POLITECNICO
MILANO 1863

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

- *User*: a person that is registered to the system. Users can log in to the system with their email or username and their password. Their first name, last name, date of birth, driving license ID are stored in the database.
- *Safe area*: a location where the user can park and leave the car. Users can end their ride and park temporarily only in these locations. The set of safe areas is predefined by the system.
- *Power grid station* or *Charging station*: a place where cars can be parked and plugged in. While a car is plugged in a power grid station its battery will be recharged. Power grid stations are by definition safe areas.
- *Available car*: a car that is currently not being used by any user, and has not been reserved either. Available cars are in good conditions (not dirty nor damaged) and don't have dead batteries.
- *Reservation*:
  - the operation of making a car reserved for a user, i.e. giving permission to unlock and use the car only for that user, forbidding reservations by other users.
  - the time period between the moment a reservation is requested and the moment the user unlocks the car, or the reservation is canceled.
- *Ride*: the time period from the moment a reserved car is unlocked to the moment the user notifies that he wants to stop using the car and closes all the doors. A ride doesn't stop when a car is temporarily parked, but continues until the user chooses to leave the car definitely.
- *Temporary parking*: the act of parking a car in a safe area and, after notifying the system, locking it and leaving it for a finite amount of time. The user that does this retains the right to use the car and can unlock it later to use it again.
- *Bill*: a record of the money owed by the user at the end of a ride.
- *Suspended user*: a user that cannot reserve or use cars. Usually users are suspended because they have outstanding bills that have not been paid.
- *Payment method*: a way to transfer money from the user to the system. Our system will only accept credit cards and online accounts like Paypal.

1.3.2 Acronyms

- **ITPD**: Integration Test Plan Document
- **DD**: Design Document
- **RASD**: Requirements Analysis and Specification Document
- **DB**: Database
- **CVV**: Card Verification Value
- **DOB**: Date of birth
- **PGS**: Power Grid Station
- **GPS**: Global Positioning System
- **API**: Application Programming Interface
- **ISDTN**: International Standard Date and Time Notation

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"

2 Project size, cost and effort estimation

2.1 Size estimation: function points

2.1.1 Internal Logic Files (ILFs)

quali internal logic files abbiamo? quali entità di dati abbiamo?

- users nome, cognome, username, password, dob, email, patente, carta di credito, cvv, stato (sospeso, non sospeso) - bills username, importo, data, id ride, car, stato (pagata/non pagata) - cars modello, targa, id, available, issues - reports id macchina, descrizione, utente, data - safe areas latitudine, longitudine, id - power grid stations latitudine, longitudine, id - plugs id plug, availability - reservations user, macchina, data, stato (attiva, annullata, completata) - rides user, macchina, data, durata, stato, bill

- mappe per navigatore in locale in modo complicato (average)

(- autenticazione id cosa, credenziali es. chiavi private)

users -> Low tutti low 7x9+10=73 FPs per ILF

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