My Taxi:

Project Plan

Authors:

Giovanni Bucci

Riccardo De Togni

Summary

1. Introduction

This document aims to evaluate the effective effort needed to entirely develop MyTaxi project in all its features, and after that give an hypothesis on how to schedule the development. The evaluation starts with a Function Point Analysis, which will provide a rough estimate of the SLOC (Source Lines of Code). Then with SLOC value, we will proceed with a COCOMO II analysis to calculate correspondent Effort and Duration.

The second part of the document will explain the project schedule through tasks identification and allocation to team members.

2. Project Size and Cost evaluation

2.1. Function Point Analysis

In order to perform the function point analysis we identify all the features of the project such that Internal Logical Files, External Interface Files and External Inquiries, Inputs and Outputs. To evaluate the complexity and the correspondent function point amount of each functionality we referred to the COCOMO II Function Point Weight Tables available at:

http://csse.usc.edu/csse/research/COCOMOII/cocomo2000.0/CII_modelman2000.0.pdf

The first Table is used to evaluate the complexity:

Table 2. FP Counting Weights

For Internal Logical Files and External Interface Files

	Data Elements		
Record 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

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

For External Input

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

The second one, given a complexity returns a function point number.

Table 3. UFP Complexity Weights

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

The total amount of function points will represent the UFP (Unadjusted Function Points) converted in SLOC through a language-dependent factor. Since the lack of any implementation technique constraint all over the project documentation, we are free to adopt any language in the evaluation. For this evaluation, we will use Java Enterprise Edition, which has a converting factor of 46 SLOC/FP.

2.1.1. Internal Logical Files

According to the E-R Diagram [DD 5.1] the system stores data about Guests, Users, Administrators (a particular type of User), Taxi Drivers, Requests, Reservations and Queues. The functionality of these entities has been deeply analysed in the previous documents so we will just summarize the complexity in the following table:

Internal Logical Files	Complexity	Function Points
User	High	15
Guest	High	15
Taxi Driver	Average	10
Request	High	15
Reservation	Average	10
Queue	Low	7
Total:		72

2.1.2. External Interface Files

The system has an interface with three different external component: Localization System, Mail Server and SMS Server. The localization system given a GPS Signal provides a couple of coordinates but we expect an intense flow of data since the

number of user could be very high. The mail server and the SMS server manages the delivery of messages. The complexity is presented in the table below:

External Interface Files	Complexity	Function Points
Localization System	Average	7
Mail Server	Average	7
SMS Server	Average	7
Total:		21

2.1.3. External Inputs

The input are divided by the entity that perform it:

- Users can Login, Logout, Register to the system.
- Taxi Drivers can Login, Logout, Register, give availability, accept/deny request

Login and Logout inputs have been count just once although they are a functionality proper of both Users and Taxi Drivers. Registration has been differentiated because the two procedures involve different steps.

External Inputs	Complexity	Function Points
Login/Logout/Register	Low	3x3
Taxi Registration	Average	4
Accept/Deny Request	Low	3
Give Availability	Low	3
Total:		19

2.1.4. External Inquiries

As we did for Inputs we will divide Inquiries by entity:

- Inquiries that involve users: Create request/Reservation, manage personal data
- Inquiries that involve taxi drivers: Manage Personal Data

External Inquiries	Complexity	Function Points
Create request/reservation	Average	4x2
Manage Personal Data	Low	3

Manage Personal Data	Low	3
Total:		14

2.1.5. External Outputs

The application alert the taxi driver with an incoming Request, and alert a user when its reservation has forwarded a request to the system (i.e. the reservation time has come).

External Output	Complexity	Function Points
Taxi Alert	Low	4
User Alert	Low	4
Total:		8

2.1.6. Unadjusted Function Points

Now we proceed with the evaluation of UFP:

$$UFP = 72 + 21 + 19 + 14 + 8 = 134$$

Therefore, the number of Source Lines of code will be:

$$SLOC = 134 * 46 = 6164$$

2.2. COCOMO II Analysis

Once estimated the size of the system-to-be, it is possible to make a first prevision of the software cost in terms of time spent and people allocated to the project.

The COCOMO (Constructive Cost Model) approach is based on effort and duration estimation using ad-hoc formulae that consider many parameters derived from previous projects data and future previsions.

