Project Plan

Alessandro Comodi, Davide Conficconi, Stefano Longari

January 27, 2016

Contents

Chapter 1

Introduction

The chapter explains what this document is intended for and all the tools in order to understand it in the correct way.

1.1 Scope & Purpose

This document has the objective to evaluate the time and the resources necessary to the development of MyTaxiService through two approaches: first will be used the Function Points approach to estimate the size of the output project and then the COCOMO approach mainly to study the time necessary to the development of the application.

1.2 Definitions & Abbreviations

In the following list are present all the meanings of the abbreviations used in this document:

- FP: Function Points;
- LOC: Lines Of Code;

- SLOC: Source Lines Of Code;
- ILF: Internal Logic Files;
- ELF: External Logic Files;
- EI: External Input;
- EIQ: External Inquiries;
- EO: External Outputs;
- CR: Conversion Ratio.

1.3 Reference Documents

The documents used as reference in order to redact the Project Plan are:

- "COCOMO II Model Manual v2.1;
- Requirement Analysis and Specification Document (RASD) for MyTaxiService;

Chapter 2

Project Estimation

In this chapter there is an analysis about the size, costs and effort needed in order to achieve the project goals. The first part regards the estimation of the size of the project through the attribution of the Function Points. The second part aims to give an effort estimation using COCOMO II.

2.1 Project Size Estimation

Function Points can give an estimation of the LOC that will be produced in the developement process of the project. They refer to the various functionalities of the software and are divided in different groups of function types, which can be seen in the table below.

Function Types	**Weight**		
	Simple	Medium	Complex
Internal Logic Files	7	10	15
External Interface Files	5	7	10
External Inputs	3	4	6
External Inquiries	3	4	6
External Outputs	4	5	7

2.1.1 Internal Logic Files

The ILFs to be stored and managed by the system regard users, taxi drivers and rides.

- Users' weight can be considered as simple as the information about the user to store are just a few (name, surname, password, email, ...).
- Taxi driver's complexity can be treated as the users' one for the same reason, so the function type's weight is simple.
- Ride needs to link both information from the user and the taxi drivers and it stores many information, differently from the user files. The ride files are relevant and they must be treated with a high complexity.

The total function points assigned to the ILF function type is found in the table below:

ILF	**Weight**	**FP**
User	Simple	7
Taxi Driver	Simple	7
Ride	High	15
Total		29

2.1.2 External Logic Files

The ELFs needed to the system are gps information and files for the maps.

- GPS information need to be decoded, processed and accurately stored in the system. Due to the high complexity of this interaction with the GPS services the weight is set as high
- Maps are obtained from external sources, but the files contain a small enough quantity of information to be stored, so this logic files have a simple complexity weight.

ELF	**Weight**	**FP**
GPS information	High	10
Maps	Simple	5
Total		15

2.1.3 External Inputs

The EI that can be performed are:

- login/logout: these two inputs have a simple weight, this due to the fact that the operations to be performed are just a few (receive data, check them and send a message).
- create ride: this input has a high weight because there are many steps to perform in order to complete the creation of a ride, for instance the matching between user and taxi driver or the calculation of the ride route.
- create/delete reservation: the creation and deletion of a reservation ride are classified as high weight, for the same reason of the previous point.
- accept/refuse ride: these inputs need many operations to be completed suitably (deallocation of the taxi driver from the queue, ...). The complexity assumes a medium weight.

- insert/update/delete user: the weight of these inputs is simple, as the user (as said previously) has only a few information to be updated, inserted or deleted.
- insert/update/delete taxi drivers: these inputs are treated as the previous ones regarding the user.

EI	Weight	FP
login/logout	Simple	2*3
create ride	High	6
create/delete reservation	High	2*6
accept/refuse ride	Medium	4*2
insert/update/delete user	Simple	3*3
insert/update/delete taxi driver	Simple	3*3
Total		50

2.1.4 External Inquiries

The EIQs that are present are:

- user profile: the set of operations needed in order to show the user profile is low as the information to retrieve are reagarding only the user. The complexity is simple.
- taxi driver profile: same as the previous point.
- ride history: the operations that have to be done so that the user can check all the rides are many thus the complexity is medium.
- notifications: the operation of sending and showing notification is simple.

