POLITECNICO
MILANO 1863

Politecnico di Milano

COMPUTER SCIENCE AND ENGINEERING

Software Engineering 2

Project Plan

PowerEnJoy

Authors:
Francesco Fabiani
Jagadesh Manivannan
Niccolò Pozzolini

Professors:
Elisabetta Di Nitto
Luca Mottola

Academic Year 2016/2017

Contents

Chapter 1

Introduction

1.1 Revision history

Version	Date	Authors	Summary
1.0	22/01/2017	Fabiani, Manivannan, Pozzolini	Initial release

Table 1.1: Changelog of this document

1.2 Purpose and scope

The Project Plan (PP) document is intended to describe the best strategies for the management of PowerEnJoy with regards to all the aspects of the project, such as costs, schedule of the activities, resource allocation and effort estimation.

The product described is PowerEnJoy, a car-sharing service which offers to its users exclusively electric cars. It includes the common functionalities of its category: permitting to registered users to obtain the position of all the available cars, reserving one within a certain amount of time and continuously displaying the up-to-the-minute cost of the ride are just few of them. Moreover, PowerEnJoy stimulates users to behave virtuously towards the ecosystem by applying various types of discounts under specific conditions.

1.3 Definitions, acronyms, abbreviations

- *ACAP*: Analyst Capability

- *APEX*: Applications Experience
- *API*: Application Programming Interface
- *BCE*: Business Controller Entity
- *Car*: electric vehicle provided by the service
- *CPLX*: Product Complexity
- *DAG*: Directed Acyclic Graph
- *DB*: Database
- *DBMS*: Database Management System
- *DD*: Design Document
- *DOCU*: Documentation Match to Life-Cycle Needs
- *ER*: Entity-Relationship
- *GPS*: Global Positioning System
- *Guest* or *Guest user*: person not registered to the service
- *ITPD*: Integration Test Plan Document
- *LTEX*: Language and Tool Experience
- *MVC*: Model View Controller
- *OS*: Operating System, related both to desktop and mobile platforms
- *PCAP*: Programmer Capability
- *PCON*: Personnel Continuity
- *PIN*: Personal Identification Number
- *PLEX*: Platform Experience
- *PP*: Project Plan
- *PVOL*: Platform Volatility
- *RASD*: Requirements Analysis and Specification Document
- *Registered user*: see *User*

- *RELY*: Required Software Reliability
- *REST*: Representational State Transfer
- *RESTful*: that follows the REST principles
- *RUSE*: Developed for Reusability
- *Safe area*: set of parking spots where a user can leave a car without penalization
- *STOR*: Main Storage Constraint
- *User*: person with a valid driving license registered to the service
- *UX*: User eXperience
- *W3C*: World Wide Web Consortium

1.4 Reference documents

The PP document has been composed following the guidelines reported in the Requirements Analysis and Specification Document delivered for this project. Moreover, the part describing the cost estimation follows the indications described in the second revision of the procedural software cost estimation model named Constructive Cost Model (COCOMO II), developed by Barry W. Boehm.

With regards to the course named Software Engineering 2 and held by professors Luca Mottola and Elisabetta Di Nitto (Politecnico di Milano, a. y. 2016/17), the document conforms to the guidelines provided during the lectures and within the material of the course.

Chapter 2

Project size, cost and effort estimation

2.1 Size estimation

2.2 Cost and effort estimation

In this section we are going to use the COCOMO II approach to estimate the cost and effort needed to develop the PowerEnJoy application.

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	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	5.0	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	Level 1 lower	Level 1 upper	Level 2	Level 3	Level 4	Level 5
SF_j	7.80	6.24	4.68	3.12	1.56	0.00

Table 2.1: Scale Factor values (SF_j) for COCOMO II Models

A brief description for each scale driver:

- *Precedentedness*: this factor determines or reveals the level of exposure or experience in development of large scale projects or similar kind of projects that out team has done before. Since we have developed few projects like this, we can set this value to be Nominal.
- *Development flexibility*: it determines the degree of flexibility in the development process with respect to the external specification and requirements. In our project, the functionalities and requirements are clear and well defined with no specific mention about the technology. Hence this value would be low.
- *Architecture/Risk resolution*: it determines the level of awareness and reactivity with respect to risks. Since we have an extremely good risk management plan, we consider this value to be very high.
- *Team cohesion*: it determines if all the Stakeholders are able to work in a team and share same vision and commitment. Since our team is highly co-operative, the value is very high.
- *Process maturity*: we have a done an extremely fair work to meet our goals successfully in this project. Since we had prior experience in successfully dealing these kind of projects, the value is set to Level 4.

