Politecnico di Milano

A.A. 2016–2017

Software Engineering 2: "PowerEnJoy"

Project Plan

Pietro Ferretti, Nicole Gervasoni, Danilo Labanca

January 21, 2017

Contents

1 Introduction

1.1 Purpose

The purpose of this document is to provide a detailed analysis of the PowerEnjoy software development project in terms of required cost and time. It highlights the estimation of

- project size, calculated using the *Function Points approach* by IBM;

- project cost and effort, calculated using the *COCOMO II* by Boehm.

Given the previous information we elaborate a feasible schedule considering all the necessary activities in detail, thus the best resources' allocation on each one. The last section of the document focuses on handling all the possible risks that could be met during the whole process, from the requirements analysis to the final testing and deployment.

1.2 Scope

The aim of this project is to specify and design a new digital management software for PowerEnJoy, a car-sharing service that employs electric cars only.

PowerEnJoy will offer a very valuable service to its users, letting them borrow cars to drive around the city freely, as an alternative to their own vehicles and public transport. Among the advantages of using PowerEnJoy we can note being able to find available cars in any place that is served by our system and having dedicated spots to park in (namely, PowerEnJoy's power grid stations). Furthermore, thanks to the fact that all the cars that we provide are electrically powered, PowerEnJoy is also very environmentally friendly.

1.3 List of Definitions and Abbreviations

1.3.1 Definitions

1.3.2 Acronyms

- **ITPD**: Integration Test Plan Document

- **DD**: Design Document

- **RASD**: Requirements Analysis and Specification Document

- **DB**: Database

- **PGS**: Power Grid Station

- **GPS**: Global Positioning System

- **API**: Application Programming Interface

- **ISDTN**: International Standard Date and Time Notation

- **EM**: Effort Multiplier

- **FP**: Function Points

- **ILF**: Internal Logic File

- **ELF**: External Logic File

- **EI**: External Input

- **EO**: External Output

- **EQ**: External Inquiries

- **UI**: User Interface

1.4 List of Reference Documents

- Requirements analysis and specification document: "RASD.pdf"

- Design document: "DD.pdf"

- Integration testing document: "ITPD.pdf"

- Project description document: "Assignments AA 2016-2017.pdf"

- Example document: "Project planning example document.pdf"

- "COCOMO II – Model Definition Manual", version 2.1, 1995-2000, Center for Software Engineering, USC

2 Project size, cost and effort estimation

Pay a Bill	
Input	*Result*
A valid session token, a bill that needs to be paid and a valid payment method	The transaction is carried out; if it succeeds the bill is marked as paid, otherwise returns failure.
A valid session token, a bill that needs to be paid and an ill-formed payment method	An exception is raised.
A valid session token and a bill that needs to be paid	The system uses the payment method saved for the user to carry out the transaction; if it succeeds the bill is marked as paid, otherwise returns failure.
A valid session token and a bill that has already been paid	An exception is raised.
A valid session token and a non-existent bill	An exception is raised.
An invalid session token and a bill	An exception is raised (bad authentication).

ELF	Complexity	FPs
elf n1	Low	5
elf n2	High	10
elf n3	medium	7
Total		**22**

Table 1: asdfasdf

ELF	Complexity	FPs
elf n1	Low	5
elf n2	High	10
elf n3	medium	7

| | *total* | **22** |

Table 2: ewerewr

Cost Driver	Rating Level	EM
Documentation match to life-cycle needs (DOCU)	Nominal	1.00
Total		**1.00**

Table 3: I'm a table.

2.1 Size estimation: function points

Function points are useful in expressing the amount of business functionality our software has to provide to a user and are used to compute an estimation of its size. After been identified and categorized into one of five types: outputs, inquiries, inputs, internal files, and external interfaces, each functional requirement is then assessed for complexity and assigned a number of function points. We based our computation on tables and values in *COCOMO II Model Definition Manual v. 2.1.*

2.1.1 Internal Logic Files (ILFs)

They are all kinds of data used and managed by the application in order to offer the expected functions.
Data will be organize in the following tables in the DB:

- **user** : name, surname, username, password, dob, email, licenseID, cvv, cardNumber, accountStatus

- **bill** : associatedLicense, total, date, rideID, carID, paymentStatus

- **car** : model, plate, ID, available, issues

- **report** :carID, description, associatedLicense, date

- **safeArea** : latitude, longitude, ID

- **PGS** : latitude, longitude, ID

- **plug** : available, ID

- **reservation** : ID, associatedLicense, carID, date, status

- **ride** : ID, associatedLicense, associatedBill, date, status, ridingTime, carID

The software will operate directly on the previously listed data and with the tables generated from their relations between each other.
All this data are modeled in simple structures so they're complexity can be considered low (referring to tables).

$$FPs(ILF) = 7 * 9 + 10 = 73$$

2.1.2 External Logic Files (ELFs)

- mappe trovare posizione da indirizzo <- Low trovare distanza in minuti tra due punti (isocrone) <- low trovare strade? - patenti (richieste api) chiedere se patente è valida <- meno che low

3x5 = 15 Fps frame per second per santa claus ELF Executable linkable format

2.1.3 External Inputs (EIs)

lista di funzionalità con input utente

- login -> low - update user (tutti i tipi) -> medium - pay bill -> high - create reservation -> medium - cancel reservation -> low - start ride -> low - end ride? -> high - park -> medium - unlock -> medium - update macchina -> low - update plugs -> low - set car unavailable -> low - set car available -> low - report issue -> medium

low x 7 = 3x7 = 21 medium x 4 = 4x4 = 16 high x 2 = 6x2 = 12

tot 49 FPs per EI ti va una schweppes solo io e te?

2.1.4 External Inquiries (EQs)

- get info utente -> low - get bills -> low - car search con position -> medium - car search con address -> medium - pgs search con position -> medium - pgs search con address -> medium - money saving option -> high - safe area search pos -> medium - " add -> medium - cars in need of maintenance -> low

lowx3 = 3x3 = 9 medium x6 = 4x6 = 24 high x1 = 6x1 = 6

39 per EQs for house music

2.1.5 External Outputs (EOs)

- lock car -> low - unlock car -> low - richiedi update dalla macchina -> medium

4x2+5x1=13 FPs per EOOOOOOOOO

2.1.6 Overall estimation

ILF 73 ELF 15 EI 49 EQ 39 EO 13

Total 189

con AVC 46 per Java EE SLOC = 189 * 46 = 8694

2.2 Cost and effort estimation: COCOMO II

2.2.1 Scale Drivers

qua ci vuole la tabella dove ci sono tutti i valori

Precedentedness -> very low 6,20 flexibility -> high 2.03 o very high 1.01 risk -> very high 1,41 team -> very high 1,10 maturity -> level 3 3.12 (non so perchè)

E=B+0.01xSUM SF,whereB=0.91

E = 0.91+0.01x(6.20+1.01+1.41+1.1+3.12)-> 12,84 = 1,0384

2.2.2 Cost Drivers

RELY -> nominal 1.0 (se l'applicazione non funziona perdiamo soldi e magari qualcuno ci fa causa, ma non succede il fini mondo) DATA -> high 1.14(come l'esempio, non saprei dare altre motivazioni)

2.2.3 Effort equation

2.2.4 Schedule estimation

3 Schedule

4 Resource allocation

5 Risk Management

A Changelog

B Hours of work