Project Plan

Matteo Greselin - Matr. 862908

Version 1.0

Contents

1 Introduction

This document will provide a general perspective about project, reguarding time spent, resource management, risk analysis.The project is been deelop by one single person but i will estimate how i could divide the work in a real situation with a real team. For the analisys has been used two diferent approaches:

1. **Functional Point Approach** and the results have been compared with the dimension of the project.

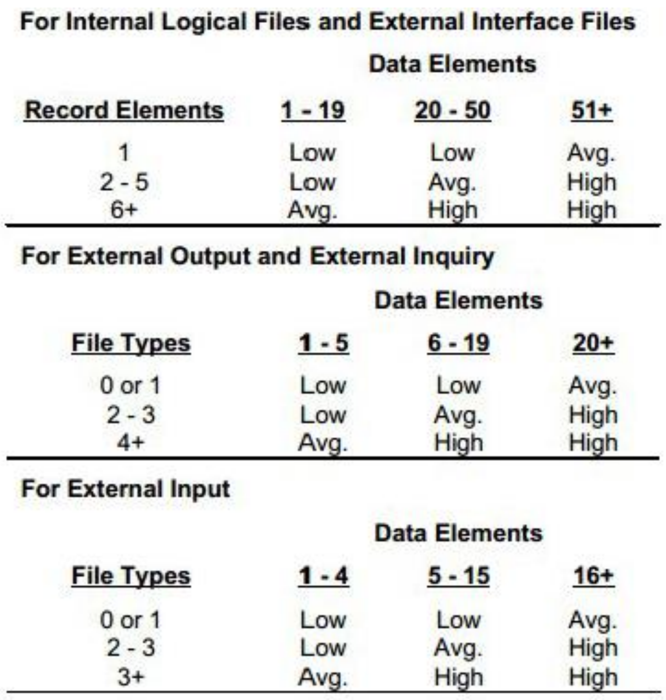
2. **COCOMO model** that has been used to evaluate the effort for smartCityAdvisor implemantetion and the results have been compared whit the time actually spent.

2 | Effort and Cost Estimation

2.1 Functional Point

2.1.1 Definistions

The Function Point estimation approach, is based on the amount of functionalities in a software and their complexity. Function Points estimators are useful since they are based on informations that are available early in the project life cycle
To perform this estimation we've based our parameters on the following tables, taken from COCOMO II, Model Definition Manual at:
http://csse.usc.edu/csse/research/COCOMOII/cocomo2000.0/CII_modelman2000.0.pdf.
In the following table are shown some table that describe the Functional Approach. The first one propose a way for determining complexity-level function counts, the second one defines the weights values that we've to use to perform the FP value and the final one provide definition about five classes of Functional Points.



For Internal Logical Files and External Interface Files

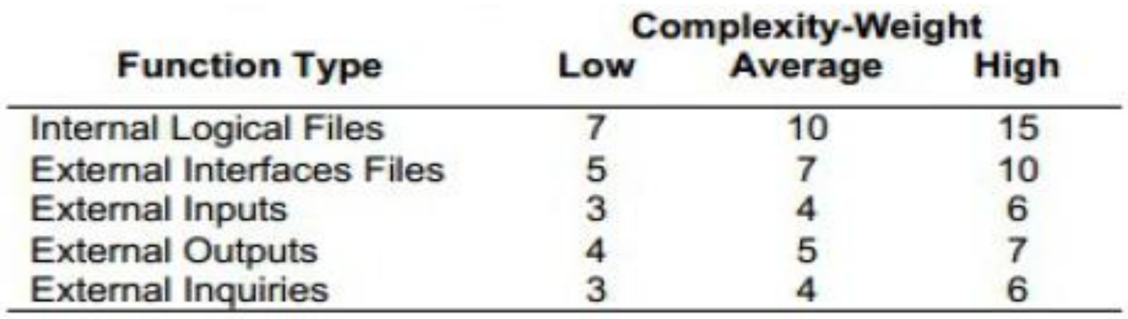
	Data Elements		
Record Elements	1 - 19	20 - 50	51+
1	Low	Low	Avg.
2 - 5	Low	Avg.	High
6+	Avg.	High	High

