POLITECNICO
MILANO 1863

MyTaxiService

Project Plan Document

Davide Citterio, Lorenzo Cunial, Massimo Beccari

February 1, 2016

Contents

1 Introduction

Project plan document aim to define time schedule, resource and people needed for the development of MyTaxyService project. About resource and people needed, we have used the function points approach. Moreover we have used COCOMO Model to evaluate the effort for MTS implementation. We have also split the project in several tasks, liked each task with time and a developer. We have used Grantt diagram to show this aspect.

2 Size, Effort and Duration of the project

2.1 Size - Function Points Approach

The function points cost estimation approach is based on the amount of functionality in a software project and a set of individual project factors. About this project we take the functionality list from the RASD document of MTS. Function points are useful estimators since they are based on information that is available early in the project life-cycle. Function points measure a software project by quantifying the information processing functionality associated with major external data or control input, output, or file types. Five user function types should be identified as defined in the following Table.

Function Point	Description
External Input (EI)	Count each unique user data or user control input type that enters the external boundary of the software system being measured.
External Output (EO)	Count each unique user data or control output type that leaves the external boundary of the software system being measured.
Internal Logical File (ILF)	Count each major logical group of user data or control information in the software system as a logical internal file type. Include each logical file (e.g., each logical group of data) that is generated, used, or maintained by the software system.
External Interface Files (EIF)	Files passed or shared between software systems should be counted as external interface file types within each system.
External Inquiry (EQ)	Count each unique input-output combination, where input causes and generates an immediate output, as an external inquiry type.

Figure 1: function types

The following table outline the number of Functional Point based on functionality and relative complexity:

Function Type	Complexity-Weight Low	Average	High
Internal Logical Files	7	10	15
External Interfaces Files	5	7	10
External Inputs	3	4	6
External Outputs	4	5	7
External Inquiries	3	4	6

Figure 2: complexity-weight relation

About the determination of complexity level we use the table below:

For Internal Logical Files and External Interface Files

Record Elements	Data Elements 1 - 19	20 - 50	51+
1	Low	Low	Avg.
2 - 5	Low	Avg.	High
6+	Avg.	High	High

For External Output and External Inquiry

File Types	Data Elements 1 - 5	6 - 19	20+
0 or 1	Low	Low	Avg.
2 - 3	Low	Avg.	High
4+	Avg.	High	High

For External Input

File Types	Data Elements 1 - 4	5 - 15	16+
0 or 1	Low	Low	Avg.
2 - 3	Low	Avg.	High
3+	Avg.	High	High

Figure 3: complexity level

2.1.1 Internal Logical Files

Application has to store data about users, drivers, administrator, rides reservation and request and global application settings. About Administrator and drivers, they have a small structure, instead users have a medium structure. About rides reservation and request, they have a high structure. Global settings have a low structure.

Users	10
Drivers	7
Administrators	7
Rides Data	10
Global Settings	15
Total ILF	**49**

2.1.2 External Interface Files

About the external interface files we have only the Google Maps API, that has a low level of complexity.

Google Maps API	5
Total EIF	**5**

2.1.3 External Input

About the external input we identify this elements: login, logout and sign in that have small weight, change reservation for the user has a small weight, set ready and status for the driver has a small weight, change settings of user has a small weight, change global settings and enable/disable services of the system has a small weight, driver report has a small weight, add and remove driver has small weight.

User login	3
User logout	3
Driver login	3
Sign in	3
Change reservation	3
Set ready	3
Set driver status	3
Change user settings	3
Change global settings	3
Managing additional services	3
Driver report	3
Add driver	3
Remove driver	3
Total EI	**39**

2.1.4 External Output

About the external output we identify the SMS messaging service and the driver call. Both them have a small weight.

SMS messaging	4
Driver call	4
Total EO	**8**

2.1.5 External Inquiries

About the external inquiries we identify user and administrator setting view with small weight, user request chronology with small weight, admin map request with medium weight, driver's accept call with small weight, driver's request bill with small weight, user request, reservation and reservation shared with medium weight.

