Software Engineering 2

Project Plan

Authors
Paolo Paterna, Lara Premi

Politecnico di Milano, Italy
Email: paolo.paterna@mail.polimi.it, lara.premi@mail.polimi.it

2nd Feb 2016

Contents

1 Introduction

1.1 Purpose

This document is the Project Plan of a system, called myTaxiService, used to manage a taxi service in a city. The main goal of this document is to estimate the overall cost and define the risk of the project and system.

1.2 Scope

The scope of this system is to manage taxis in a city. A town is divided in zone of 2 square kilometers and, for each zone, the system defines a queue, composed by the identifier, which is the vehicle plate, of free taxis in that specific zone. A user can require a taxi ride from a zone, but he/she can also book one for another moment, using the web application or the mobile one. Furthermore, about long-term reservations, after the user has created one, he/she is able to modify the date or the hour or both of his/her booking, and he/she can delete it.

1.3 Acronyms and abbreviations

1.3.1 Acronyms

- FP: Function Points;
- ILF: Internal Logic File;
- EIF: External Interface File;
- EO: External Output;
- UFP: Unadjusted Function Points;
- JEE: Java Enterprise Edition.

1.3.2 Abbreviations

- EInput: External Input;
- EInquiry: External Inquiry;
- SLOC: Source Line of Code;
- E: Exponent derived from Scale Drivers;

- EAF: Effort Adjustment Factor derived from Cost Drivers;
- PREC: Precedentedness;
- FLEX: Development Flexibility;
- RESL: Risk Resolution;
- TEAM: Team Cohesion;
- PMAT: Process Maturity;
- RELY: Required Software Reliability;
- DATA: Database Size;
- CPLX: Product Complexity;
- RUSE: Developed for Reusability;
- DOCU: Documentation Match to Life-Cycle Needs;
- TIME: Execution Time Constraint;
- STOR: Main Storage Constraint;
- PVOL: Platform Volatility;
- ACAP: Analyst Capability;
- PCAP: Programmer Capability;
- PCON: Personnel Continuity;
- APEX: Application Experience;
- PLEX: Platform Experience;
- LTEX: Language and Tool Experience;
- TOOL: Use of Software Tools;
- SITE: Multisite Development;
- SCED: Required Development Schedule.

1.4 Reference Documents

To redact this document, we used the following other documents as references:

- Design Document from previous delivery
- *Cost Estimation.pptx* slides from BeeP site

2 Function Points

The Functional Point approach is an algorithmic technique, defined in 1975 by Allan Albrecht, that allows to evaluate the effort needed to develop a project. The functionalities list has been obtained from the RASD and the Design Document evaluating every specific function. The functionalities have been groped in:

- **Internal Logic Files:** homogeneous set of data used and managed by the application;
- **External Interface Files:** homogeneous set of data used by the application but generated and maintained by other applications;
- **External Input:** elementary operation to elaborate data coming from the external environment;
- **External Output:** elementary operation that generates data for the external environment[1];
- **External Inquiry:** elementary operation that involves input and output[2].

The following table specifies the function points values we have used:

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

Based on the specified combination of program characteristics, a weight is associated with each of these FP counts; the total is computed by multiplying each raw count by the weight and summing all partial values.

$$\sum \#ElementsOfGivenType * Weight$$

2.1 Internal Logic Files

In the system, there are different kind of ILFs. For the booking history, bookings info and taxi list we have identified a simple weight complexity;

[1] It usually includes the elaboration of data from logic files

[2] Without significant elaboration of data from logic files

the queue management, instead, has a complex weight since has to manage a high number of taxis, zones and queues.

$$FP_{ILF} = 7 * 3 + 15 * 1 = 36$$

2.2 External Interface Files

In the system, there are only one EIF which is the location management. It is a simple complex weight since it only has to retrieve information from the taxis GPSs. We can consider this as a simple complexity.

$$FP_{EIF} = 5 * 1 = 5$$

2.3 External Input

There are several external inputs in the system. On the side of the user, there are the bookings registration, long-term reservations modification and long-term reservations deletion. On the side of the taxi, there are the login/logout and the registration. All the inputs can be considered as simple since they do not require much data manipulation.

$$FP_{EInput} = 3 * 5 = 15$$

2.4 External Output

The only external output in the system are the different types of notification. We consider them as medium complexity since there are different types of notification.