For External Output and External Inquiry

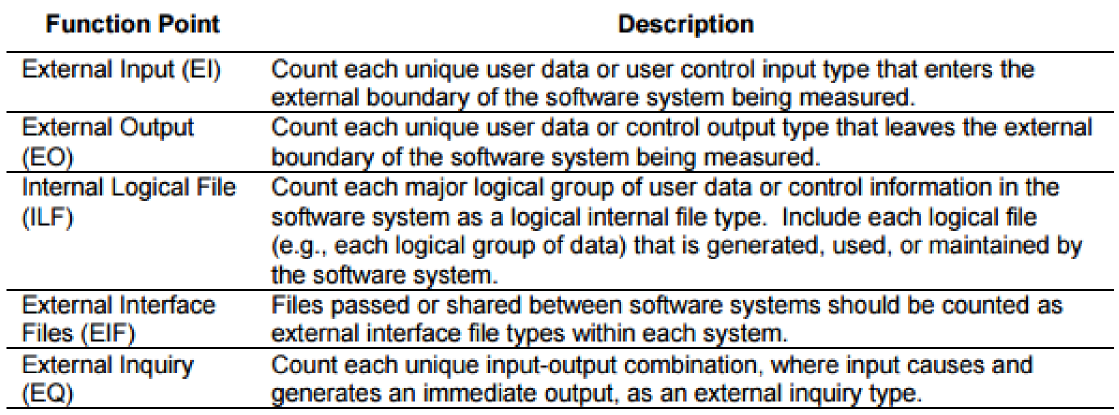
	Data Elements		
File Types	1 - 5	6 - 19	20+
0 or 1	Low	Low	Avg.
2 - 3	Low	Avg.	High
4+	Avg.	High	High

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



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



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.

2.1.2 Estimations

2.1.2.1 Internal Logical File

The application includes a number of ILFs that will be used to store the information about users, event, accident and each other situation taht can be consider relevant for the scope of this application. Now we analyse them more in detail in this section.

- The system stores few information about users, it saves: an username, a password and an email To provide updated weather forecasts, the system has to save the location, the effective weather forecasts already retrieved and the date/time related to an event.
- The system store information about special user: if it is an hospital, system will store name, location, specializations, phone number, emergency number; if it is a Fire Fighter Stations or a Polices Station, system will store name, location, phone number, emergency number; if it is a Public Tranport Company, system will store data about service as number of line, route, type of transport.
- The system store general information about city car parking, as name, location, opening and closing time, costs, number of car park.
- The system has to manage events, that are scheduled by City Administratio but that interest all users: it has to save the data, the route, the scope and the organization that require the event.

ILFs	Complexity	FP
Users	Avarage	10 +
Hospitals	Avarage	10 +
Fire Fighters Stations	Avarage	10 +
Police Stations	Avarage	10 +
Car Parking	Avarages	10 +
Events	High	15 +
Accident	High	15 =
Total		80

2.1.2.2 External Interface File

The application has to manage different data acquiret from external service that, in detail, will be obtained using the service APIs. They will be returned in JSON or XML format and must be converted to be used by our SmartCityAdvisor. This EIF can be resume in the next table:

ELF	Complexity	FP
Traffic Light	Avarage	7 +
Main Monitors	Low	5 +
Car Park Sensors	Avarage	7 +
Google Maps API	High	10 +
CO2 Sensors	Avarage	7 +
Transport GPS	High	10 =
Total		51

2.1.2.3 External Inputs

The application interacts with the user, without difference respect roles, to allow him/her to:

- Login/Logout: these are simple operations related only to the session manager, sothe contribute can be considered low
- Create / Delete / Update an Event: Each of these operations is made on a single form, only by city administratio, but are involved multiple entities so these operations are classified as complex
- Add / Remove an Help Request: as the last one also these set of operation involve multiple entities and also could be performed by different entities. These operation are classified as complex
- Reserve Car Parking: this operation is an automatic service. It involve system of the car park selected and user that require it. It can be considere as avarege complexity
- Require / Manage Traffic Situation: Each of this operation required the contribute of different component in the system, for managing the traffic situation in also required the interface with traffic light and main monitors. These are complex EIs.
- Require / Update Hospital Queue Situation: both the two types of operation can be considered with avarage complexity.
- Send / Receive / Manage Notification: low.

