Politecnico di Milano
Dipartimento di Elettronica, Informazione e Bioingegneria

myTaxiService
Cost Estimation

Supervisor:
Prof. Elisabetta Di Nitto

Students:
Alberto Gasparin
Vito Matarazzo

Year 2015-2016

Contents

Revision History

Date	Reason for changes	Version
February 2, 2016		v1.0

1 Introduction

1.1 Purpose and Scope

This document proposes a project plan for myTaxiService application. In the document an effort estimation and a cost estimation are presented applying Function Points and COCOMO II. Moreover, the main tasks of the project are identified and an estimation of both time and resources to be allocated to each task is performed. The last section of the document presents an overview of potential problems (risks) that might happen or not along the duration of the project.

1.2 Definitions, Acronyms, Abbreviations

1.2.1 Acronyms

DD: Design Document

DB: Database

FP: Function Point

ITP: Integration Test Plan

RASD: Requirement Analysis and Specification Document

1.3 Reference Document

Specification Document: myTaxiService Project AA 2015-2016.pdf.

RASD, Alberto Gasparin, Vito Matarazzo, v1.1, November 2015.

DD, Alberto Gasparin, Vito Matarazzo, v1.0, December 2015.

ITP, Alberto Gasparin, Vito Matarazzo, v1.0, January 2015.

COCOMO II - Model Definition Manual, v2.1

2 Function Point

This methodology determines a cost estimation considering the functionalities of the system, in other words we will consider data and processes significant to the end user. The method quantifies the information stored in the system and those ones going in and out of the system. The functionalities list has been obtained from the RASD document and for each one of them the realization complexity has been evaluated. First we will briefly describe the FP types, that are:

- Internal Logic File: it represents a set of homogeneous data handled by the system.
- External Interface File: it represents a set of homogeneous data used by the application but handled by external application.
- External Imput: elementary operation that allows input of data in the system.
- External Output: elementary operation that creates a bitstream towards the outside of the application.
- External Inquiry: elementary operation that involves input and output operations.

The following table outlines the number of Functional Points based on funtionality and relative complexity:

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

Now, we will procede computing the FP for each FP type.

2.1 Internal logical file (ILF)

ILFs included in the application are user, taxi, ride, reservation and zone. The information stored about user are those requested in the registration procedure (firstname, lastname, email, username, password, ...) and a few more fields are included, like a type field used to identify the user type (passenger, driver, developer, administartor), the driving license for taxi drivers and the assigned APIkey for developer's only. Taxi information are needed in order to monitor the fleet of the taxi company. Those information include the taxi code, the driver and the number of available seats. Ride and reservation are two entities that are updated on a regular basis, and they are mapped to several different entities so their managing is not simple. Finally the system has to manage zones in which the city has been divided. For each zone the coordinates of the vertices are stored in the system. The system uses this information to map a specific location into a certain zone.

ILF	Complexity	FP
User	Medium	10
Taxi	Simple	7
Ride	Medium	10
Reservation	Medium	10
Zone	Medium	10
Total		47

2.2 External interface file (EIF)

The system has to acquire external data like ETA and best path available using GoogleMapsAPI. The data are returned in JSON format, our system will be able to interpret them and supply them to the users(Passenger and Taxi driver). Moreover GoogleMapsAPI are used to obtain the coordinates of a certain location (pickup, destination). Those coordinates will be used to compute the zone corresponding to the location submitted.

ELF	Complexity	FP
ETA	Simple	5
Optimal path	Complex	10
Coordinates	Simple	5
Total		20

2.3 External input (EI)

The application interacts with the user to allow him/her to:

- Login/Logout: these are simple operations because only one entity is involved (User).
- Register: this is a simple operation because only one entity is involved (User).
- Update profile: this is a simple operation because only one entity is involved (User).
- Request a taxi: this is a Complex operation because it involves at least 4 entities.
- Reserve a taxi: this is a Complex operation because it involves at least 4 entities.
- Cancel a reservation: this is a simple operation because it involves only one entity.
- Change the current status: this is a simple operation.
- Signal the occurrence of an unexpected event: this is a medium operation because even if it does not involve directly a lot of entities, it triggers a chain of events of average complexity.

- Request an APIkey: this is a simple operation because it involves only one entity (User).

