Tools Adquisition
42 Components Implementation
43 Location Helper
44 Email Helper
45 Car Controller
46 Payment Controller
47 Ride Controller
48 Reservation Controller
49 User Report Controller
50 User Controller
51 CRM Controller
52 User Component
53 CRM Component
54 Car Component
55 Chat Service
56 Code Inspection
57 Unit Testing
58 Component Integration
59 Integration Testing
60 System Testing
61 System Demostration
62 QC Refinement
63 Stakeholders Presentation
64 QC Refinement
65 Document Revision
66 Final Stakeholders Presentation


Task Name
Apr
May
Jun
Apr 2
Apr 9
Apr 16
Apr 23
Apr 30
May 7
May 14
May 21
May 28
Jun 4
Jun 11
Jun 18
Jun 25
67 Deployment
68 System Deployment
69 CRM Demo and Training
70 Car Systems Distribution
71 Re-Charging Station Registration
72 Field Testing
73 Field Preparations
74 System Start

4 Resource Allocation

5 Risk Management

This section aims at defining at an early stage what are the risks regarding this project, and by doing so, define what are the indicators of a risk materializing itself and the actions that should be taken. The risk management is a continuous process. Therefore, the following are guidelines that should be upgraded as the project progresses.
A common risks computer softwares face are that the delivered product is not in total accordance with the vision the client had, and thus does not fully attend the functionalities proposed. Or it may happen that the users and CRM staff are not pleased with the final version, resulting in some resistance to use the product and bad public acceptance. This type of risks are managed by involving both the client and the users into early stages of development. The client should participate in the requirement analysis, design, and be presented with constant reports and demonstrations of new functionalities, while the users need to be brought to test functionalities and interfaces every time it is possible, especially before they are final versions. The same is the case for the CRM team, apart from bringing them closer to the development, it is important to also provide training on how to use the final system, to ensure a low resistance to the software and increase the quality of the deployed service.
A predictable risk is a turnover in the client situation, demanding the decrease of resources or the project to be completed in less time. Such scenario may arise for different reason, as example, a competitor company announces that will release a similar service before PowerEnjoy or a financial crisis is causing the client to invest less money than expected. Those are completely different problems, but the overall guidelines are the same: promote a negotiation with the client, explaining the consequences of this action. If, however, it is really necessary to do such change, establish a plan of incremental deployments, being the first one on the date set or using the money provided with limited functionalities, and make incremental deliveries as time or money become available.
There are risks that represent greater problems as they cannot have clear contingency plans beforehand, such are the cases of legislation changes regarding the car sharing service in Italy. New laws concerning the parking spots and other traffic rules can cause significantly changes in the product. And, although this scenario is hard to predict, such changes donâĂŹt come unnoticed, and by maintaining a good relationship with the city hall is possible to be aware with enough antecedence to adapt the service.
The deployed system, inevitably will depend on external solutions, as the

cloud infrastructure and the mapping service. For the first, it is necessary to do a complete market research before choosing the company that will provide the service, some key points are: a clear contract, defining what are the changes that can happen in the cloud infrastructure and binding them to send a notice before doing so; the structure used should be as generic as possible, predicting that in case of having to migrate, the damage is minimized; the line between price and quality here should be pending to the quality side. As for the second, the used solution is well-known and widely accepted, Google maps. For this reason, instability or changes in the service are not a high concern, nonetheless, in case it does not meet the company needs anymore, a replacement should already be kept in background, with some preliminary tests on migration.
Also the database, payment and e-mail are factor of risks, they have a serious effects but low probability and are easily fixed due to the fact they are relatively standardized. Backup routines should be established and followed rigorously, with redundant copies in different medias. Still regarding data loss, the source code needs to be included in the backups and the developers should be constantly reminded to keep the quality of the GIT repository, following the specifications for the commits.
Some internal problems may be summarized as the unavailability of a staff member. To ensure that no functionality is left undone, no member is single handed responsible for a specific part, or in other words, the work distributions overlaps themselves. But, as this is a small team, it may result in overload of work for the remaining members. This may be a difficult point to solve, as the period of absence in unsure, one of the management technique adopted that helps this and almost most issues in the development phase is to include extra time in the ending of major tasks.
A final predicted risk, is the failure of the screen inside the car, as it is a hardware piece it is subjected multiple malfunctions. As for the quality of the product, extensive testing with different hardwares was predicted in the testing phase, so, in case of the hardware proves to be insufficient to the needs of the system, there are other tested options. And for punctual failures, the company is advised to keep spare screens in stock for a fast change. Of course, the later a problem is detected here, the more expensive is to fix it. As an example, if on development it is discovered that the first option for the screen breaks easily after little use, there are only a few to be replaced, but if the system was already deployed, this may result in huge losses for the company.
The above consideration (the later a risk presents itself the more expensive it is to fix it) tends to be true for most cases, hence it is why is crucial to keep reviewing the contingency plans and monitoring the signals that a risk, predicted or non predicted, may arise.

6 Effort Spent

Date	**Domenico**	**Caio**	**Matheus**
15/01/17	2h	2h	2h
16/01/17	-	-	-
17/01/17	4h	-	-
18/01/17	4h	-	-
19/01/17	-	6h	-
20/01/17	-	6h	1h
21/01/17	-	-	6h
22/01/17	-	-	-

7 Changelog

As the project and design decisions may change during the development this document is also prone to change. We'll document every version in this part.

- **Version 1.1:** 22/01/2017