POLITECNICO
MILANO 1863

Software Engineering 2 A.A. 2015-16

Project Plan Document

for

MyTaxiService

by Francesco Picciotti (854021)

and Marco Travaglini (859186)

Table of Contents

Revision History

Date	**Reason For Changes**	**Version**
02/02/2016	First submission of the document	1.0

1. Introduction

1.1 Purpose

The following document aims to establish the phases that arrange the project lifecycle and explain how the team concerned has deal with them. Starting from estimation criteria, such like Function Points and Cocomo II, it is provided a project's evaluations in different terms.

The function point consists of fulfilling many points of evaluation describing the final system and it yields a value, which can be converted in LOC estimation by multiplying it with a prefix conversion factor.

On the other hand, the Cocomo II approach gives a richer result starting from the previous analysis of the function point for instance; it takes as parameter the LOC estimation and features about the team's skills and capability. As result, it provides a global estimation of the project in terms of efforts, such like the whole time, people need per month and costs.

Moreover, there are listed all the tasks belonging to the project's realization, in particular here it is focused on the precedencies among tasks and their operating method during the time, which are expressed through, respectively the precedence diagram and the Gantt diagram.

Finally, there is the description of all the most risks that would threatened the project during all its phases and affect its outcomes, than it also shown how respectively those risks would be handled by the team's components.

1.2 Definition and Acronyms

- *LOC: Lines Of Code*
- *COCOMO: COnstructive COst MOdel*

1.3 Reference documents

- *Cost estimation.pptx*
- *Lesson PM 1.pdf*
- *Example of project reporting(Meteocal Project).pdf*
- *Cocomo II reference manual*
- *Function Point reference manual*

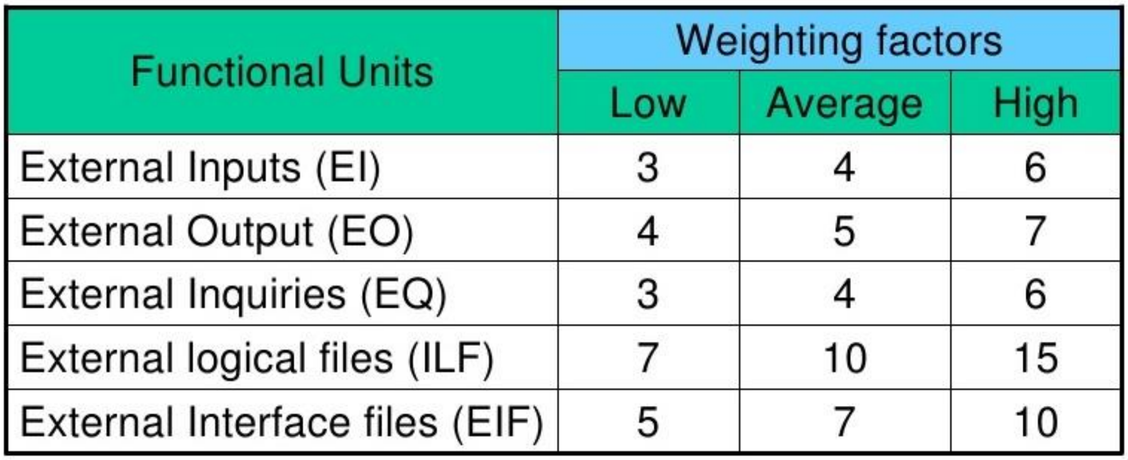
2. Cost Estimation

2.1 Function Point

In this section is described how we approach to the Function Point Analysis. This technique allow evaluating the effort needed for developing the project. Function point is a standard unit of measure that represent the functional size of a software application.
First, we extract all the functionalities of the system from the RASD document. Then we group them by the following schema:

- *Internal Logic File (ILF): represents data that is stored and maintained within the boundary of the application.*
- *External Logic File (ELF): represents the data that the application will use, but that is not maintained by our application.*
- *External Input (EI): represents an elementary operation that processes data or control information that comes from outside the application boundary.*
- *External Output (EO): represents an elementary operation that sends data or control information outside the application boundary.*
- *External Inquiries (EQ): represents an elementary operation that involves both input and output processes.*

Finally, using the following Weight Table, we assign a value to each functionality and we sum up them to obtain the Unadjusted Function Points (UFP), which will be used for the analysis.



