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
- *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