EIQ	Weight	FP
User profile	Simple	3
Taxi Driver profile	Simple	3
Ride history	Medium	4
Notifications	Simple	3
Total		13

2.1.5 External Outputs

The system does not require any External Output thus no FP are being assigned to this function type.

2.1.6 Function Points

All the calculated Function Points can be summarized in the following table:

Function Type	Value
Internal Logic Files	29
External Logic Files	15
External Inputs	50
External Inquiries	13
External Outputs	-
Total	107

We suppose that the software is developed using Java programming language and the conversion from FP to SLOC is the following:

$SLOC = FP * CR = 107 * 53 = 5671$

$KSLOC = 5.7$

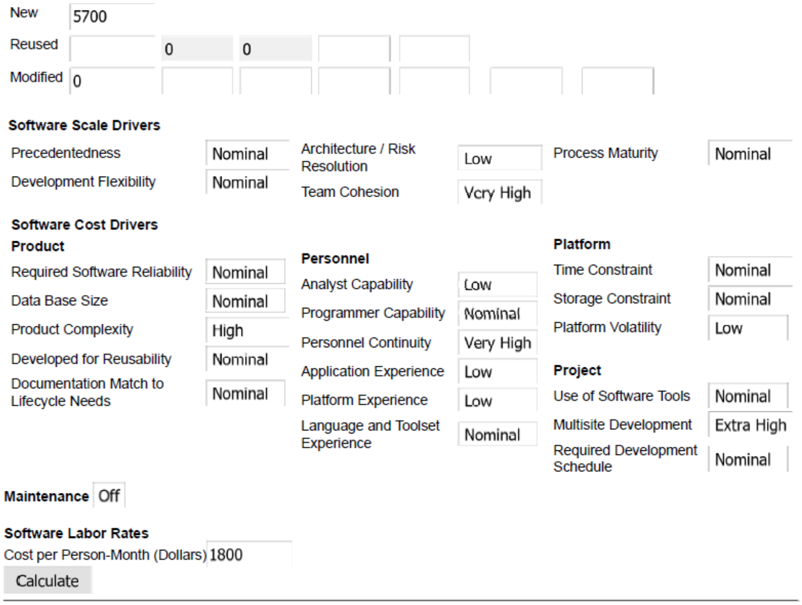
The value of the Conversion Ratio is taken from “COCOMO II Model Manual v2.1” (page 6).

2.2 Project Effort and Cost Estimation

Thanks to the use of COCOMO II (COnstructive COst MOdel), using the SLOC estimation done in the previous section, we could estimate the Cost and Effort needed to complete the project.

In Figures 2.1 and 2.2 there is the input and output of COCOMO II.

Figure 2.1: COCOMO II input



New	5700		
Reused		0	0
Modified	0		

Software Scale Drivers

Precedentedness	Nominal	Architecture / Risk Resolution	Low	Process Maturity	Nominal
Development Flexibility	Nominal	Team Cohesion	Vcry High		

Software Cost Drivers

Product

Required Software Reliability	Nominal
Data Base Size	Nominal
Product Complexity	High
Developed for Reusability	Nominal
Documentation Match to Lifecycle Needs	Nominal

Personnel

Analyst Capability	Low
Programmer Capability	Nominal
Personnel Continuity	Very High
Application Experience	Low
Platform Experience	Low
Language and Toolset Experience	Nominal