Functional Units	Weighting factors		
	Low	Average	High
External Inputs (EI)	3	4	6
External Output (EO)	4	5	7
External Inquiries (EQ)	3	4	6
External logical files (ILF)	7	10	15
External Interface files (EIF)	5	7	10

2.1.1 Internal Logic File:

- *Customer Data: it is composed of a small number of fields so its structure is simple. SIMPLE*
- *Driver Data: it is composed of a small number of fields so its structure is simple. SIMPLE*
- *Ride data: it is composed of a small number of fields so its structure is simple. SIMPLE*
- *Reservation data: it is composed of a small number of fields so its structure is simple. SIMPLE*

ILF FP = 4x7 = 28 FPs

2.1.2 External Logic File:

- *Taxi driver info: it is composed of a small number of fields so its structure is simple. SIMPLE*
- *GPS position: respect to the data described above it has a more complex structure. MEDIUM*

ELF FP = 1x5 + 1x7 = 12 FPs

2.1.3 External Input:

- *Registration: it is a simple operation and it involves only a user entity. SIMPLE*
- *Ride processing: it is a complex operation and it involves three entities such as customer data, driver data and ride data. COMPLEX*
- *Reservation processing: it is a complex operation and it involves three entities such as customer data, driver data and reservation data. COMPLEX*
- *Driver availability: it is a simple operation and it involves only the driver entity. SIMPLE*
- *Driver confirmation: it is a simple operation but it involves three entities such as customer data, driver data and a request data. MEDIUM*
- *Data modification: it is a simple operation and it involves only a user data. SIMPLE*
- *Login/logout: they are simple operations and they involve only a user entity. SIMPLE*

EI FP = 5x3 + 1x4 + 2x6 = 31 FPs

2.1.4 External Output:

- *New Request Notification: it is a simple notification and it involves two entities such as driver data and a request data. SIMPLE*

EO FP = 1x4 = 4 FPs

2.1.5 External Inquiries:

- *<u>Request visualization:</u> it is a simple operation and it involves two entities such as a user data and a request data. SIMPLE*

EQ FP = 1x3 = 3 FPs

2.1.6 Total FP number:

UFP = (ILF +ELF + EI + EO + EIQ) FP = 28 + 12 + 31 + 4 + 3 = 78 FPs

2.2 Cocomo II

The ***Co****nstructive* ***Co****st* ***Mo****del is an algorithmic cost estimation model. The computation of this algorithm is based on our estimates of the project's size in terms of SLOC (Source Lines Of Code). The calculate of SLOC depends on the value of UFP, calculated before, and a conversion factor, which is a constant that vary based on the Programming Language that developers will use; in this case the Programming Language is Java EE so 46 is the constant value.*

