Software Engineering 2 project - Cost Estimation and Planning Document

Gabriele Giossi (id:854216)

v1.0 29/01/2016

Contents

1 Introduction **3**
- 1.1 Revision History . . . 3
- 1.2 Document Description . . . 3
- 1.3 Problem Description . . . 3

2 Function Points **4**
- 2.1 General Introduction . . . 4
- 2.2 Function Points calculation . . . 4
  - 2.2.1 EIF . . . 4
  - 2.2.2 ILF . . . 4
  - 2.2.3 Inputs . . . 5
  - 2.2.4 Inquiries . . . 5
  - 2.2.5 Outputs . . . 5
- 2.3 Final FP calculation . . . 6

3 COCOMO Cost Estimation **6**
- 3.1 Estimation 1: All nominal drivers . . . 7
- 3.2 Estimation 2: Drivers modified to match real team characteristics and experience . . . 9

4 Resource Allocation **11**

5 Risk definition **13**
- 5.1 Project risks . . . 13
- 5.2 Technical risks . . . 13
- 5.3 Business risks . . . 14

6 Appendix **15**
- 6.1 Working hours in fulfillment of the deadline . . . 15
- 6.2 Working hours for the whole project summarization . . . 15

1 Introduction

1.1 Revision History

v1.0: first release 29/01/2016;

1.2 Document Description

This Planning and cost estimation document for the MyTaxiService project is focused on describing and planning all development phases of the MyTaxiService application - it makes use of methods such as Function points, COCOMO, and risk calculation to plan all the development stages, calculate their costs and needed time, and personnel allocation.

1.3 Problem Description

MyTaxiService will be a web-based application, supported both on smartphones and PCs; its aim is, as said, optimizing the taxi service of a city: the system divides the city in which it operates into "zones" of about 2 square kilometers, and to each of them computes a taxi queue

When a passenger, by means of the application, requests a taxi ride, the system assigns an available taxi, the first in queue near the starting point of the ride; the taxi driver will confirm a ride before actually carrying it out.

Some of the benefits of implementing the MyTaxiService system are that it is possible to achieve a real-time balance of numbers of rides requested in a given zone with respect to the available taxis in a zone, providing taxi drivers with an almost-real-time response to spikes of ride requests, improving service quality while simplyfing the interaction of requesting a taxi ride and finding a taxi for any passenger, wherever in the city he might be.

2 Function Points

2.1 General Introduction

The Function points approach requires the following steps:

1. Evaluate ILF (Internal Logic Files)
2. Evaluate EIF (External Interface Files)
3. Evaluate EI (External Inputs)
4. Evaluate EIQ (External Inquiries)
5. Evaluate EO (External Outputs)
6. Compute UFP (Unadjusted FPs)

With the following weighting:

Function Types	Weights		
	Simple	Medium	Complex
Inputs	3	4	6
Outputs	4	5	7
Inquiry	3	4	6
ILF	7	10	15
EIF	5	7	10

2.2 Function Points calculation

2.2.1 EIF

The application interacts with external network services such as the GPS system to query for location data: using a somewhat pessimistic approach, thus weighting it at least as a medium complexity interaction, this leads to $FP = 1 \times 7 = 7FPs$

2.2.2 ILF

The application stores information about:

- Registered users data;
- Taxi data;
- GPS data;
- Request data;
- City map data and divison in zones

Thefirst four entities can be considered simple, as they are composed of a handful of fields of simple data (numbers, strings, and such); thus, they can all be considered simple data, thus for them $FP = 7 \times 4 = 28FPs$.

Moreover, the citymap data and its divison in zones is possibly a large and complex data type, thus we have to give it a complex weight: $FP = 1 \times 15 = 15FPs$

In total, the ILF section calculations for Function Points sums up to $FP = 28 + 15 = 43FPs$

2.2.3 Inputs

The application interacts with the users as follows:

- login/logout: these are two simple operations: thus, we can consider them with a simple weight: $FP = 2 \times 3 = 6FPs$
- Queuing operation: this, again, is a simple operation, and can again be weighted with a simple weight: $FP = 1 \times 3 = 3FPs$
- Request handling: this operation set (request making, request assigning, acceptance, closing) are singularly very simple, and similar; we can consider them all simple weighted: $FP = 4 \times 3 = 12FPs$
- GPS data exchange: judging by the amount of data exchanged and the GPS system complexity, we consider this to have a complex weight: $FP = 1 \times 6 = 6FPs$

In total, we have $FP = 6 + 3 + 12 + 6 = 27FPs$

2.2.4 Inquiries

The application allows users (including administrator level users) to inquire for:

- Diagnostics data
- Account data
- Statistics data

All in all, these are all operations that can be considered of medium complexity, thus: $FP = 3 \times 4 = 12FPs$

