



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	3
1.1	Purpose	3
1.2	Scope	3
1.3	List of Definitions and Abbreviations	4
1.3.1	Definitions	4
1.3.2	Acronyms	5
1.4	List of Reference Documents	5
2	Project size, cost and effort estimation	5
2.1	Size estimation: function points	6
2.1.1	Internal Logic Files (ILFs)	6
2.1.2	External Logic Files (ELFs)	6
2.1.3	External Inputs (EIs)	7
2.1.4	External Inquiries (EQs)	7
2.1.5	External Outputs (EOs)	7
2.1.6	Overall estimation	7
2.2	Cost and effort estimation: COCOMO II	7
2.2.1	Scale Drivers	7
2.2.2	Cost Drivers	7
2.2.3	Effort equation	8
2.2.4	Schedule estimation	8
3	Schedule	8
4	Resource allocation	8
5	Risk Management	8
A	Changelog	8
B	Hours of work	8

1 Introduction

1.1 Purpose

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

Pay a Bill	
<i>Input</i>	<i>Result</i>
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).

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 $7 \times 9 + 10 = 73$ 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

$3 \times 5 = 15$ 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

Forse è più corretto 53? vedi COCOMO II 2000

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 -> nominal 3,04 risk -> very
high 1,41 team -> very high 1,10 maturity -> level 3 3.12 (non so perchè)

rifare il conto, e metterlo sotto effort equation

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 finimondo) al massimo moderate, easily
recoverable losses

DATA -> dimensioni database: ordine di grandezza database di test 1GB
linee di codice: 9000 D/P = $10^9/10^4 = 1*10^5 = 100'000$ Very High -> 1.28

CPLEX -> - control operations: "Reentrant and recursive coding. Fixed-priority interrupt handling. Task synchronization, complex callbacks, heterogeneous distributed processing. Single- processor hard real-time control." Very High

- computational operations Nominal
- device-dependent operations Low
- data management operations High
- User Interface management operations nominal

$(5+3+2+4+3)/5=3$.qualcosa

totale Nominal 1.00

RUSE -> none, Low 0.95

DOCU -> right-sized to life cycle needs, Nominal 1.00

TIME -> un numero a caso High 1.11

STOR -> abbiamo un sacco di spazio, Nominal 1.00

PVOL -> major update 2 mesi, High 1.15

ACAP -> Nominal 1.00

PCAP -> capacissimi con java e che talento Very High 0.76

PCON -> no turnover, Very High 0.81

APEX -> poca esperienza, Low 1.10

PLEX -> veery low, Very Low 1.19

LTEX -> Nominal, 1.00

TOOL -> abbiamo eclipse, Very High 0.78

SITE -> same city or metro area, High 0.93

SCED -> boh, teniamo una schedule non accelerata, Nominal 1.00

2.2.3 Effort equation

$PM = A * (SLOC)^{\hat{E}} * \text{produttoria}(\text{cost drivers})$ $A = 2.94$

2.2.4 Schedule estimation

TDEV time to develop (in mesi) $TDEV = C * PM^{\hat{F}} F = (D + 0.2 * (E-B))$ $B = 0.91$ $C = 3.67$ $D = 0.28$ $E = \text{quello della effort equation}$

3 Schedule

4 Resource allocation

5 Risk Management

A Changelog

B Hours of work