EI	Complexity	FP
Login/Logout	Simple	2x3
Register	Simple	3
Update profile	Simple	3
Request a taxi	Complex	6
Reserve a taxi	Complex	6
Cancel a reservation	Simple	3
Change the current status	Simple	3
Signal unexpected event	Medium	4
Request APIkey	Simple	3
Total		37

2.4 External output (EO)

Once a user is logged into the system he/she can be notified about incoming requests (for the taxi driver) or proposals (for the passenger). Moreover the passenger will receive updates after requesting a taxi (for example the passenger is notified if the expected taxi driver cannot arrive at the desired place due to some unexpected events). All these notifications involve two entities so we can consider them of medium complexity. Finally the system computes and shows to the requesting developer his/her APIkey, which is a simple operation.

EI	Complexity	FP
Notifications	Medium	3x5
Show APIkey	Simple	4
Total		19

2.5 External inquiry (EQ)

The application allows users to visualize information about:

- Ride history: displays the information about the past rides and reservations of a user. It involves three entities so we can consider it as a medium operation.

- User profile: displays the information about the user profile. It involves only the user entity so it's a simple operation.

EI	Complexity	FP
Show ride history	Medium	4
Show user profile	Simple	3
Total		7

2.6 UFP

Given the values of each function point count, we obtain the following Unadjusted Function Point (UFP) value:

FP type	**FP**
ILF	47
EIF	10
EI	37
EO	19
EQ	7
Total	120

The final result can be adjusted to obtain the FP value, which is an estimation of the effort of the project, but this does not always improve the estimation. The better choice is to use the UFP value in combination with the COCOMO approach.

3 COCOMO II

COCOMO is a technique used to estimate the effort required for the development of a software product. This estimation is achieved through a complex, non linear model that takes into account the characteristics of product, people and process. The main elements of COCOMO II model are: SLOC (Source Lines Of Code), scale drivers, cost drivers and the effort equation.

3.1 SLOC

The effort estimation of COCOMO model is based on estimates of the project's size, expressed in source lines of code(SLOC). Im our case, this estimate comes from the UFP count calculated in the previous section multiplied by an average conversion factor of 46, which is the factor associated to J2EE programming language. The result is:

$$SLOC = 46 * UFP = 5520$$

3.2 Scale drivers

Scale drivers are the most important factors contributing to a project's duration and cost. We used the following table taken from the COCOMO II manual to evaluate them:

Table 10. Scale Factor Values, SF_j, for COCOMO II Models

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:	5.07	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	The estimated Equivalent Process Maturity Level (EPML) or					
	SW-CMM Level 1 Lower	SW-CMM Level 1 Upper	SW-CMM Level 2	SW-CMM Level 3	SW-CMM Level 4	SW-CMM Level 5
SF_j:	7.80	6.24	4.68	3.12	1.56	0.00

- Precedentness: reflects the previous experience that we had with this kind of projects. Since for us this is the first experience of designing and developing a project like this, this value will be low.

- Development flexibility: reflects the degree of flexibility in the development process. Since the project assignments contained only general specifications without going too much in detail, this value will be high.
- Risk resolution: reflects the extent of risk analysis carried out. We have taken into account the most important risks in our project, doing an average risk analysis, so this value will be nominal.
- Team cohesion: reflects how well the development team know each other and work together. In our case we had already worked together in some other projects without particular issues, so this value will be very high.
- Process maturity: reflects the process maturity of the organization. This is evaluated using the CMMI framework to establish the level of maturity, and we obtained a level 2, so nominal value.

The results are resumed in the following table:

Scale Driver	**Scale Factor**	**SF value**
Precedentness	low	4.96
Dev. flexibility	high	2.03
Risk resolution	nominal	4.24
Team cohesion	very high	1.10
Process maturity	nominal	4.68
Total		17.01

3.3 Cost drivers

Cost drivers are multiplicative factors used to evaluate different characteristics of the project. Using an average value of nominal for all the cost drivers, we obtained the following results:

Cost Driver	Rating level	Multiplier
RELY	low	0.92
DATA	nominal	1.00
CPLX	nominal	1.00
RUSE	high	1.07
DOCU	nominal	1.00
TIME	nominal	1.00
STOR	nominal	1.00
PVOL	low	0.87
ACAP	high	0.87
PCAP	high	0.88
PCON	very low	1.29
APEX	very low	1.22
PLEX	low	1.09
LTEX	nominal	1.00
TOOL	nominal	1.00
SITE	high	0.93
SCED	nominal	1.00
Total Product		1.046