2.2.5 Outputs

The system creates notifications to client applications that are to be sent through to clients: this is a complex operation, as it needs to create a notification, compute the client to send it to, and send it; thus we can give it a complex weight: $FP = 1 \times 7 = 7FPs$

2.3 Final FP calculation

Summarizing the previous data about Function Points that have been calculated, the following table is created:

Function Types	Function Points
Inputs	27
Outputs	7
Inquiries	12
ILF	43
EIF	7
TOTAL	96

It has been decided that the function points will NOT be adjusted, but will be left as-is;

In order to give a better explanation of this result, we have exploited the table that relates lines of code for a specific language to the number of function points that we have derived - the resource is located at `http://www.qsm.com/resources/function-point-languages-table` - the following table shows the projected lines of code with this Function Points analysis as baselin in various languages (the value taken is the median one, that we think shows a good estimation of a projected result):

Language	Approximate LOC (Lines of Code)
Java	5,088
J2EE	4,704

3 COCOMO Cost Estimation

The COCOMO approach has been calculated using the approximate Source Lines of Code method, by inputting the approximate median Lines od Code amount calculated in the previous section with Function points, using the J2EE Language; the tool used is available at `http://csse.usc.edu/tools/COCOMOII.php`

3.1 Estimation 1: All nominal drivers

COCOMO II - Constructive Cost Model

Software Size Sizing Method: Source Lines of Code

	SLOC	% Design Modified	% Code Modified	% Integration Required	Assessment and Assimilation (0% - 8%)	Software Understanding (0% - 50%)	Unfamiliarity (0-1)
New	4704						
Reused		0	0				
Modified							

Software Scale Drivers

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

Software Cost Drivers

Product		Personnel		Platform	
Required Software Reliability	Nominal	Analyst Capability	Nominal	Time Constraint	Nominal
Data Base Size	Nominal	Programmer Capability	Nominal	Storage Constraint	Nominal
Product Complexity	Nominal	Personnel Continuity	Nominal	Platform Volatility	Nominal
Developed for Reusability	Nominal	Application Experience	Nominal	**Project**	
Documentation Match to Lifecycle Needs	Nominal	Platform Experience	Nominal	Use of Software Tools	Nominal
		Language and Toolset Experience	Nominal	Multisite Development	Nominal
				Required Development Schedule	Nominal

Maintenance Off

Software Labor Rates

Cost per Person-Month (Dollars) 1500

Calculate

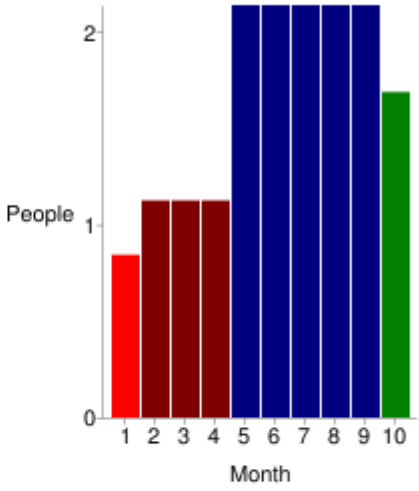
Results

Software Development (Elaboration and Construction)