The formula used for effort calculation is the following:

$$\mathrm{PM} = \mathrm{A} \times \mathrm{Size}^{\mathrm{E}} \times \prod_{i=1}^{n} \mathrm{EM}_i$$

$$\text{where } \mathrm{A} = 2.94 \text{ (for COCOMO II.2000)}$$

The effort is calculated in Persons-Month; in this case, the parameter Size is derived from the Function Points evaluation done before, E and EM are factors derived from respectively

scale factors and cost drivers, i.e. elements that let the project manager consider the system necessities and have a preview of what the team needs to deal with.

The exponent E is obtained from the following expression:

$$E = B + 0.01 \times \sum_{j=1}^{5} SF_j$$

$$\text{where } B = 0.91 \text{ (for COCOMO II.2000)}$$

SF are the mentioned scale factors, consider elements like developers experience, team cohesion, and project specifications.

Their value is decided with the help of the dedicated table.

Let's analyse them in detail:

Precedenceness:

It reflects the previous experiences related to this kind of projects. In this case the team had already developed similar systems so the nominal value will be reflecting the actual situation.

Development flexibility:

It reflects the flexibility of costraints in the development process. The stakeholders set precise specifications but without letting the development team free to choose the majority of implementation details, for this reason this value will be nominal.

Risk resolution:

Reflects the extent of risk analysis. A well developed risk management plan corresponds to a high value in the table. In this case the value considered is nominal.

Team cohesion:

Reflects how the development team know each other and cooperate. In this case the team is united; people communicate and cooperate in an efficient way, so it is possible to consider a high value for this parameter.

Process maturity:

Reflects team maturity regarding project development management. Organization and adopted techniques influence this factor. For this project, the correct value is the nominal one since the project is developed under standard conditions.

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

In this particular project the values considered are often the ones in the “Nominal” column, since the project conditions are standard and often idealized. The only higher value is the one regarding Team Cohesion.

With the chosen factors, we can calculate the exponent E with the already presented formula.

$E = 0.91 + 0.01 \times (3.72 + 3.04 + 4.24 + 2.19 + 4.68) = 0.91 + 0.01 \times 17.87 = 0.91 + 0.1787 \approx 1.08$

Now it is necessary to calculate effort multipliers, and it is done in the same way used for scale factors, using the dedicated tables.

RELY Descriptors:	slight inconven-ience	low, easily recoverable losses	moderate, easily recoverable losses	high financial loss	risk to human life	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	0.82	0.92	1.00	1.10	1.26	n/a

DATA* Descriptors		Testing DB bytes/Pgm SLOC < 10	10 ≤ D/P < 100	100 ≤ D/P < 1000	D/P ≥ 1000	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	n/a	0.90	1.00	1.14	1.28	n/a

Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	0.73	0.87	1.00	1.17	1.34	1.74

RUSE Descriptors:		none	across project	across program	across product line	across multiple product lines
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	n/a	0.95	1.00	1.07	1.15	1.24

DOCU Descriptors:	Many life-cycle needs uncovered	Some life-cycle needs uncovered.	Right-sized to life-cycle needs	Excessive for life-cycle needs	Very excessive for life-cycle needs	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	0.81	0.91	1.00	1.11	1.23	n/a

TIME Descriptors:			≤ 50% use of available execution time	70% use of available execution time	85% use of available execution time	95% use of available execution time
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	n/a	n/a	1.00	1.11	1.29	1.63

STOR Descriptors:			≤ 50% use of available storage	70% use of available storage	85% use of available storage	95% use of available storage
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	n/a	n/a	1.00	1.05	1.17	1.46

PVOL Descriptors:		Major change every 12 mo.; Minor change every 1 mo.	Major: 6 mo.; Minor: 2 wk.	Major: 2 mo.;Minor: 1 wk.	Major: 2 wk.;Minor: 2 days	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	n/a	0.87	1.00	1.15	1.30	n/a

ACAP Descriptors:	15th percentile	35th percentile	55th percentile	75th percentile	90th percentile	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	1.42	1.19	1.00	0.85	0.71	n/a

PCAP Descriptors	15th percentile	35th percentile	55th percentile	75th percentile	90th percentile	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	1.34	1.15	1.00	0.88	0.76	n/a