Platform

Time Constraint	Nominal
Storage Constraint	Nominal
Platform Volatility	Low

Project

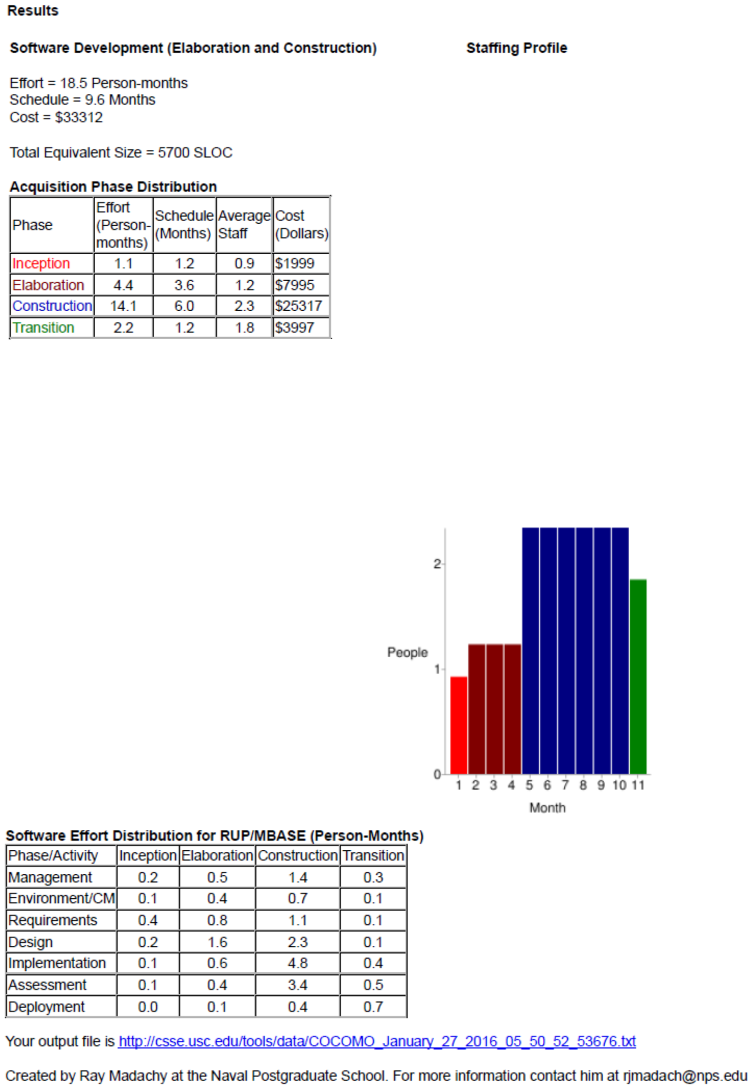
Use of Software Tools	Nominal
Multisite Development	Extra High
Required Development Schedule	Nominal

Maintenance Off

Software Labor Rates

Cost per Person-Month (Dollars) 1800

Figure 2.2: COCOMO II output



Results

Software Development (Elaboration and Construction)

Effort = 18.5 Person-months
Schedule = 9.6 Months
Cost = $33312

Total Equivalent Size = 5700 SLOC

Acquisition Phase Distribution

Phase	Effort (Person-months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	1.1	1.2	0.9	$1999
Elaboration	4.4	3.6	1.2	$7995
Construction	14.1	6.0	2.3	$25317
Transition	2.2	1.2	1.8	$3997

Staffing Profile

Software Effort Distribution for RUP/MBASE (Person-Months)

Phase/Activity	Inception	Elaboration	Construction	Transition
Management	0.2	0.5	1.4	0.3
Environment/CM	0.1	0.4	0.7	0.1
Requirements	0.4	0.8	1.1	0.1
Design	0.2	1.6	2.3	0.1
Implementation	0.1	0.6	4.8	0.4
Assessment	0.1	0.4	3.4	0.5
Deployment	0.0	0.1	0.4	0.7

Your output file is http://csse.usc.edu/tools/data/COCOMO_January_27_2016_05_50_52_53676.txt

Created by Ray Madachy at the Naval Postgraduate School. For more information contact him at rjmadach@nps.edu

2.2.1 Scale Drivers

The COCOMO tool needs to define these parmaters to determine the exponent used in the Effort Equation. There are five Scale drivers, that are defined in table 2.3: precedentness, development flexibilty, arcihtecture or risk resolution, team cohesion and process maturity