Effort = 16.1 Person-months
Schedule = 9.2 Months
Cost = $24207

Total Equivalent Size = 4704 SLOC

Staffing Profile



Acquisition Phase Distribution

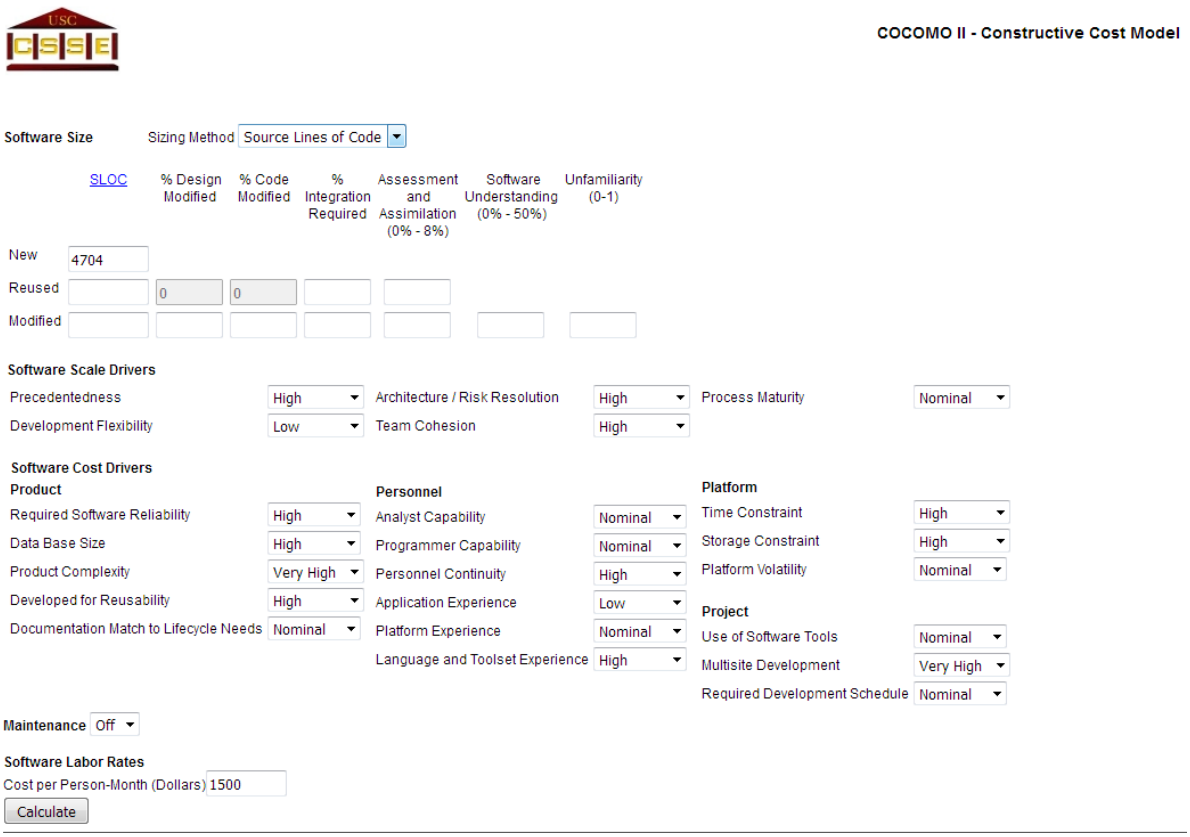
Phase	Effort (Person-months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	1.0	1.1	0.8	$1452
Elaboration	3.9	3.4	1.1	$5810
Construction	12.3	5.7	2.1	$18398
Transition	1.9	1.1	1.7	$2905

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

Phase/Activity	Inception	Elaboration	Construction	Transition
Management	0.1	0.5	1.2	0.3
Environment/CM	0.1	0.3	0.6	0.1
Requirements	0.4	0.7	1.0	0.1
Design	0.2	1.4	2.0	0.1
Implementation	0.1	0.5	4.2	0.4
Assessment	0.1	0.4	2.9	0.5
Deployment	0.0	0.1	0.4	0.6

This assessment supposes all drivers are set to nominal; it is the perfectly median situation.

3.2 Estimation 2: Drivers modified to match real team characteristics and experience



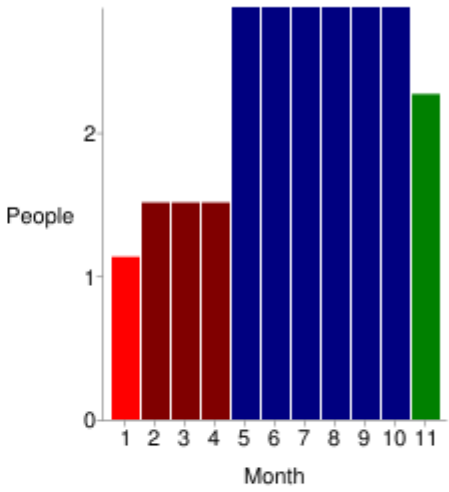
Results

Software Development (Elaboration and Construction)

Effort = 25.1 Person-months
Schedule = 10.6 Months
Cost = $37670

Total Equivalent Size = 4704 SLOC

Staffing Profile



Acquisition Phase Distribution

Phase	Effort (Person-months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	1.5	1.3	1.1	$2260
Elaboration	6.0	4.0	1.5	$9041
Construction	19.1	6.6	2.9	$28629
Transition	3.0	1.3	2.3	$4520

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

Phase/Activity	Inception	Elaboration	Construction	Transition
Management	0.2	0.7	1.9	0.4
Environment/CM	0.2	0.5	1.0	0.2
Requirements	0.6	1.1	1.5	0.1
Design	0.3	2.2	3.1	0.1
Implementation	0.1	0.8	6.5	0.6
Assessment	0.1	0.6	4.6	0.7
Deployment	0.0	0.2	0.6	0.9

This assessment takes into account drivers being set to match a possible real-life situation of a localized development team tasked with the project; costs are almost twice with respect to the all nominal drivers case, as is the Effort amount, but we see the schedule has not increased as much.