PCON Descriptors:	48% / year	24% / year	12% / year	6% / year	3% / year	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	1.29	1.12	1.00	0.90	0.81	

APEX Descriptors:	≤ 2 months	6 months	1 year	3 years	6 years	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	1.22	1.10	1.00	0.88	0.81	n/a

PLEX Descriptors:	≤ 2 months	6 months	1 year	3 years	6 year	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	1.19	1.09	1.00	0.91	0.85	n/a

LTEX Descriptors:	≤ 2 months	6 months	1 year	3 years	6 year	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	1.20	1.09	1.00	0.91	0.84	

TOOL Descriptors	edit, code, debug	simple, frontend, backend CASE, little integration	basic life-cycle tools, moderately integrated	strong, mature life-cycle tools, moderately integrated	strong, mature, proactive life-cycle tools, well integrated with processes, methods, reuse	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	1.17	1.09	1.00	0.90	0.78	n/a

SITE: Collocation Descriptors:	Inter-national	Multi-city and Multi-company	Multi-city or Multi-company	Same city or metro. area	Same building or complex	Fully collocated
SITE: Communications Descriptors:	Some phone, mail	Individual phone, FAX	Narrow band email	Wideband electronic communication.	Wideband elect. comm., occasional video conf.	Interactive multimedia
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	1.22	1.09	1.00	0.93	0.86	0.80

SCED Descriptors	75% of nominal	85% of nominal	100% of nominal	130% of nominal	160% of nominal	
Rating Level	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multiplier	1.43	1.14	1.00	1.00	1.00	n/a

RELY: The measure of how much reliable the software must be. Since this is a taxi management application, the malfunctioning of the system will cause an easily recoverable loss, so the chosen value is the low one.

DATA: This measures the effects of data dimensions on project development. It represents the effort needed to assemble and maintain the required data. Since the amount of data is acceptable, the nominal value is considered for this multiplier.

CPLX: This factor represents the required complexity of the system, in terms of operation, code, data management. The product needs to interact with external components and with a notification system, so it is considered of nominal complexity.

RUSE: This cost driver represents the additional effort needed to project thinking about reusing components on current or future projects. The intention here is to have reusable components inside the system, but without the necessity to make them available for future products, so the chosen value is nominal.

DOCU: The level of required documentation. Since standard documentation is requested, the considered value is nominal.

TIME: Measure of the execution time constraints imposed upon the system. No particular constraints are imposed upon this project, o the value chosen is the nominal one.

STOR: Measure of data occupation constraint imposed upon the system. No particular constraints are imposed in this case, the chosen value is nominal.

PVOL: Measure of the necessary changes ratio on the system to keep it up-to-date with platform and functions. Not many major modifications are expected, so it is fine to consider a nominal value for this cost driver.

ACAP: The capability of the analysts who work on high-level design. The value is decided basing on the percentile in which they fall. In this case this is a supposed nominal value since the analysts team is not present.

PCAP: The programmers ability to deal with new technologies. This value do not consider the programmers experience. It is measured considering the percentile in which the programmers fall. It is assumed as nominal for the current project.

PCON: Represents the personnel continuity. Since the project team remains unchanged for the whole project duration, it is possible to consider a high value for this driver.

APEX: This factor considers the programmers team experience in developing the requested kind of system. Since the project is of standard complexity, with already studied technologies, it is fair to assume a high value for this field.

PLEX: The developer team experience regarding the importance of platforms. It is possible to consider a nominal value, reflecting the team actual experience.

LTEX: Measure of the level of programming language and tool knowledge. Considering previous experiences, the team level can be rated as high.

TOOL: The usage of software tools to code, edit or management. This project utilized only the basic tools, so the cost driver can be considered as nominal.

SITE: Multisite development factor. The team is fully collocated, so it is possible to assume an extra high value for this effort multiplier.

SCED: This rating measures the schedule constraints imposed on the project team. In this particular case, the schedule varied from 100% to 130% due to deadlines, so it is fair to consider the nominal/high value.

Once the cost drivers are decided, it is possible to calculate the product of the effort multipliers, and so eventually obtain the effort.