The results of our evaluation is the following:

Scale Driver	Factor	Value
Precedentedness (PREC)	Nominal	3.72
Development flexibility (FLEX)	Low	4.05
Risk resolution (RESL)	Very high	1.41
Team cohesion (TEAM)	Very high	1.10
Process maturity (PMAT)	Level 4	1.56
Total		11.84

2.2.2 Cost drivers

Product factors

- *Required Software Reliability (RELY)*:

  The software application is developed in such a way that the main aim is to reserve and take a ride in the Cars in the city. Any malfunctioning could lead to important financial loss. Considering this, the RELY cost driver is set to high.

RELY cost drivers						
RELY descriptors	slightly inconvenience	easily recoverable losses	moderate recoverable losses	high financial loss	risk to human life	
Rating level	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)*:

  This factor considers the effective size of our database. We do'nt know this value exactly. But based on the lower and upper bound values of the SLOC, which is 10.000-15.000 SLOC, we can estimate roughly that our system can reach a 3GB database size. Since it is distributed over 10.000-15.000 SLOC, the ratio D/P (measured as testing DB bytes/program SLOC) is between 209 and 314, resulting in the DATA cost driver being high.

DATA cost drivers						
DATA descriptors		$\frac{D}{P} < 10$	$10 \leq \frac{D}{P} < 100$	$100 \leq \frac{D}{P} < 1000$	$\frac{D}{P} \geq 1000$	
Rating level	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)*:

  This factor is related to the complex logics involved in implementing the product as a whole. Hence, we set it to very high according to the CPLX cost driver table.

CPLX cost drivers						
Rating level	Very low	Low	Nominal	High	Very high	Extra high
Effort multipliers	0.73	0.87	1.00	1.17	1.34	1.74

- *Developed for Reusability (RUSE)*:

  In our project, we use many individual piece of codes that can be made reusable for other services or functions. Hence the RUSE cost driver is set to nominal.

RUSE cost drivers						
RUSE descriptors		None	Across project	Across program	Across product line	Across multiple product
Rating level	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)*:

This factor describes the relationship between the documentation and the application requirements. The product life-cycle needs are explicitly mentioned clearly in the documentation. Hence the DOCU cost driver is set to nominal.

DOCU cost drivers						
DOCU descriptors	Many life-cycle needs uncovered	Some life-cycle needs uncovered	Right sized to life-cycle needs	Excessive for life-cycle needs	Very excessive for life-cycle needs	
Rating level	Very low	Low	Nominal	High	Very high	Extra high
Effort multipliers	0.81	0.91	1.00	1.11	1.23	n/a

Platform factors

- *Execution Time Constraint (TIME)*:

  This factor describes the approximated value of CPU usage with respect to the hardware specifications. Our PowerEnJoy application has vast functionalities as a software and hence the TIME cost driver is set to be very high.

TIME cost drivers						
TIME descriptors			$\leq$ 50% use of available execution time	70% use of available execution time	85% use of available execution time	90% use of available execution time
Rating level	Very low	Low	Nominal	High	Very high	Extra high
Effort multipliers	n/a	n/a	1.00	1.11	1.29	1.63

- *Main Storage Constraint (STOR)*:

  This factor describes the approximated storage space with respect to the hardware specifications. Our PowerEnJoy application has

vast functionalities as a software. Keeping this in mind, the disk drives can store up to enough terabytes and hence the STOR cost driver is set to be high.

STOR cost drivers						
STOR descriptors			$\leq$ 50% use of available storage	70% use of available storage	85% use of available storage	90% use of available storage
Rating level	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)*:

  This factor describes the change in the basic or fundamental platform in which the system is designed. We do'nt change the platform often except for very few major releases or updated requested by the client. This will be done approximately for every 5months to be in sync with the latest evolving or trending technologies. Hence, the PVOL cost driver is set to nominal.