User settings view	3
Administrator settings view	3
Request chronology	3
Administrator rides chronology	3
Administrator map	4
Request bill	3
Request	4
Reservation	4
Reservation with sharing	4
Total EINQ	**31**

2.1.6 Total FP number

Internal Logical Files	49
External Interface Files	5
External Input	39
External Output	8
External Inquiries	31
Total FP	**132**

2.2 Effort and Duration - COCOMO Approach

We integrate FP results with cocomo approach to calculate effort estimation and how many people are needed to reach the goal of the project.

To switch between FP and SLOC (Source Line Of Code) we use factor 46 due to the use of JavaEE language
(http://www.qsm.com/resources/function-point-languages-table)

$SLOC = FPtotal * 46 = 132 * 46 = 6072\ SLOC$

Now we make a first not accurate estimation of the effort and the people needed, then by using a dedicated tool we will be more detailed.

Considering a project with all "Nominal" Cost Drivers and Scale Drivers would have an EAF of 1.00 and exponent E of 1.0997. So, considering EAF like the Effort Adjustment Factor derived from Cost Drivers and E like the Exponent derived from Scale Drivers:

$EFFORT = 2.94 * EAF * (KSLOC)^E = 2.94 * 1.00 * (6.072)^{1.0977} = 21.4\ Person/Month$

Now, we calculate the time duration of the project with this formula

$DURATION = 3.67 * (EFFORT)^E$

Considering an exponent E of 0.3179:

$DURATION = 3.67 * (21.4)^{0.3179} = 9.72\ Months$

And finally we can estimate the people needed for the developing of the project:

$N = EFFORT/DURATION = 21.4/9.71 = 2.25\ people \approx 3\ people$

Now, with a dedicated tool we make an accurate analysis. The result of the accurate analisys is not so far from the not accurate one. All the parameter inserted and the result is show below:

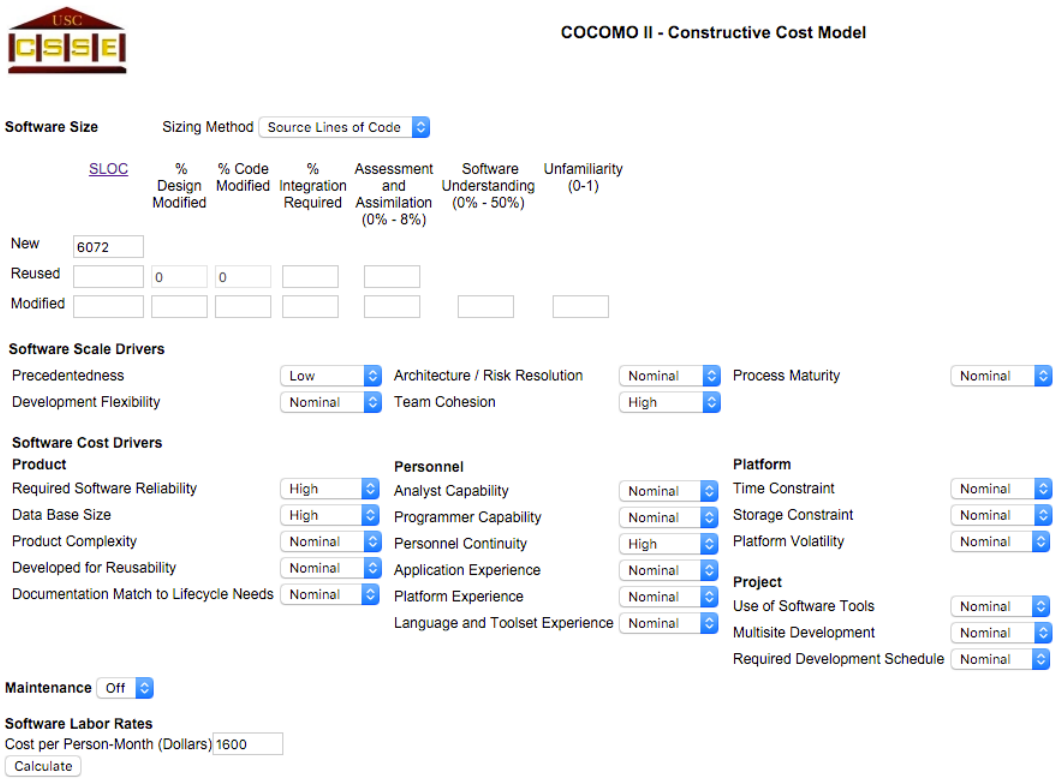


Figure 4: COCOMO web tool page 1

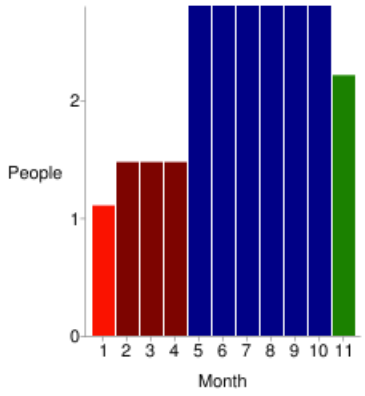
Results

Software Development (Elaboration and Construction)

Effort = 24.2 Person-months
Schedule = 10.5 Months
Cost = $38684

Total Equivalent Size = 6072 SLOC

Staffing Profile



Acquisition Phase Distribution

Phase	Effort (Person-months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	1.5	1.3	1.1	$2321
Elaboration	5.8	3.9	1.5	$9284
Construction	18.4	6.6	2.8	$29400
Transition	2.9	1.3	2.2	$4642

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

Phase/Activity	Inception	Elaboration	Construction	Transition
Management	0.2	0.7	1.8	0.4
Environment/CM	0.1	0.5	0.9	0.1
Requirements	0.6	1.0	1.5	0.1
Design	0.3	2.1	2.9	0.1
Implementation	0.1	0.8	6.2	0.6
Assessment	0.1	0.6	4.4	0.7
Deployment	0.0	0.2	0.6	0.9

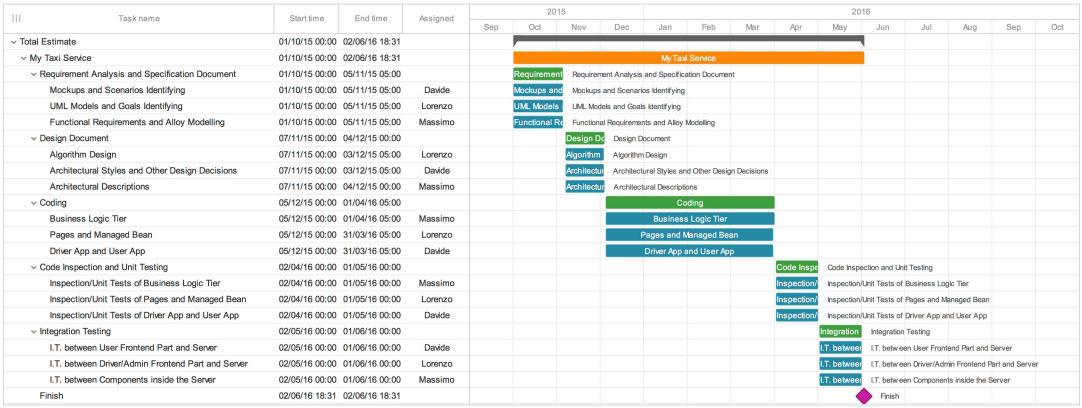
Figure 5: COCOMO web tool page 2

3 Tasks

The main task for our project are :

- The creation of the **Requirement Analysis and Specification Document**,which describe the system in terms of functional/non-functional requirements and analyse the real need of the customer to modelling the system;
- The creation of the **Design Document**, which will be used to aid in software development by providing the details for how the software should be built;
- The drafting of the **Integration Test Plan**;
- The preparation of the **Project Plan**, in order to guide both project execution and project control;
- The developement of the software,a.k.a. **Coding**;
- The **Code Inspection** and the **Unit Testing**;
- The **Integration Testing**;