EAF (Effort Adjustment Factor) = $\prod$ EMi

EAF = 0.92 x 1 x 1 x 1 x 1 x 1 x 1 x 1 x 1 x 1 x 0.90 x 0.88 x 1 x 0.91 x 1 x 0.83 x 1 = **0.55**

Effort is then calculated by the already presented formula:

PM = A x EAF x (Size)^E = 2.94 x 0.55 x (6.164) ^ 1.08 = 2.94 x 0.55 x 7.13 = **11.52**

The duration of the project is then estimated using the dedicated formula:

$$TDEV_{NS} = C \times (PM_{NS})^{F}$$

$$\text{where } F = D + 0.2 \times 0.01 \times \sum_{j=1}^{5} SF_j$$

$$= D + 0.2 \times (E - B)$$

Considering the following parameters

A = 2.94	B = 0.91
C = 3.67	D = 0.28

F = 0.28 + 0.2 x (1.08 – 0.91) = 0.28 + 0.2 x 0.17 = 0.28 + 0.034 = 0.314

TDEV = 3.67 x (11.52) ^ 0.314 = 3.67 x 2.15 ≈ **8 months**

And finally the number of people allocated (obtained as Effort / Duration) matches the actual availability, in fact:

Number of People = PM / TDEV = 11.8 / 8 ≈ **2 people**

3. Project Scheduling

3.1. Tasks Identification

The aim of this paragraph is to highlight the main tasks of the project development. These tasks are just a guideline since the project could be subject of modification or introduction of new requirements and functionalities.

There are the main tasks:

[T1]: Write and deliver Requirements Analysis and Specification Document (RASD)

[T2]: Write and deliver Design Document

[T3]: Write and deliver Integration Test Plan

[T4]: Write and deliver Project Plan

[T5]: Project Implementation

[T6]: Unit Test

[T7]: Integration Test

[T8]: Deliver and test a Beta Release

[T9]: Final Release

In the table below, we present the task interdependencies:

Task	Start	Deadline	Dependencies
T1	15/10/2015	6/11/2015	//
T2	12/11/2015	4/12/2015	T1
T3	6/12/2015	3/01/2016	T2

T4	3/01/2016	10/01/2016	T2,T1
T5	10/01/2016	3/05/2016	T2,T4
T6	3/05/2016	10/05/2016	T5,T4
T7	10/05/2016	20/05/2016	T5,T6
T8	20/05/2016	30/05/2016	T5,T6,T7
T9	30/05/2016	15/6/2016	T8

The entire duration is about 8 months according to COCOMO II analysis. Deadlines are estimated on previous experience.

3.2. Tasks Allocation

Here are presented the tables that explains the tasks development and division.

The division of the work is just indicative since the team often work together so many parts are developed in communion.

In the table are presented the name of the task, the time spent (or expected) and, for every member of the team, the assigned parts of the project.

4. RASD (from 5th October to 6th November)			
	1st week	2nd week	3rd week
Giovanni	Description of the problem, Glossary, Goals	Possible future implementations, Stakeholders, Functional Requirements	Actor: Guest, Actor: User, Alloy Code, Generated world
Riccardo	Domain properties, Assumptions, Proposed system	Non-functional requirements, Actors identifying, Possible scenarios	Domain model, Actor: Taxi Driver, Actor: Administrator, Tools

Design Document (from 12th November to 4th December)			
	1st week	2nd week	3rd week
Giovanni	Purpose, Scope, Overview, High level components	Component diagram: Web Service, Component interfaces, Deployment view, Algorithm design	Design overview, User interface and navigation flow, Further preview of UI, User experience
Riccardo	Glossary, References, Document structure, Component view	Component diagram: User Management, Runtime view, Selected architectural styles and patterns, User interface	E-R Diagram, Relational model, Requirements traceability

Integration Test Plan (from 6th December to 3rd January)			
	1st week	2nd week	3rd week
Giovanni	Purpose and scope, Glossary	Elements to be integrated, Integration testing strategy	Business logic test, Subsystem test
Riccardo	References, Entry criteria	Subcomponents and subsystems integration sequence, Persistence module test	Tools and test equipment, Program stubs and test data

Project Plan (from 3rd January to 10th January)			
	1st week	2nd week	3rd week
Giovanni	Description of the problem, Glossary, Goals	Possible future implementations, Stakeholders, Functional Requirements	Actor: Guest, Actor: User, Alloy Code, Generated world
Riccardo	Domain properties, Assumptions, Proposed system	Non-functional requirements, Actors identifying, Possible scenarios	Domain model, Actor: Taxi Driver, Actor: Administrator, Tools