PVOL cost drivers						
PVOL descriptors		Major change every 12 months; minor change every 1 month	Major change every 6 months; minor change every 2 weeks	Major change every 2 months; minor change every 1 week	Major change every 2 weeks; minor change every 2 days	
Rating level	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)*:

This factor describes the potential analysis that has been done with respect to the potential implementation in real world. Since we have done a regressive analysis, the ACAP cost driver is set to be high.

ACAP cost drivers						
ACAP descriptors	15th percentile	35th percentile	55th percentile	75th percentile	90th percentile	
Rating level	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)*:

  This factor describes the ability of the programmer to do a work without much difficulty. Our project has not been implemented yet our programmers have executed several projects like this successfully and hence the PCAP cost driver is set to be high.

PCAP cost drivers						
PCAP descriptors	15th percentile	35th percentile	55th percentile	75th percentile	90th percentile	
Rating level	Very low	Low	Nominal	High	Very high	Extra high
Effort multipliers	1.34	1.15	1.00	0.88	0.76	n/a

- *Applications Experience (APEX)*:

  Our team members are quite experienced with this kind of project development and hence the APEX cost driver is set to be high.

APEX cost drivers						
APEX descriptors	≤ 2 months	6 months	1 years	3 years	6 years	
Rating level	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)*:

  Our team has a good and stable experience in Java EE platform and also a good knowledge about the integration with UI, Database and other tiers. Hence the PLEX cost driver is set to be high.

PLEX cost drivers						
PLEX descriptors	≤ 2 months	6 months	1 years	3 years	6 years	
Rating level	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)*:

  As we have mentioned before, since the knowledge of our programmers are good enough on this kind of project and Java EE platform, they possess a good standard of using tools in the development environment, server side and client side integration,etc. Hence the LTEX cost driver is set to be high.

LTEX cost drivers						
LTEX descriptors	≤ 2 months	6 months	1 years	3 years	6 years	
Rating level	Very low	Low	Nominal	High	Very high	Extra high
Effort multipliers	1.20	1.09	1.00	0.91	0.84	n/a

- *Personnel Continuity (PCON)*:

  This factor describes the personnel turnover annually. Since our project is a short term project, the PCON cost driver is set to be very low.

PCON cost drivers						
PCON descriptors	48%/year	24%/year	12%/year	6%/year	3%/year	
Rating level	Very low	Low	Nominal	High	Very high	Extra high
Effort multipliers	1.29	1.12	1.00	0.90	0.81	n/a

Project factors

- *Use of Software Tools (TOOL)*:

  Our application environment is complete and well integrated, so we will set this parameter as high.

TOOL cost drivers						
TOOL descriptors	edit, code, debug	simple, front-end, back-end CASE, little integration	basic life-cycle tools, moderately integrated	strong, mature life-cycle tools, moderately integrated	strong, mature, proactive life-cycle tools, well integrated with processes, methods, reuse	
Rating level	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)*:

Our application is designed in such a way that it relies on wide-band electronic communication at extremely good speeds (e.g. 3G, 4G) for connection. Hence the SITE cost driver is set to be very high.

SITE cost drivers						
SITE collocation descriptors	international	multi-city and multi-company	multi-city or multi-company	same city or metro area	same building or complex	fully collocated
SITE communications descriptors	some phone, mail	individual phone, FAX	narrow band email	wideband electronic communication	wideband electronic communication, occasional video conference	interactive multimedia
Rating level	Very low	Low	Nominal	High	Very high	Extra high
Effort multipliers	1.22	1.09	1.00	0.93	0.86	0.80

General factor

- *Required Development Schedule (SCED)*:

  The efforts was distributed or split equally in our project for all the documentation, yet there were certain time consuming process in analysing and development of the RASD and the DD documents for precision. Hence, the SCED cost driver is set to be high.