$$FP_{EO} = 5 * 1 = 5$$

2.5 External Inquiry

There are several external inquiries in the system. On the user side, there are the company information and the long-term reservations information. On the taxi side, there are the rides information and the current ride information. We consider the user long-term reservations information and the taxi ride history as medium complexity; instead, we consider the company information and the current ride information as simple.

$$FP_{EInquiry} = 3 * 1 + 4 * 2 = 11$$

2.6 Summing Up

Summing all the partial results of the previous sections, we obtain the Unadjusted Function Point.

$$UFP = FP_{ILF} + FP_{EIF} + FP_{EInput} + FP_{EO} + FP_{EInquiry}$$

$$UFP = 36 + 5 + 15 + 5 + 11 = 72$$

3 COCOMO Approach

3.1 Manual Calculation

Considering to use JEE as programming language, to pass from FP to SLOC, we can use the average conversion factor 46, as described at the site `http://www.qsm.com/resources/function-point-languages-table`.

$$SLOC = UFP * ConvertionFactor$$

$$SLOC = 72 * 46 = 3312$$

According to the table below, we have considered the underlined values in yellow for the scale drivers and the underlined values in green for the cost drivers.

Table 62. COCOMO II 2000 Calibrated Post-Architecture Model Values

Baseline Effort Constants: **A** = 2.94; **B** = 0.91
Baseline Schedule Constants: **C** = 3.67; **D** = 0.28

Driver	Symbol	VL	L	N	H	VH	XH
PREC	SF_1	6.20	4.96	3.72	2.48	1.24	0.00
FLEX	SF_2	5.07	4.05	3.04	2.03	1.01	0.00
RESL	SF_3	7.07	5.65	4.24	2.83	1.41	0.00
TEAM	SF_4	5.48	4.38	3.29	2.19	1.10	0.00
PMAT	SF_5	7.80	6.24	4.68	3.12	1.56	0.00
RELY	EM_1	0.82	0.92	1.00	1.10	1.26	
DATA	EM_2		0.90	1.00	1.14	1.28	
CPLX	EM_3	0.73	0.87	1.00	1.17	1.34	1.74
RUSE	EM_4		0.95	1.00	1.07	1.15	1.24
DOCU	EM_5	0.81	0.91	1.00	1.11	1.23	
TIME	EM_6			1.00	1.11	1.29	1.63
STOR	EM_7			1.00	1.05	1.17	1.46
PVOL	EM_8		0.87	1.00	1.15	1.30	
ACAP	EM_9	1.42	1.19	1.00	0.85	0.71	
PCAP	EM_{10}	1.34	1.15	1.00	0.88	0.76	
PCON	EM_{11}	1.29	1.12	1.00	0.90	0.81	
APEX	EM_{12}	1.22	1.10	1.00	0.88	0.81	
PLEX	EM_{13}	1.19	1.09	1.00	0.91	0.85	
LTEX	EM_{14}	1.20	1.09	1.00	0.91	0.84	
TOOL	EM_{15}	1.17	1.09	1.00	0.90	0.78	
SITE	EM_{16}	1.22	1.09	1.00	0.93	0.86	0.80
SCED	EM_{17}	1.43	1.14	1.00	1.00	1.00	

Considering only the scale drivers, we make the following addition:

$$E = B + 0.01 * (PREC + FLEX + RESL + TEAM + PMAT)$$

where B = 0.91 for COCOMO-II

$$E = 0.91 + 0.01 * (3.72 + 2.03 + 4.24 + 3.29 + 4.68) = 1.0896$$

Considering only the cost drivers, we make the following multiplication:

$$EAF = RELY*DATA*CPLX*RUSE*DOCU*TIME*STOR*PVOL* \\ *ACAP*PCAP*PCON*APEX*PLEX*LTEX*TOOL*SITE*SCED$$

$$EAF = 1.10 * 1.00 * 1.00 * 1.00 * 1.00 * 1.00 * 1.00 * 1.00 * *1.00 * 1.00 * \\ 1.00 * 1.00 * 1.00 * 1.00 * 1.00 * 1.00 * 1.00 = 1.10$$