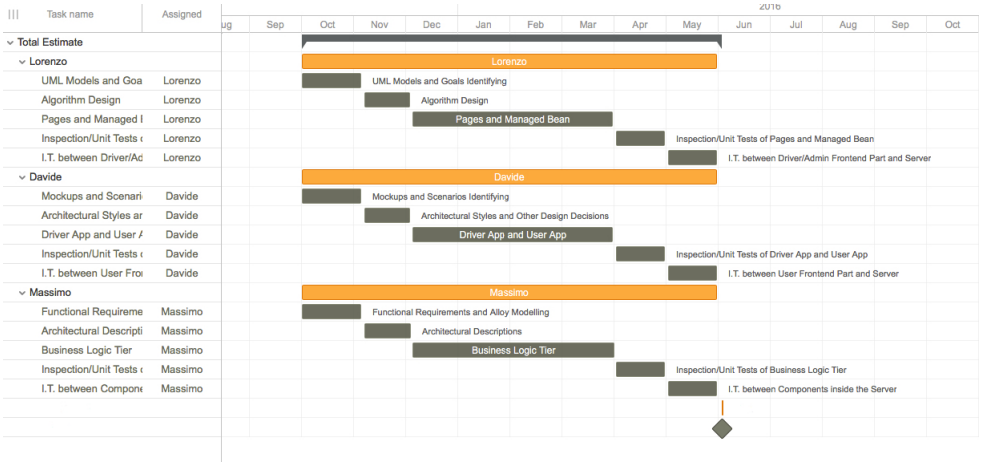
Below there is a Gantt chart represent the schedule of the main tasks of our project:



The **Integration Test Plan** and the **Project Plan** are drafting before and during the Coding phase:are not indicated upon because don't cover a large part of work and also because they change continously.

4 Resource allocation

Below there is a representation of the estimated work assigned to each member of the group:



We suppose that a student like us works 25-30 hours a week. Accordingly, each component of the group works 100-120 hours/month.

The total amount of hours of work is obtained by (100-120)hours * 3 student * 8 months: is equal to 2400 - 2880 hours. Instead, the duration of Cocomo II is 21.4 person/month * 152 hours: is equal to 3252.8 hours. We can obviously notice that the estimation of COCOMO II is oversized respect to the real time we suppose to spend for the project. This could be linked to the statistic nature of COCOMO II.

Inspite of all, the time we suppose to spend for the project differs from the COCOMO II estimation more or less by 20%: this make our project plan acceptable.

5 Risk Management

In the following subsections we have identified the possible risks of our project, we have classified them by probability and impact and we have defined the recovery actions to be executed in the eventuality these risks will become true.

5.1 Risk 1

The first risk we have identified is a Project risk. In the first phase of the project developing (the RASD delivery), the mockups production (assigned to Davide) and most of the UML diagrams writing (assigned to Lorenzo) depend on the functional requirements specification (assigned to Massimo). In other words, to write the UML diagrams e the mockups it is first needed to have specified the functional requirements. So, if Massimo becomes unavailable for a period of time before having completed the funcional requirements specification, the activities of Lorenzo and Davide will be blocked.

Risk Probability: Moderate

Risk Impact: Serious

Contingency plan: in the case of Massimo unavailability, the functional requirements specification will be assigned to Davide or Lorenzo and some of their work will be carried out by Massimo, when he will be available again.

5.2 Risk 2

The second risk we have identified is a Technical risk. Our system will be used in a big city, virtually with many users. All the operations that an user, a driver or an administrator can do implies a query (read only) or a transaction (read and write) on the data. So, since we will buy a database cloud service from a datacenter, this service may be undersized for our purpose.

Risk Probability: Moderate

Risk Impact: Serious

Contingency plan: at the end of the system developing, we will test the system simulating a huge number of "parallel" operations. We will evaluate the performance of the cloud system and we will eventually consider to buy a service with more powerful properties.

6 Work time

To produce this document, each member have worked around 5 hours.