Test Plan Document

Michele Madaschi Lidia Moioli Luca Martinazzi

January 31, 2016

Contents

Introd	\mathbf{uction}													2
1.1	Scope	and pur	pose .											2
Functi	on Poi	\mathbf{nts}												3
2.1	Comp	lexity ar	d cost e	valu	atic	n								3
	2.1.1	Interna	l Logic	File										3
	2.1.2	Extern	al Interf	ace	File									4
	2.1.3	Extern	al Input											4
	2.1.4	Extern	al Outpi	ut .										4
	2.1.5	Extern	al Inquii	ry .										4
	2.1.6	Overall												4
COCO	MO II	Ι												5
Schedu	ıle and	l resour	ces allo	cat	ion									6
4.1	Gant	t's diagr	am											6
Risk e	valuati	on												8
5.1	Risk e	valuatio	n and ax	zoid:	ance									8

Introduction

1.1 Scope and purpose

In this document we want to give an overall description of the project planning of $My\ Taxi\ Service$. We will:

- 1. identify deliverables and deadlines
- 2. extimate the total effort required
- 3. analize possible risks and contengency plans

Function Points

In order to evaluate the cost of the project we have to identify the function points and estimate the complexity of each one. To each point we assign a weight referring to this table:

Function types	Weight						
	Simple	Medium	Complex				
External Input	3	4	6				
External Output	4	5	7				
External Inquiry	3	4	6				
Internal Logic File	7	10	15				
External Interface File	5	7	10				

Internal Logic File: users (guest, taxidriver and passenger), ride, sharedride, taxiqueue

External Interface File: gps coordinates, map service

External Input: login, logout, request, reserve, delete, reserve shared, accept call,

refuse call, report, taxi available, taxi not available, change settings

External Output: message (eta, no taxi message)

External Inquiry: see profile, see active ride list

2.1 Complexity and cost evaluation

2.1.1 Internal Logic File

According to our previous specification (explained in the RASD and DD documents), users and ride have to store few informations, thus we can adopt the simple cost weight for those ones. On the other hand, **sharedride** and **taxiqueue** have to store a dynamic list, that require more attention, so we adopt a medium cost weight.

$$4 * 7 + 2 * 10 = 48 \text{ FPs} \tag{2.1}$$

2.1.2 External Interface File

The interactions with **gps** coordinates and the **map service** are very simple, because we need to gather few information from them, so we adopt a simple weight for both of External Internal Files.

$$2 * 5 = 10 \text{ FPs}$$
 (2.2)

2.1.3 External Input

Most of the external inputs are simple actions involving a few number of entities, therefore we can adopt a simple weight cost for all of them. **request** and **change settings** however are more complex, and thus require a medium weight cost.

$$10 * 3 + 2 * 4 = 38 \text{ FPs} \tag{2.3}$$

2.1.4 External Output

Sending **eta** requires to access the map service that calculate, on its own, the appropriate value, so we adopt a simple cost weight for message.

$$2*4 = 8 \text{ FPs}$$
 (2.4)

2.1.5 External Inquiry

see profile requires only to send some fields saved in the current user, while see active ride list requires to scan the ridehistory and check its status (active or not). Therefore, we adopt a simple cost weight for the former, and a medium cost weight for the latter.

$$1*3+1*4=7$$
 FPs (2.5)

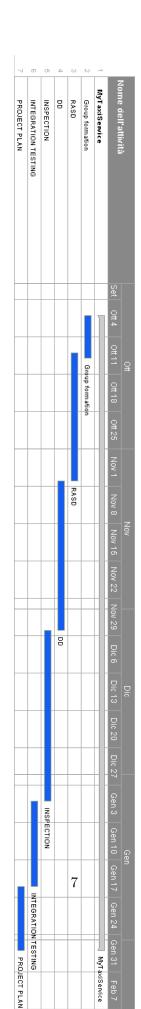
2.1.6 Overall

In summary we have
$$FPs = \sum_{i=1}^{5} FPi = 111$$

COCOMO II

Schedule and resources allocation

4.1 Gantt's diagram



Risk evaluation

5.1 Risk evaluation and avoidance

Risk	Probability	Effects
1.Key staff are ill at critical times in the project	Moderate	Serious
2. Changes to requirements that require major design rework	Moderate	Serious
3. Loss of data	Low	Catastrophic
4. Poor collaboration among team members	Moderate	Serious

Risk	Strategy
1.	Each member is aware of the job done by other components so that he can re-
	view/finish the task if someone gets sick
2.	Pay attention and if needed ask for clarification
3.	Keep all material synchronized with Github
4.	Keep in good relationship and talk about the project issues