So, now we can compute the effort, using the formula below:

$$effort = 2.94 * EAF * (KSLOC)^E$$

$$effort = 2.94 * 1.10 * (3.312)^1.0896 = 11.92 \text{ person/months}$$

Now we calculate the duration of the project in months, using the following formula:

$$duration = C * (effort)^{(D+0.2*(E-B))}$$

where C = 3.67, D = 0.28, B = 0.91 for COCOMO-II

$$duration = 3.67 * (11.92)^(0.31592) = 8.03 \text{ months}$$

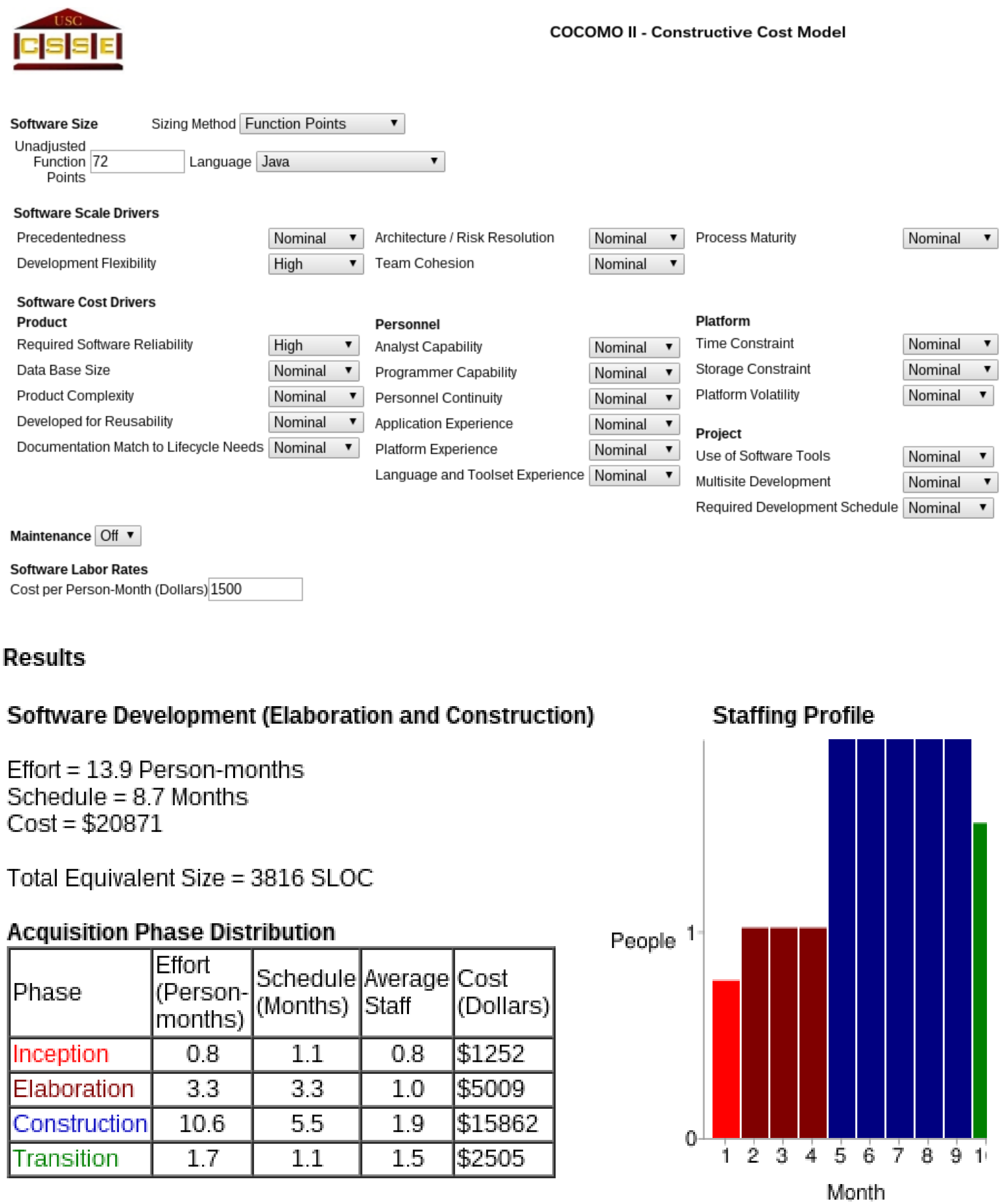
The people needed to complete the project are calculated through the formula below:

$$N_{people} = \lceil effort/duration \rceil$$

$$N_{people} = \lceil 11.92/8.03 \rceil = \lceil 1.48 \rceil = 2$$

3.2 Software Calculation

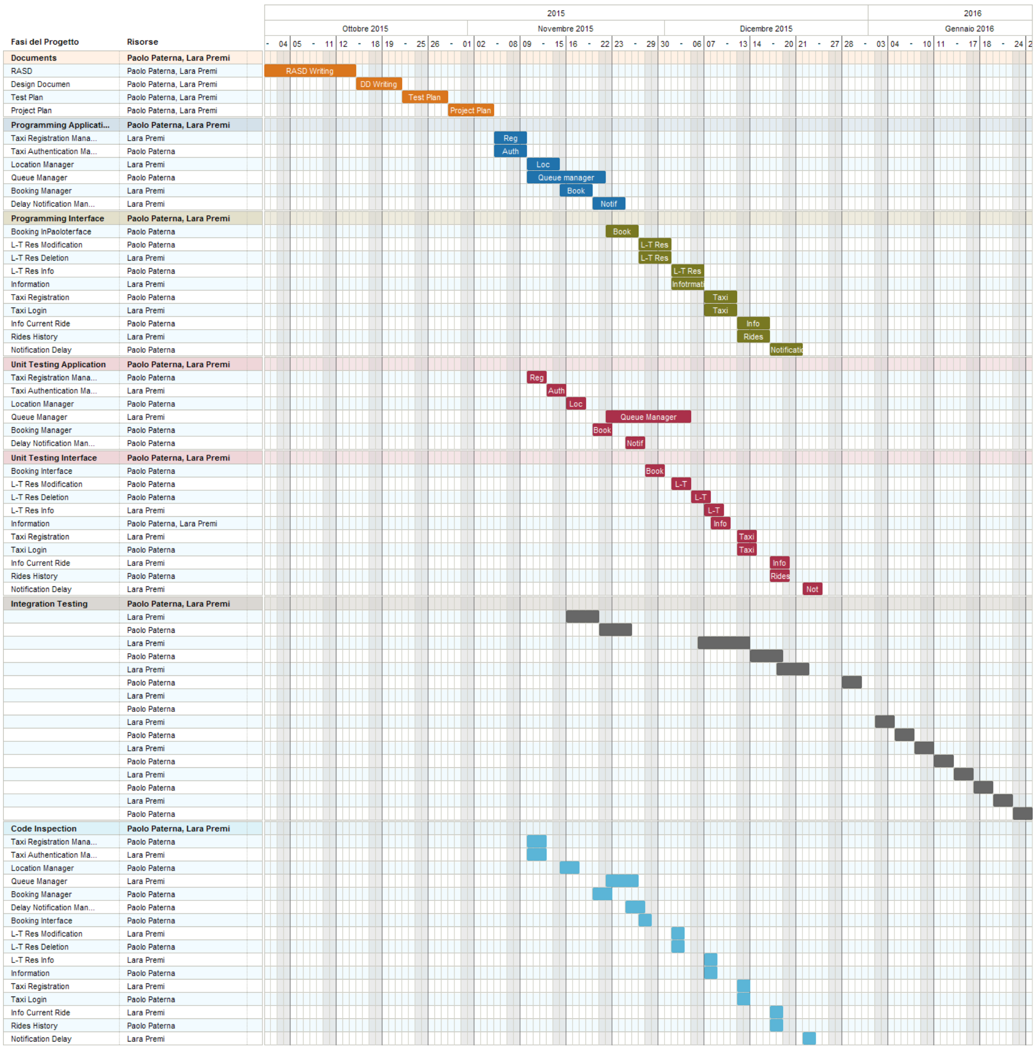
Using the online tool `http://csse.usc.edu/tools/COCOMOII.php`, we give a more accurate estimation of the previous calculus. In the following images, we can see the results.



Phase	Effort (Person-months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	0.8	1.1	0.8	$1252
Elaboration	3.3	3.3	1.0	$5009
Construction	10.6	5.5	1.9	$15862
Transition	1.7	1.1	1.5	$2505

4 Gantt chart

We identify the tasks of our project and their schedule, using the Gantt chart below.



5 Risks

We have considered two risks. First of all, it could happen that the project staff is ill at critical time, during the work: this risk is characterized by a moderate probability to occur and it can have serious effects. To solve it, the team will be reorganize, impling that there will be more overlap of work and people therefore will understand each other's jobs. The second risk on which we have focused on is that the database used in the system cannot process as many transactions per second as expected: this risk is characterized by a moderate probability to happen and it can have serious effects, like the first risk that we have said before. In this case, the staff will investigate the possibility of buying a higher-performance database.

6 Hours of Work

In order to write this document, we have done the following hours of work:

- Paolo Paterna: 8 Hrs;
- Lara Premi: 8 Hrs.