3.4 Effort Equation

The effort equation gives us the effort estimation measured in Person-Months (PM):

$$Effort := A * EAF * KSLOC^{E}$$

Where:

- A := 2.94 (for COCOMO II);
- EAF := product of all the cost drivers, equal to : 1.046 ;
- E := exponent derived from Scale Drivers. Is calculated as:

  $$B + 0.01 * \sum_{i} SF[i] := B + 0.01 * 17.01 = 0.91 + 0.1701 = 1.0801; \quad (1)$$

  in which B is equal to: 0.91 for COCOMO II;
- KSLOC := estimated lines of code using the FP analysis: 5.520;

With these parameters we can compute the Effort value, that is equal to:

$$Effort := 2.94 * 1.046 * 5.520^{1.0801} = 19.4647PM$$

3.5 Schedule Estimation

The schedule equation predicts the number of months required to complete a software project. The duration is computed as:

$$Duration := 3.67 * Effort^{SE}$$

Where:

$$SE := 0.28 + 0.2 * (E–B) = 0.31402$$

Follows then:

$$Duration := 3.67 * 19.4647^{0.31402} = 9.32 \simeq 9$$

Finally we can calculate the number of required people N:

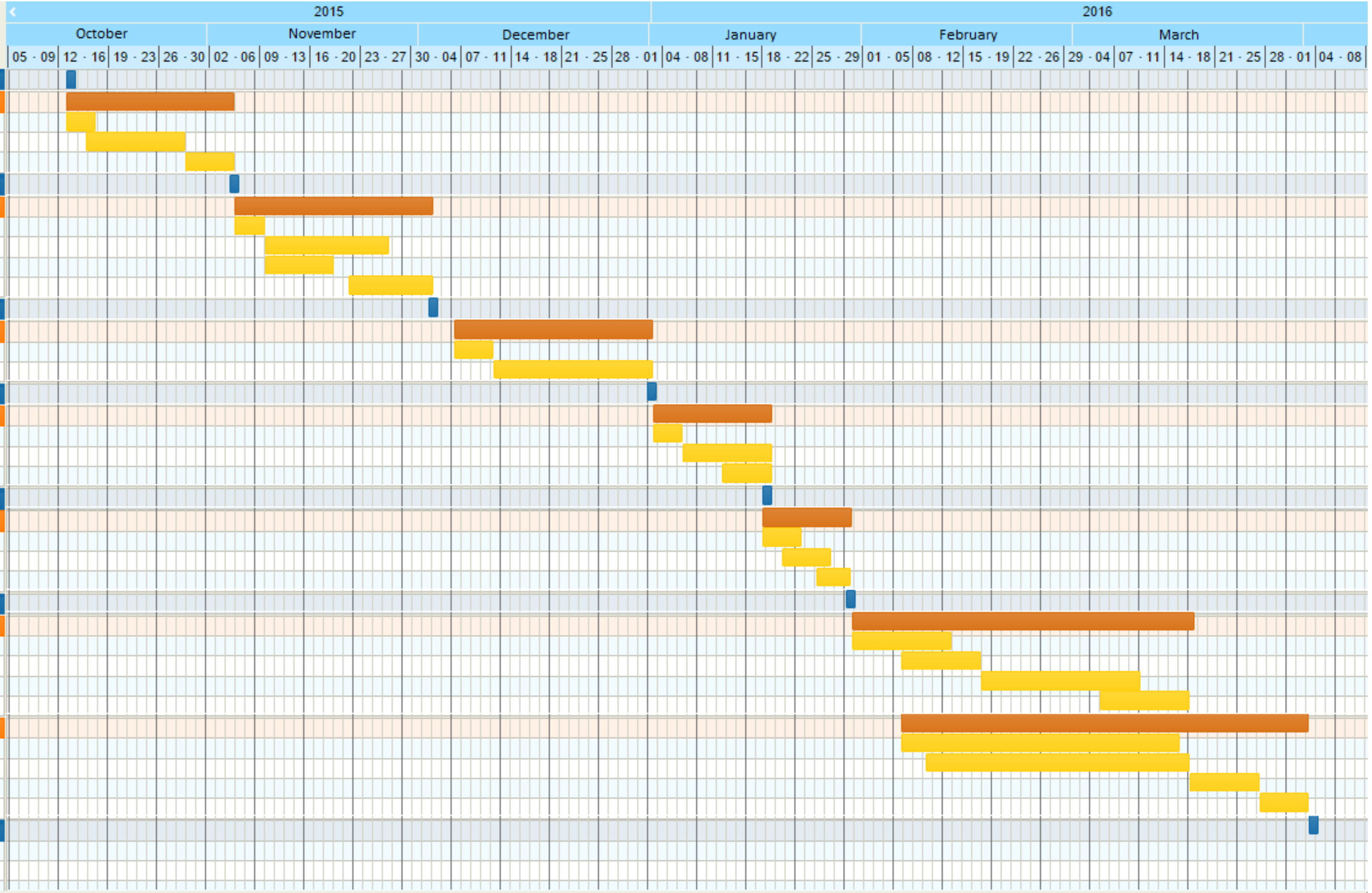
$$N := Effort/Duration = 2.16 \simeq 2$$