SLOCs = conversionFactor x UFP = 46 x 75 = 3450

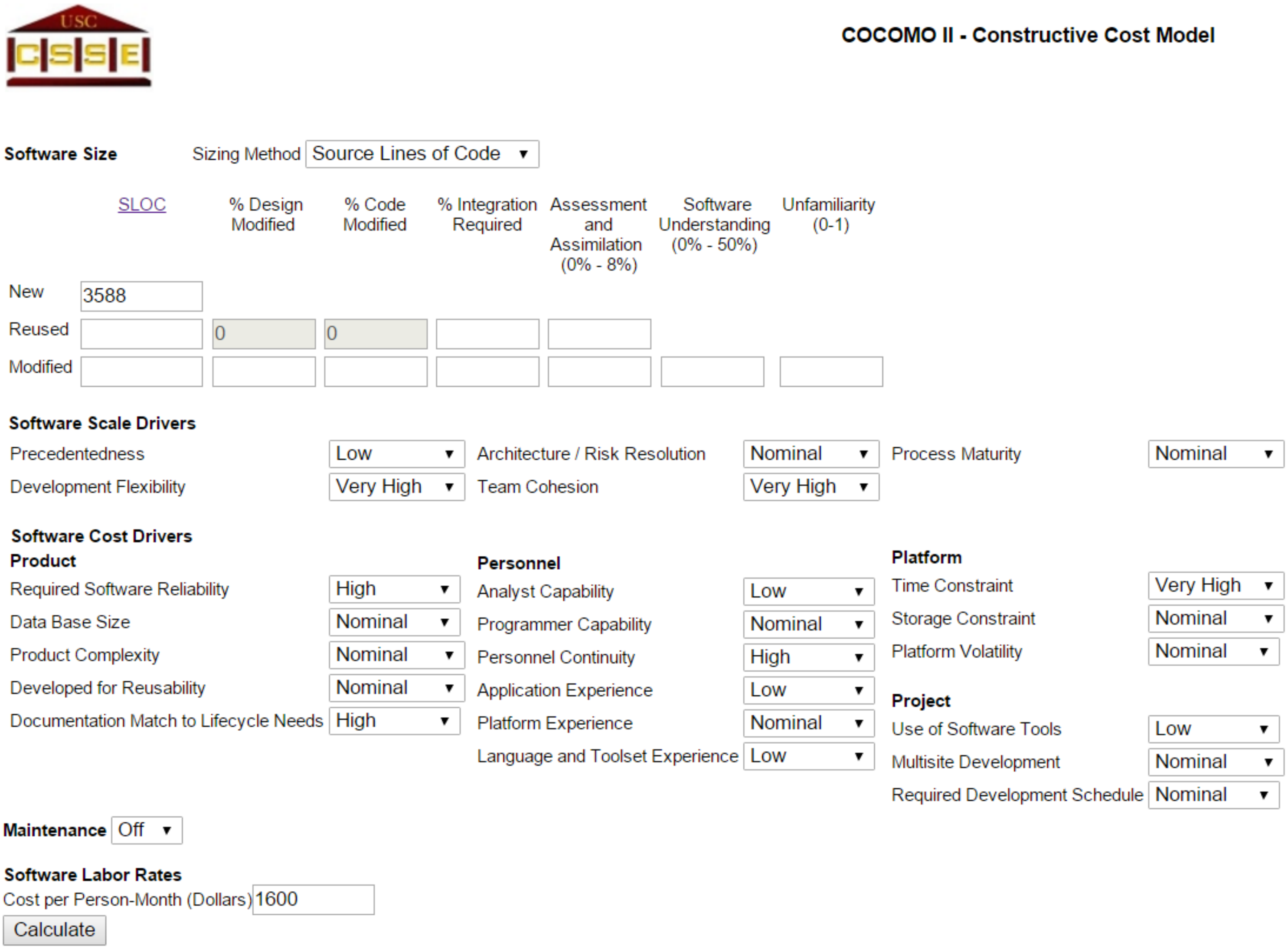
After that, to proceed with the Analysis, we have to define the Scale drivers and the Cost driver. There are 5 different Scale drivers and 17 Cost driver to be set. The first ones determines the exponent used in the effort equation and the second ones determines the multiplication factors used in that equation.
Then we can calculate the effort equation:

Effort = 2,94 x EAF x (kSLOC)^E

Where EAF is the Effort Adjustment Factor derived from the Cost drivers and E is an Exponent derived from the Scale drivers.
Finally, we calculate the duration of the software project in terms of number of months. This value is based on the effort predicted by the Effort Equation.

Duration = 3,67 x (Effort)^SE

Where Effort is the effort from the Effort Equation and SE is the schedule equation exponent derived from the Scale drivers.
In the following pages, there are two pictures that shows the results of our estimation, using the tool available at http://csse.usc.edu/tools/COCOMOII.php .