SCED cost drivers						
SCED descriptors	75% of nominal	85% of nominal	100% of nominal	130% of nominal	160% of nominal	
Rating level	Very low	Low	Nominal	High	Very high	Extra high
Effort multipliers	1.43	1.14	1.00	1.00	1.00	n/a

Results

Overall our results are expressed in the following table:

Cost Driver	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
Main storage constraint (STOR)	High	1.11
Platform volatility (PVOL)	Nominal	1.00
Analyst capability (ACAP)	High	0.85
Programmer capability (PCAP)	High	0.88
Application experience (APEX)	High	0.88
Platform Experience (PLEX)	High	0.91
Language and Tool Experience (LTEX)	High	0.91
Personnel continuity (PCON)	Very low	1.12
Usage of Software Tools (TOOL)	High	0.90
Multisite development (SITE)	Very high	0.86
Required development schedule (SCED)	High	1.00
Total		1.13694

2.2.3 Effort equation

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

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

where:

$A = 2.94$ (for COCOMO II)

$EAF = 1.13694$ (product of all cost drivers)

$E = B + 0.01 \times \sum_{i} SF[i] = B + 0.01 * 11.84 = 0.91 + 0.1464 = 1.0284$

(exponent derived from the scale drivers, with B = 0.91 for COCOMO II)

With this parameters we can compute the effort value, which has a lower bound of:

$$\begin{aligned} Effort = A \times EAF \times KSLOC^E &= 2.94 \times 1.13694 \times dash^{1.0284} \\ &= 4 \text{ PM} \approx dash \text{ PM} \end{aligned}$$

and an upper bound of:

$$\begin{aligned} Effort = A \times EAF \times KSLOC^E &= 2.94 \times 1.13694 \times dash^{1.0284} \\ &= 4 \text{ PM} \approx dash \text{ PM} \end{aligned}$$

2.2.4 Schedule estimation

Regarding the final schedule, we are going to use the following formula:

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

As a lower bound, we consider

$$\begin{aligned} &F = 0.28 + 0.2 \times (E - B) = 0.28 + 0.2 \times 0.1464 = 0.30928 \\ &Effort = 56.957 \text{ PM} \\ &Duration = 3.67 \times 56.957^{0.30928} = 12.81 \text{ months} \end{aligned}$$

while as an upper bound, we consider

$$\begin{aligned} &F = 0.28 + 0.2 \times (E - B) = 0.28 + 0.2 \times 0.1464 = 0.30928 \\ &Effort = 84.737 \text{ PM} \\ &Duration = 3.67 \times 84.737^{0.30928} = 14.49 \text{ months} \end{aligned}$$

Chapter 3

Schedule

The main tasks involving this project are:

- Delivering the Requirement Analysis and Specification Document, containing the goals, the domain assumptions, and the functional and non-functional requirements of the software system.
- Delivering the Design Document, containing the architecture and the design of the software system.
- Delivering the Integration Testing Plan Document, containing the strategy used to perform integration testing on the system.
- Delivering the Project Plan, which is this document.
- Preparing a brief presentation about the delivered documents, with slides.
- Implementing the software system and write unit tests.
- Performing integration testing on the system.

Please note that, as new requirements can emerge, new choices are made and the development goes on, the process can be iterated multiple times. In particular, unit and integration testing will be continuously performed throughout the development process.

However, some tasks need to be concluded before some other can begin: the dependency graph for the activities is shown in figure 3.1.

The first five tasks for the project are already defined by the document about describing the assignment, together with the deadlines for the delivery of the RASD, the Design Document and the ITPD. The date for the presentation is also fixed. So, those activities are already scheduled.

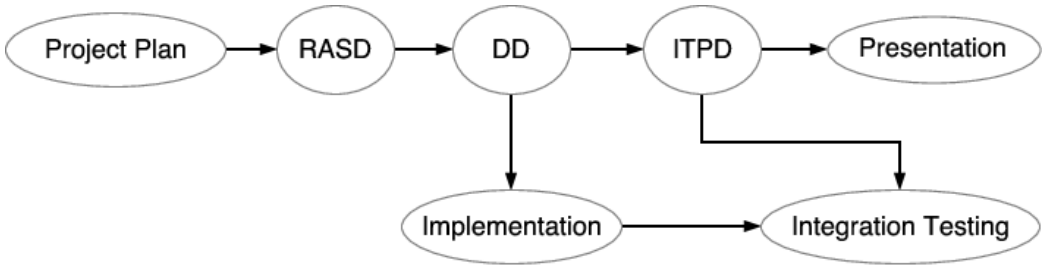

Figure 3.1: DAG for the dependencies among tasks

There are no fixed deadlines, instead, for the development of the software. Based on the COCOMO estimation performed in chapter 2, we expect the entire project to last 8 months, so it will be presumably finished by June 2017. The schedule for our project is outlined in table 3.1, while figure 3.2 shows the Gantt chart for PowerEnJoy.