Which is exactly the number of people in our group.

4 Schedule of the project

The following Gantt chart shows the different tasks of the project and their schedule. It contains the actual schedule of the activities completed so far, and an estimation of the following tasks to be performed. We assumed that a Unit Test Plan was done in the period between 9/12/2015 and 5/01/2016 instead of the code inspection assignment.

Task ID	Activity	Status
	Project assignment	**15-10-2015**
	RASD	**Complete**
T1	Def . of goals and assumptions	Complete
T2	Requirements specification	Complete
T3	Alloy model specification	Complete
	RASD delivery	**6-11-2015**
	Design document	**Complete**
T4	Def of high level architecture	Complete
T5	Arch. and design specification	Complete
T6	Alghorithm design	Complete
T7	UI design	Complete
	Design document delivery	**4-12-2015**
	Unit test plan	**Complete**
T8	Def. of testing strategy	Complete
T9	Individual tests description	Complete
	Unit test plan delivery	**5-01-2016**
	Integration test plan	**Complete**
T10	Def. of integration strategy	Complete
T11	Individual steps description	Complete
T12	Specification of tools and data	Complete
	ITP delivery	**21-01-2016**
	Project planning	**Complete**
T13	Estimation of size and effort	Complete
T14	Scheduling	Complete
T15	Risk analysis	Complete
	Project plan delivery	**2-02-2016**
	Development	**Pending**
T16	Development of clients	Pending
T17	Development of UI logic	Pending
T18	Development of server backend	Pending
T19	Database setup	Pending
	Testing	**Pending**
T20	Unit testing	Pending
T21	Integration testing	Pending
T22	System testing	Pending
T23	Performance/ Stress testing	Pending
	Expected delivery	**7-4-2016**



project: myTaxiService
authors: Alberto Gasparin
Vito Matarazzo

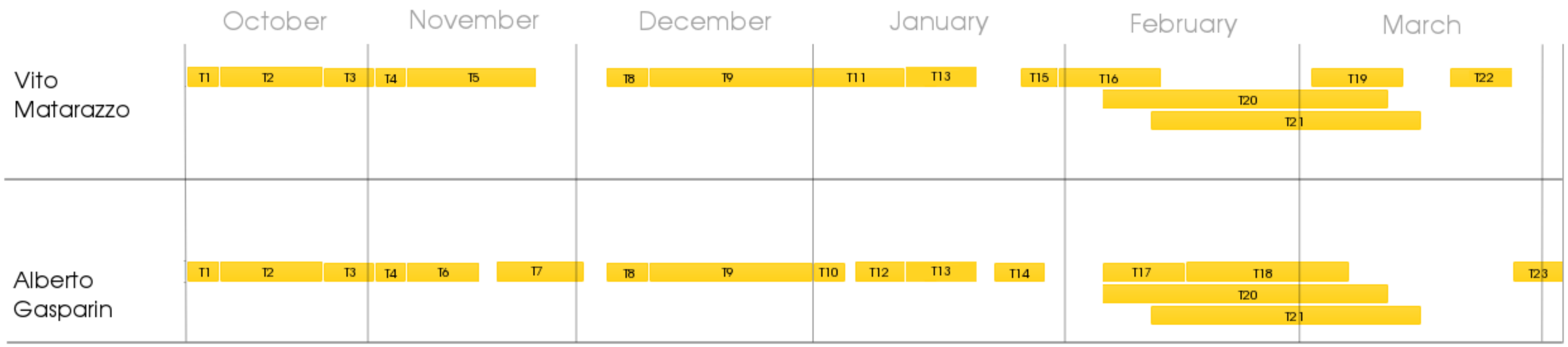
Primary activities
Specific activities
Milestones