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	3588						
Reused		0	0				
Modified							

Software Scale Drivers

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

Software Cost Drivers

Product

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

Personnel

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

Platform

Time Constraint	Very High ▼
Storage Constraint	Nominal ▼
Platform Volatility	Nominal ▼

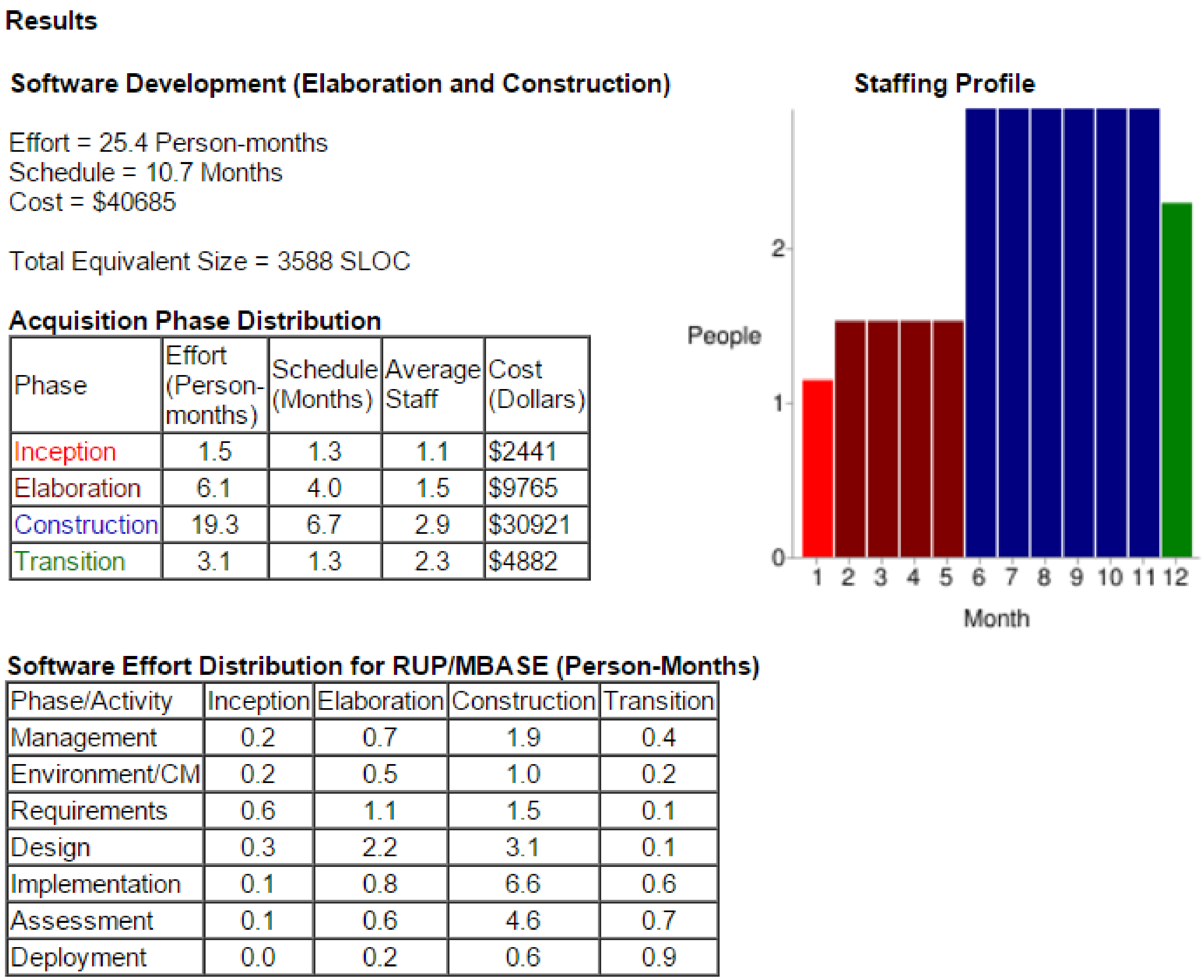
Project

Use of Software Tools	Low ▼
Multisite Development	Nominal ▼
Required Development Schedule	Nominal ▼