Activity	**Start date**	**Deadline**
RASD	16/10/2016	13/11/2016
DD	14/11/2016	11/12/2016
IPTD	12/12/2016	15/01/2017
Project Plan	05/01/2017	22/01/2017
Presentation		
Implementation		
Integration testing		

Table 3.1: Schedule for the project tasks

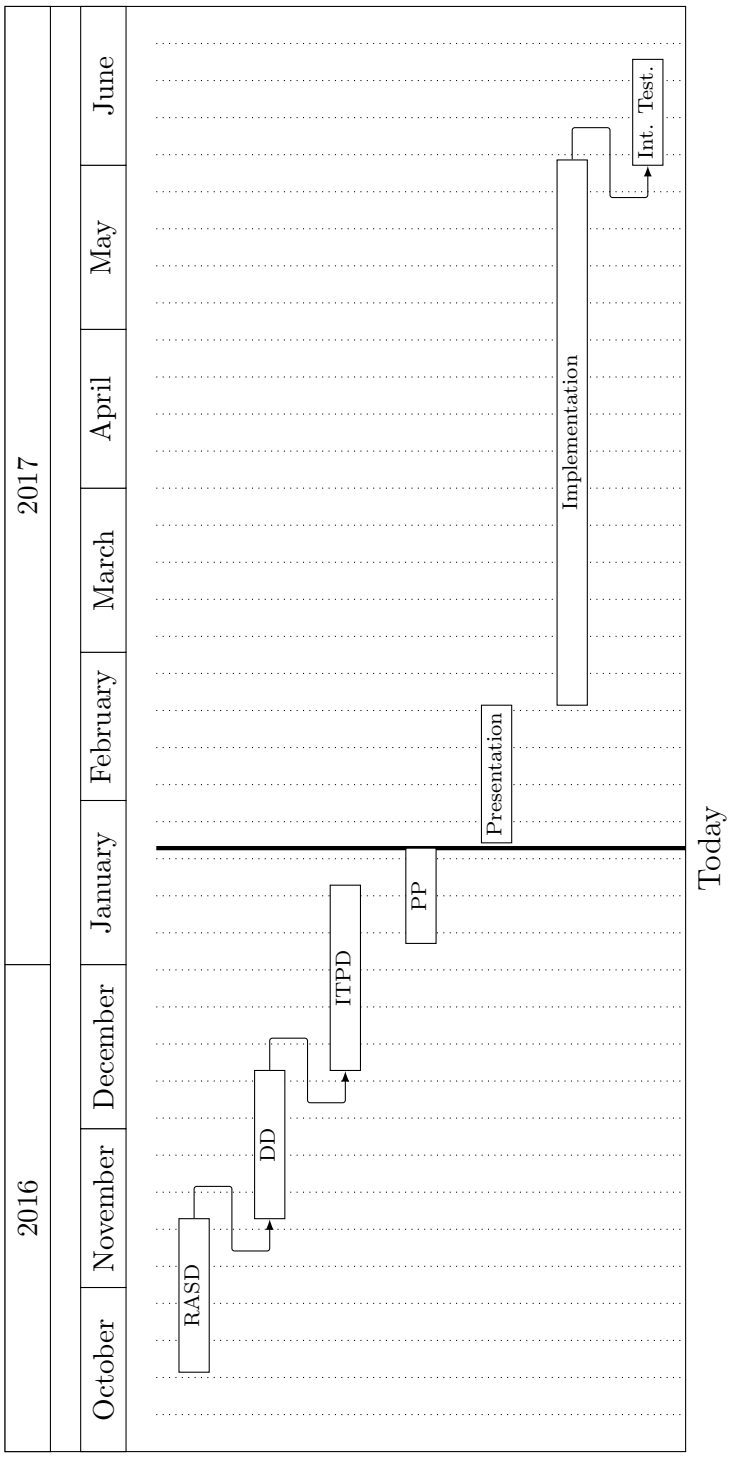

Figure 3.2: Gantt chart of the project