By comparing the duration calculated by COCOMO estimation with the estimated duration of about 6 months indicated in this schedule, we have a difference of about 3 months. This can be due to two possibilities:

- some of the estimations we made are not appropriate for the actual problem we are analysing;
- the team will have more time to dedicate to the project from the beginning of February to the delivery day, so it is likely that the amount of daily work will increase.

5 Resource allocation

Starting from the tasks identified in the previous section, the following is a bar chart that represents the staff allocated to each task. This allocation is as close as possible to the real tasks' division between the members of the group.



6 Risk analysis

In this section we will define the main risks that the project can encounter. Risks are potential problems that can cause unwanted consequences or losses if they become real problems. So the following analysis is aimed at identifying the different risks, estimating their probability and impact, and explaining the associated recovery actions to perform in case of risk realization.

6.1 Risk Identification and ranking

In the following table we list the risks that may be encountered during the project development. The risks are listed in decreasing order w.r.t the impact they may have on the project itself.

Risk	Probability	Effects
Key personnel leave the project taking critical information with them and subsequent impossibility to recruit new personnel with the required skills.	Low, since the team is composed of only two members which don't have any reason to leave the project.	Catastrophic, because, as said, the team is composed of two members so if one leaves, the other would find hard to complete the project.
Key staff are unavailable at critical times during the project.	Moderate, because unexpected problems could always happen.	Critical, for the same reason of the previous risk, but not catastrophic since the lack of members would not be definitive.
Changes to requirements that require major design rework are proposed.	Moderate, because even if at this time of the project it is unlikely that the requirements change, the design choices may be not definitive because of our inexperience.	Critical, because design changes could delay the whole project.
Expansion of project scope, due to addition of features and/or requirements.	Low, because the initial assignment has never been changed	Critical, because this could expand the time necessary to accomplish the project; in a real world scenario this may also lead to budget overruns.
Failure to manage end user expectation.	Moderate, because users of the application could find it difficult to use some features, especially in the first releases of the software	Critical, because if the application doesn't satisfy user expectation, it could require major requirements or design revision.
Misunderstanding of requirements.	Low, because at this time of the project the requirements have been reviewed many times.	Critical, because it can lead to important revisions in the initial phases of the project.
Faults in reusable software components have to be repaired before using them.	Moderate, since our inexperience could lead to wrong choices of the components to be used	Critical, because if the components have faults the whole application could work wrong.
The database used in the system cannot process as many transactions per second as expected	Moderate, since our inexperience could lead to a wrong choice of the database to be used.	Critical, because if the database has faults the whole application could work wrong.
Unrealistic time estimate	High, since this is our first experience in this kind of projects, the estimates might not be correct.	Marginal, because this is a university project; in a real world scenario this would be critical, because a delay in a critical activity may have a cascading effect on the entire project.

6.2 Strategy

In the following table we propose a strategy for each of the risk identified above.

Risk	**Strategy**
Personnel leaves	Increase the collaboration and information sharing among the team; if not enough, investigate recruiting new members
Key staff unavailable	Increase the collaboration and information sharing among the team, so that people understand each other's job.
Requirements changes	Use requirement traceability table in order to identify those components that need to be modified.
Expansion of project scope	Increase collaboration between customers and developers, plan regular discussion about features and estimates on a regular basis.
Failed user expectation	Collect feedback from a group of users that act as beta testers.
Misunderstanding of requirements	Increase collaboration between customers and requirement engineers in order to be sure that what is being produced matches the customers' needs.
Defective components	Replace potentially defective components with bought-in components of known reliability.
Database performance	Investigate the possibility to adopt a different database.
Unrealistic time estimate	Plan for two releases, the first one with a limited set of functionalities and the second one more complete.

7 Appendix

7.1 Software Tool used

- TexMaker (http://www.xm1math.net/texmaker/): LaTeXeditor, used to redact this document.

7.2 Hours of work

- Alberto Gasparin $\sim$ 12 h
- Vito Matarazzo $\sim$ 12 h