Maintenance Off ▼

Software Labor Rates

Cost per Person-Month (Dollars) 1600



Results

Software Development (Elaboration and Construction)

Effort = 25.4 Person-months
Schedule = 10.7 Months
Cost = $40685

Total Equivalent Size = 3588 SLOC

Acquisition Phase Distribution

Phase	Effort (Person-months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	1.5	1.3	1.1	$2441
Elaboration	6.1	4.0	1.5	$9765
Construction	19.3	6.7	2.9	$30921
Transition	3.1	1.3	2.3	$4882

Staffing Profile

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	6.6	0.6
Assessment	0.1	0.6	4.6	0.7
Deployment	0.0	0.2	0.6	0.9

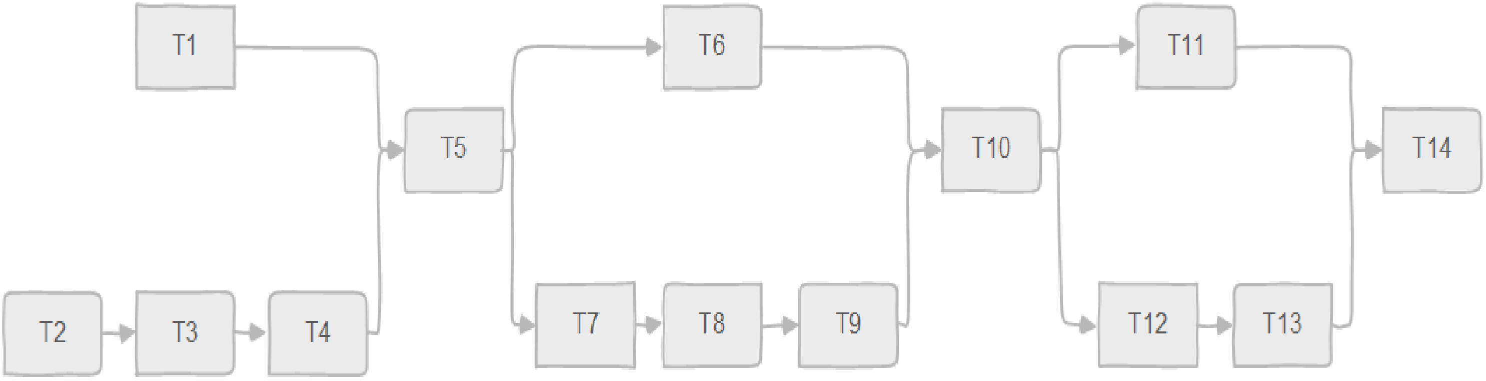
3. Tasks

Inside the project can be pointed out the following tasks that are above listed:

***T1. RASD definition**: It refers to the writing and redaction of the RASD document.*
***T2. Requirements elicitation:** The phase in which requirements are extracted and collected.*
***T3. Requirements formalization:** Here all requirements are well-formalized and checked their satisfiability.*
***T4. Model consistency proofing:** In this part, it is checked the model built-up based on the collected requirements and through Alloy the model is validated.*
***T5. RASD submission:** Document's delivery.*
***T6. DD definition:** It refers to the writing and redaction of the DD document.*
***T7. Architectural choices:** In this phase the software's architecture is defined.*
***T8. Algorithms design:** Here the most important algorithm are designed.*
***T9. User interfaces design:** Through Mockups there are described the graphic interfaces exposed to the user.*
***T10. DD submission:** Document's delivery.*
***T11. ITPD definition:** It states to the writing and redaction of the ITPD document.*
***T12. Unit testing:** Here the unit tests first are minutely chosen then are done*
***T13. Integration test:** The part in which is done the integration among component and tested its correctness*
***T14. ITPD submission:** Document's delivery.*