4 Resource Allocation

For the development of the application MyTaxiService, the development team Beta will be in charge; this team is composed of:

- Mr. Gabriele Giossi;
- Mr. Jack Jameson;
- Ms. Anne Smithers;

We suppose each member works 20 to 40 hours a week, given cases of having to devote time towards other work activities and/or study - in this case, due to being a full-time university student, mr: Giossi has 20 hours a week of availability, while Ms. Smithers and Mr. jameson have 40 hours a week availability as full-time employees. The total of working hours for all the team in one working day is 20 hours.

We also have to identify each task, which is done in the table below (The start date is October 1st, 2015):

Task ID	Task Description		Dependencies
T1	Requirements Analysis and RASD redaction		
T2	Design Phase and DD Redaction		
T3	Implementation in Code		T1, T2
T4	Testing (all levels of internal testing)		T3
T5	Review and redaction of documentation		T4
T6	Deployment		T5
T7	Client teaching		T6

Take note that his table is purely speculative and optimistic, as it assumes there are no setbacks or problems during development that might slow or halt the project; it is meant to show the road-map for the whole development project for MyTaxiService. Moreover, in the calculation of the needed working hours is included the possibility that during the carrying out of the tasks a need to review or modify any of the artefacts produced in the previous tasks could arise; moreover, festivities are already accounted for.

The resource (personnel) allocation to tasks is as follows:

1. October 2015
   (a) T1 carried out by Mr. Giossi
   (b) T2 carried out by Ms. Smithers and Mr. Jameson in the same time frame as T1;
   (c) Time allocated: one month
2. November 2015
   (a) T3 carried out by the entire team as soon as tasks T1 and T2 are completed - estimated time needed: 2 months;

3. December 2015
    (a) T3 allocated to all personnell for the whole month;
4. January 2016
    (a) T4 allocated to all personnel - estimated time: 2 weeks;
    (b) T5 allocated to all personnel after T4 is done - estimated time: 2 weeks;
5. February 2016
    (a) T6 carried out by Mr. Jameson and Mr. Giossi - estimated time: 1 week;
    (b) T7 carried out by Ms. Smithers and Mr. Giossi - time allocated: 2 weeks;

With this allocation, the project is supposed to be terminated in 4 months and a week, plus 2 weeks of client teaching - it is a very optimistic estimate, given the analysis done by COCOMO in the preivous section.

5 Risk definition

In this section will be presented possible risks and hazards that could arise during project development, their severity, relevance, and the recovery actions.

5.1 Project risks

1. Employee disagreements/animosity
   (a) Severity: Critical, might delay/jeopardize the whole project
   (b) Probability: Very Low
   (c) Prevention Actions: Foster group motivation, do not assign excessive work loads, create group spirit;
2. Company-wide issues (e.g. leadership changes, general internal turmoil)
   (a) Severity: Potentially critical
   (b) Probability: Very Low, we suppose a company with problems will not accept commissions for such a project
   (c) Prevention Actions: Enable the team to work outside of company environment and through internal channels
3. Impossibility to deliver artefacts within given deadlines
   (a) Severity: High
   (b) Probability: Medium to Low, it depends also un unexpected and unforeseeable factors
   (c) Prevention Actions: Notify stakeholders of any possible problems and schedule meetings to discuss project state

5.2 Technical risks

1. Loss of project data/documentation/code due to hardware failure
   (a) Severity: Very High
   (b) Probability: Low
   (c) Recovery Actions: Prevent the total loss of data; in order to do so, have multiple copies of everything related to the project stored in multiple local drives, in clouds and in fail-proof hardware.
2. Team technical competence in the project is discovered to be below expected, resulting in timeliness and code quality below needed standards
   (a) Severity: Potentially Critical
   (b) Probability: Medium (the assumption is based on past project experience, but the project itself is rather complex)
   (c) Reaction: In case of problems in technical competence, avoid rushing the project and instead notify stakeholders of possible issues.

5.3 Business risks

1. Stakeholders/Clients not interested in application
   (a) Severity: Devastating
   (b) Probability: Very low
   (c) Prevention: Maintain a high standard of quality in the artefacts to provide
2. Stakeholder changes project specifications during development
   (a) Severity:; Variable (Depends on the extension of the modification to the specification and its impact)
   (b) Probability: Medium
   (c) Prevention: Maintain communication with stakeholders during all stages of development and organize periodic meetings to discuss project status and possible modifications.

6 Appendix

6.1 Working hours in fulfillment of the deadline

Towards the release deadline for the document, group members worked the following hours:

Gabriele Giossi: 6 hours;

6.2 Working hours for the whole project summarization

Concerning the whole document redaction work, the total amount of work hours for each member follows:

- Gabriele Giossi:
  - RASD: 30 hours
  - DD: 20 hours
  - Testing Document: 12 hours
  - This document: 10 hours
  - Total: 72 hours