Figure 2.3: Scale Drivers Factor

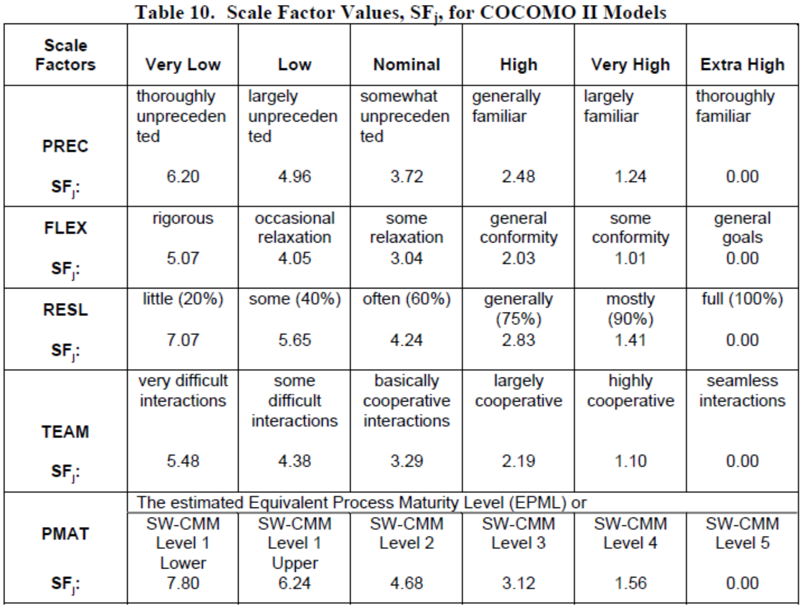


Table 10. Scale Factor Values, SF$_j$, for COCOMO II Models

Scale Factors	Very Low	Low	Nominal	High	Very High	Extra High
PREC SF$_j$:	thoroughly unprecedented 6.20	largely unprecedented 4.96	somewhat unprecedented 3.72	generally familiar 2.48	largely familiar 1.24	thoroughly familiar 0.00
FLEX SF$_j$:	rigorous 5.07	occasional relaxation 4.05	some relaxation 3.04	general conformity 2.03	some conformity 1.01	general goals 0.00
RESL SF$_j$:	little (20%) 7.07	some (40%) 5.65	often (60%) 4.24	generally (75%) 2.83	mostly (90%) 1.41	full (100%) 0.00
TEAM SF$_j$:	very difficult interactions 5.48	some difficult interactions 4.38	basically cooperative interactions 3.29	largely cooperative 2.19	highly cooperative 1.10	seamless interactions 0.00
PMAT	The estimated Equivalent Process Maturity Level (EPML) or					
SF$_j$:	SW-CMM Level 1 Lower 7.80	SW-CMM Level 1 Upper 6.24	SW-CMM Level 2 4.68	SW-CMM Level 3 3.12	SW-CMM Level 4 1.56	SW-CMM Level 5 0.00

Our organisation has a previous experience in project of this kind, the development process has normal degree of flexibility and our process can be seen at level 2 or 3 of CMMI. According to the Document COCOMO® II.2000.0 Model Manual v2.1 this brings us to set this value to Nominal, while the poor analysis of risk carry out a low value instead our team cohesion is high.

2.2.2 Cost Drivers

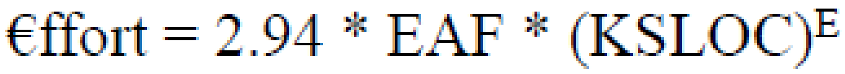
Cost Drivers are multiplicative factors that determine the EAF, a factor required in the computation of the effort, derived by the charts in COCOCMO® II.2000.0 Model Manual v2.1. As it can be seen in 2.1, in order to compute the effort through the tool we've set the Software Cost Drivers accordingly to the referenced document. We conform our experience to the definition in the document that brings us to this set of value for the Cost Drivers.

For example the parameter Personnel Continuity is set to Very High due to the fact that our team is stable and not subject to change of people, while the Application Experience cost driver is set to Low because our team experience in develop application of this kind is lesser than one year.