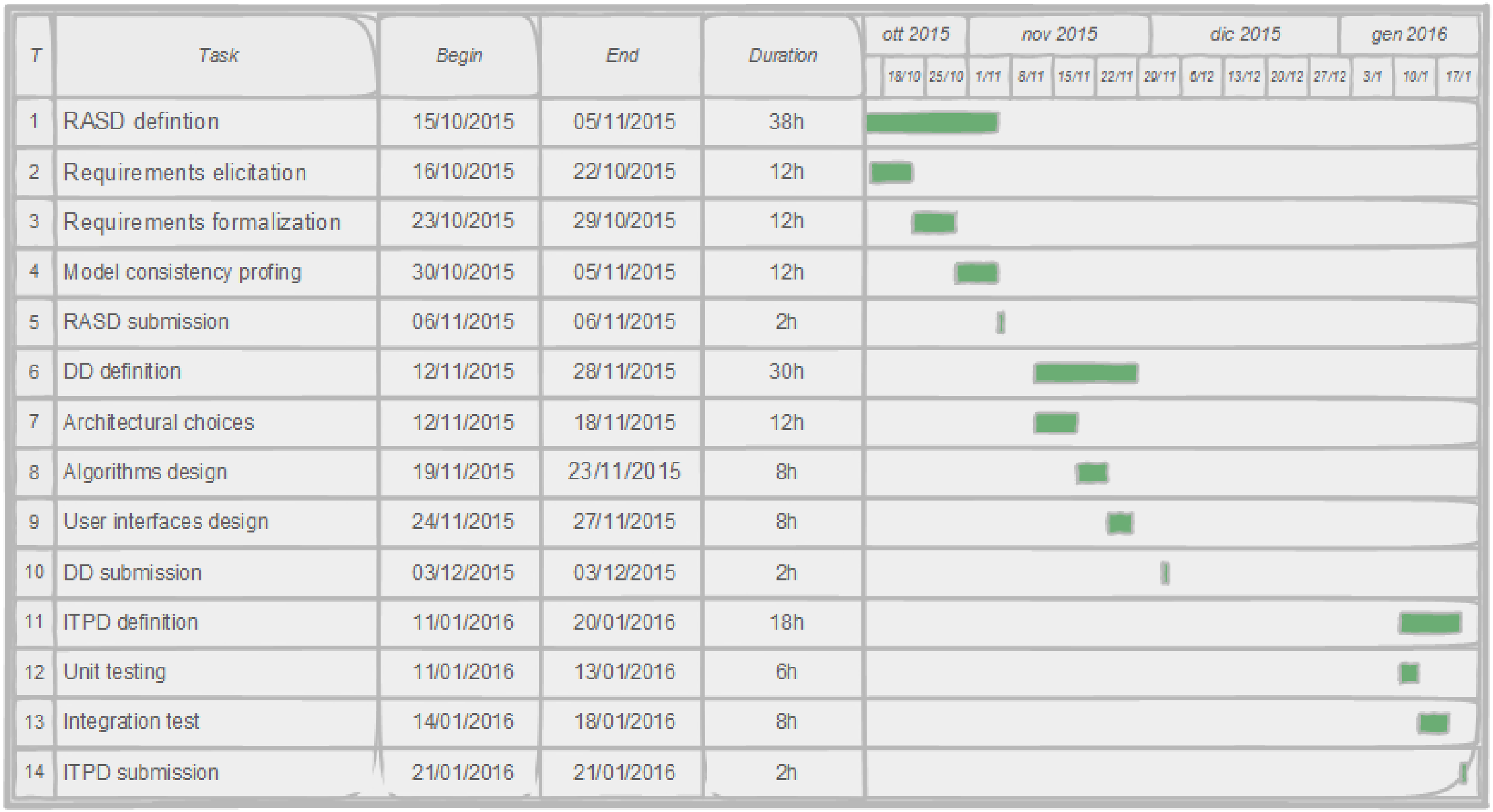
3.1 Precedence Diagram

The diagram above underlines the precedencies between each task arranging the whole project.



3.2 Gantt’s Diagram

The diagrams shows the duration, begin and end of each task.