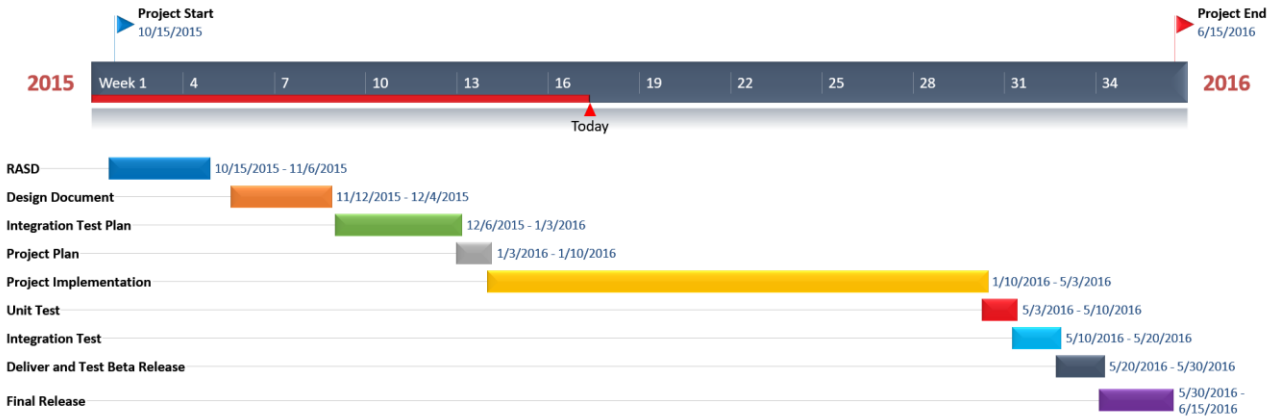
Project Implementation (from 10th January to 3rd May)				
	1st month	2nd month	3rd month	4th month
Giovanni	Web Tier	Business Tier	Persistence Module	External Components Interfaces
Riccardo	Web Tier	Business Tier	Persistence Module	External Components Interfaces

Unit Test (from 3rd May to 10th May)	
	1st week
Giovanni	Unit Testing
Riccardo	Unit testing

Integration Test (from 10th May to 20th May)	
	1st week
Giovanni	Integration Testing
Riccardo	Integration testing

Deliver and Test of Beta Release (from 20th May to 30th May)	
	1st week
Giovanni	Beta Release Testing
Riccardo	Beta Release testing

Final Release (from 30th May to 15th June)		
	1st week	2nd week
Giovanni	Analyzing Beta testing results and revisions/changes	Adjustments and final release
Riccardo	Analyzing Beta testing results and revisions/changes	Adjustments and final release



The picture above is the Gantt diagram of the project development.

4. Risk Analysis

After a risk analysis, we discover several possible issue that may occur during the development of the project. They are divided in Project Risks, Technical Risks and Business Risks.

4.1. Project Risks

- Requirement Change: it is an unpredictable risk that could provoke the entire re-structuration of the system. It can be managed using much reusable code as possible.
- Lack of experience: during the implementation, it is possible to find the programmer unable to develop some components. That will cause a delay on the deliveries since the team member has to update its knowledge.

4.2. Technical Risks

- Database Performance: the database cannot stand all the transactions due to exceeded number of users. The strategy to resolve this risk is to improve the Database Layer for instance buying a higher performance database.
- Server failures: this risk will cause the unavailability of the service and a possible lack of data (request or reservation done during the failure). It is unpredictable therefor the correct strategy is to have a periodic maintenance and deploy a team which restore the mainframe in case of failure.

- Lack of data: since this risk is unpredictable it is possible to avoid it providing the Database with a backup system (i.e. Mirroring).

4.3. Business Risks

- Competitors: the main business risk is the possibility that another company develops a similar application. This is unpredictable and unavoidable, the only possible strategy is to make the product the better as possible.
- Unused Product: the risk is that our product will not be used due to the existence of parallel services already commonly used. The strategy to avoid this risk is to implement specific functionalities that are exclusive of our system.