2.2.3 Effort Estimation

The effort estimation through COCOMO can be computed using the equation in 2.4: where EAF is the Effort Adjustment Factor derived by a product of the Cost Drivers multipliers and E is the Exponent computed according to the Model Manual procedure through Scale Drivers factor.

Figure 2.4: Effort Equation



$$\text{€ffort} = 2.94 * EAF * (KSLOC)^{E}$$

We compute the effort with the tool on the CSSE site about COCOMO , as we can see in 2.2. Despite the tool analysis our team is based on three people so we estimate a duration of the project is about six months. If we speculate a compensation of 1800 dollars/month and a project period of 6 months, the project has a cost estimation of approximately 32K$ circa .

Chapter 3

Project Schedules & Resources Allocation

Introduction for the Teacher: The following schedule starts at the developing point we actually are, so we take as granted the documents (RASD,DD,ITPD) we already did, even if the project plan should have been probably done before. This to say that to create the schedule we used all the information we had about testing and the developing order of the application.

Also, we want to clarify that the time spent before now is really excessive. The three months that passed represent the first one of the six for each person that we predicted with COCOMO II.

In February we will start implementing the lower level components (Queue Manager, Ride Manager, Web Client, Mobile Manager and DBMS). As the two Managers are the core of the project they will be the components that will take the most of the time, so the developers working on the Clients and the DBMS will converge on the Manager as soon as they will have finished their parts. This part of the project should take about one month, after which these components will be tested (the Managers need the creation of the Driver for the application controller) possibly in the first half of March.

The Middle level components (Application Controller, Data access Manager, Web Server and Connection Handler) should be finished in the first half of may, after which they will be tested. This parts should actually take about the same developing time so each developer should work only on one group of components.

In the end, in the second part of May and until the end of June we will develop and test the Security Manager and integrate all the parts, finally being able to put the application online.

Month	Tasks	Developer
February		
	Queue Manager	Alessandro, Davide, Stefano
	Ride Manager	Alessandro
	Web Client	Davide
	Mobile Client	Davide
	DBMS	Stefano
March		
	Application Controller Driver	Alessandro
	Lower level components testing	Alessandro, Davide, Stefano
	Application Controller	Alessandro
	Data Access Manager	Stefano
	Web Server & Connection Handler	Davide
April		
	Application Controller	Alessandro
	Data Access Manager	Stefano
	Web Server & Connection Handler	Davide
May		
	Security Manager	Stefano
	Subsystem Linking	Alessandro, Davide
June		
	Testing Subsystems	Alessandro, Davide, Stefano
	Integration of all subsystems	Alessandro, Davide, Stefano
	Deployment	Alessandro, Davide, Stefano

Chapter 4

Project Risks

In this chapter there is an analysis of the possible risks regarding the project.

- Loss of personnel:
  - Probability: of this risk is very low, though still present (the team member could get sick or have personal issues).
  - Relevance: moderate, as the project could have delays if the team member is not replaced.
  - Recovery Actions: The team member must be replaced and updated in order to understand his duties.
- Change in the Requirements:
  - Probability: the probability that the client who commissioned the project changes some of the requirements is low.
  - Relevance: moderate.
  - Recovery Actions: The team has to correct all the changes in the requirements modifing all the concerned documents.

- Wrong time estimation:
  - Probability: medium.
  - Relevance: moderate.
  - Recovery Actions: change in the scheduling and in the assignments of the tasks to the various team members.
- No proper subject training:
  - Probability: low.
  - Relevance: moderate.
  - Recovery Actions: External consulence or self teaching.
- Wrong budget estimation:
  - Probability: low.
  - Relevance: medium.
  - Recovery Actions: redistribute the budget and try to find some compromises with the funder.

Chapter 5

Other Info

5.1 Working Hours

	Alessandro	Davide	Stefano
PP hours	10	10	10
Total hours	76	78	76.5

5.2 Used Tools

- Text editors;
- COCOMO II tool : http://csse.usc.edu/tools/COCOMOII.php
- GitHub;
- LyX and LaTeX.