T	Task	Begin	End	Duration
1	RASD defintion	15/10/2015	05/11/2015	38h
2	Requirements elicitation	16/10/2015	22/10/2015	12h
3	Requirements formalization	23/10/2015	29/10/2015	12h
4	Model consistency profing	30/10/2015	05/11/2015	12h
5	RASD submission	06/11/2015	06/11/2015	2h
6	DD definition	12/11/2015	28/11/2015	30h
7	Architectural choices	12/11/2015	18/11/2015	12h
8	Algorithms design	19/11/2015	23/11/2015	8h
9	User interfaces design	24/11/2015	27/11/2015	8h
10	DD submission	03/12/2015	03/12/2015	2h
11	ITPD definition	11/01/2016	20/01/2016	18h
12	Unit testing	11/01/2016	13/01/2016	6h
13	Integration test	14/01/2016	18/01/2016	8h
14	ITPD submission	21/01/2016	21/01/2016	2h

4. Resources Allocation

In the following paragraph, it is shown the allocation of the team's personnel.
In particular, each member carries each task out together with the rest of the group.

Task	*Francesco*	*Marco*
1. RASD definition	*38h*	*38h*
2. Requirements elicitation	*12h*	*12h*
3. Requirements formalization	*12h*	*12h*
4. Model consistency proofing	*12h*	*12h*
5. RASD submission	*2h*	*2h*
6. DD definition	*30h*	*30h*
7. Architectural choices	*12h*	*12h*
8. Algorithms design	*8h*	*8h*
9. User interfaces design	*8h*	*8h*
10. DD submission	*2h*	*2h*
11. ITPD definition	*18h*	*18h*
12. Unit testing	*6h*	*6h*
13. Integration test	*8h*	*8h*
14. ITPD submission	*2h*	*2h*

5. Risk Management

For what concerns the risk management, the team adopts a proactive approach to all possible risks that would threatened the development of the project. Hence, there is a list of the risks and the related strategy to take in order to minimize the deriving problems that could damage the team's work.

5.1 Project Risks

- *Members of the group are ill or busy at critical times*
  Probability: *Moderate*
  Effect: *Serious*
  Strategy: *The whole staff is assigned to same task at a time in order to avoid problem deriving from this scenario, so in this way the rest of the members could carry the project on.*

- *Changes to requirements that require major design rework are proposed*
  Probability: *Moderate*
  Effect: *Serious*
  Strategy: *During the project's design part, the team will trace each requirement on the current high-level architecture of the software in order to speed-up the search and modification to each of them.*

5.2 Technical Risks

- *The database used in the system cannot process as many transactions per second as expected*
  Probability: *Moderate*
  Effect: *Serious*
  Strategy: *The team has to estimate the number of final user after the software's first release through a market analysis and then overestimating it taking a larger infrastructure.*

- *Underestimating development time*
  Probability: *Moderate*
  Effect: *Serious*
  Strategy: *The team has to take in account to buy some software's component or program generators.*

- *Faults in reusable software components have to be repaired before these components are reused*
  Probability: *Moderate*
  Effect: *Serious*
  Strategy: *All reusable components has to be tested deeply with unit testing in order to ensure a correct working and a high reliability.*

5.3 Business Risks

- *The organization is restructured so that different management are responsible for the project.*
  Probability: *High*
  Effect: *Serious*
  Strategy: *Prepare a briefing document for the management showing off the importance of the project with the estimation of possibly final user attaching the result of the market analysis.*

- *Organizational financial problems force reductions in the project budget*
  Probability: *Low*
  Effect: *Catastrophic*
  Strategy: *The briefing document should underline even the profitability of the project so that cuts on the budget would be disadvantageous.*

- *The final software doesn't agree with the requirements*
  Probability: *Low*
  Effect: *Catastrophic*
  Strategy: *A copy of the RASD document has to be submitted to the Quality Assurance team during the development.*

6. Appendix

6.1 Tools

- *Microsoft Word: used for creating this document.*
- *Microsoft Visio: for creating the diagrams such like the Gantt and Precedence ones*
- *Microsoft OneDrive: use for sharing files among the group.*
- *COCOMO II: Used for the project's estimation*

6.2 Hours of work

- *Francesco: 10 hours.*
- *Marco: 10 hours.*