EIs	Complexity	FP
Login/Logout	Low	6 +
Create/Delete/Update an Event	High	18 +
Add/Remove an Help Request	High	12 +
Reserve Car Parking	Avarage	4 +
Require/Manage Traffic Situation	High	12 +
Require/Update Hospital Queue Situation	Avarage	8 +
Send/Receive/Manage Notification	Low	9 =
Total		69

2.1.2.4 External Outputs

The system has to manage operation that generate data for the external environment.

- Send Notification: the generation of a new message involve different components, in some cases it could require the approvation of City Administration, in other one it should be forward to all registered users. For this reasons this operation can be consider complex.
- New Help Request: when a new help request is inserted int the system, it involves Hospital, Fire Fighters and Police, it must update the Traffic Situation in order to advise citizens. It has an high complexity.
- Hospilat Queue Update: this is a periodical operation that requires to update all information that can be send to users if required. This has to be done each time a persone ask a medical visit and each time a mediccal visit is finished.
- Car Parking Situation Update : this operation must be done each time that someone perform a reservation or enter in a precise car parking or leave a precise car parking.

EOs	Complexity	FP
Send Notification	High	7 +
New Help Rquest	High	7 +
Hospital Queue Updating	Avarage	5 +
Car Parking Updating	Low	4 =
Total		23

2.1.2.5 External Inquiries

SmartCityAdministration allows to require different information. In the next will be presented these as EIs and will be presented them complexity:

- General Reports : this operation involve different component with respect to whi require the service (city administration has privileges for seeing everything, normal user could only see his/hers personal data). However this operatio is only a query for the DB, so it has low complexity.
- Event Scheduled: as general reports, this is a simple query for the DB. It has low complexity.
- Traffic Situation: this require the interaction with different component of the system that must be all synchronized. It has an high complexity
- Hospital Queue Situation: it depends on an automated system so it could be considered with avarage complexity
- Parking Situation: for this one can be done the same consideration that are valid for Hospital Queue Situation, avarage complexity
- Public Transport Real-Time Position: it is more similar to Traffic Situation, becouse require the interaction betwen different component in the system. It has high complexity.

EQs	Complexity	FP
General Reports	Low	3 +
Event Scheduled	Low	3 +
Require Traffic Situation	High	6 +
Require Hospital Queue	Avarage	4 +
Parking Situation	Avarage	4 +
Public Transport Position	High	6 =
Total		26 FP

2.1.3 Conclusions

Categories	Sum
Internal Logical Files	80 +
External Interfaces Files	51 +
External Inputs	69 +
External Outputs	23 +
External Inquiries	26 =
Total:	249

The line of code estimation is done by using the usual formula, that is

$$LOC = AVC \cdot Number\ of\ FP$$

by using 46 as AVC parameter:

$$249 \cdot 46 = 11454 \text{ LOC}.$$

2.2 COCOMO II

Here FP are used as a basis to estimate the size of the project in SLOC and then COCOMO method is applied to estimate the effort exploiting the SLOC estimation.

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	11454						
Reused		0	0				
Modified							

Software Scale Drivers

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

Software Cost Drivers

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

Maintenance Off

Software Labor Rates
Cost per Person-Month (Dollars) 2000

Software Development (Elaboration and Construction)

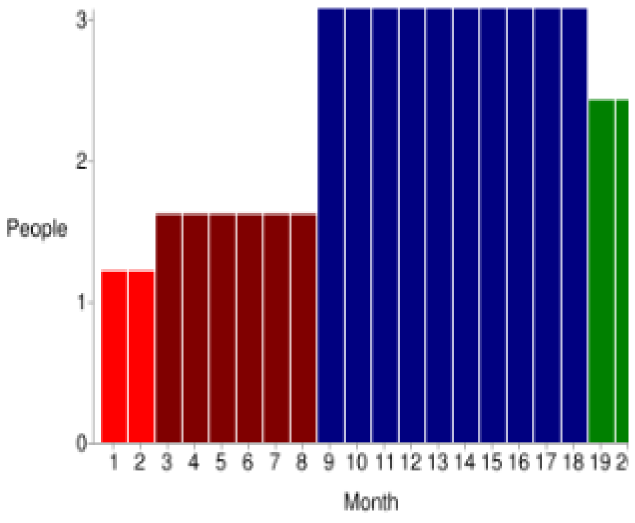
Effort = 41.0 Person-months
Schedule = 16.3 Months
Cost = $82046

Total Equivalent Size = 11454 SLOC

Acquisition Phase Distribution

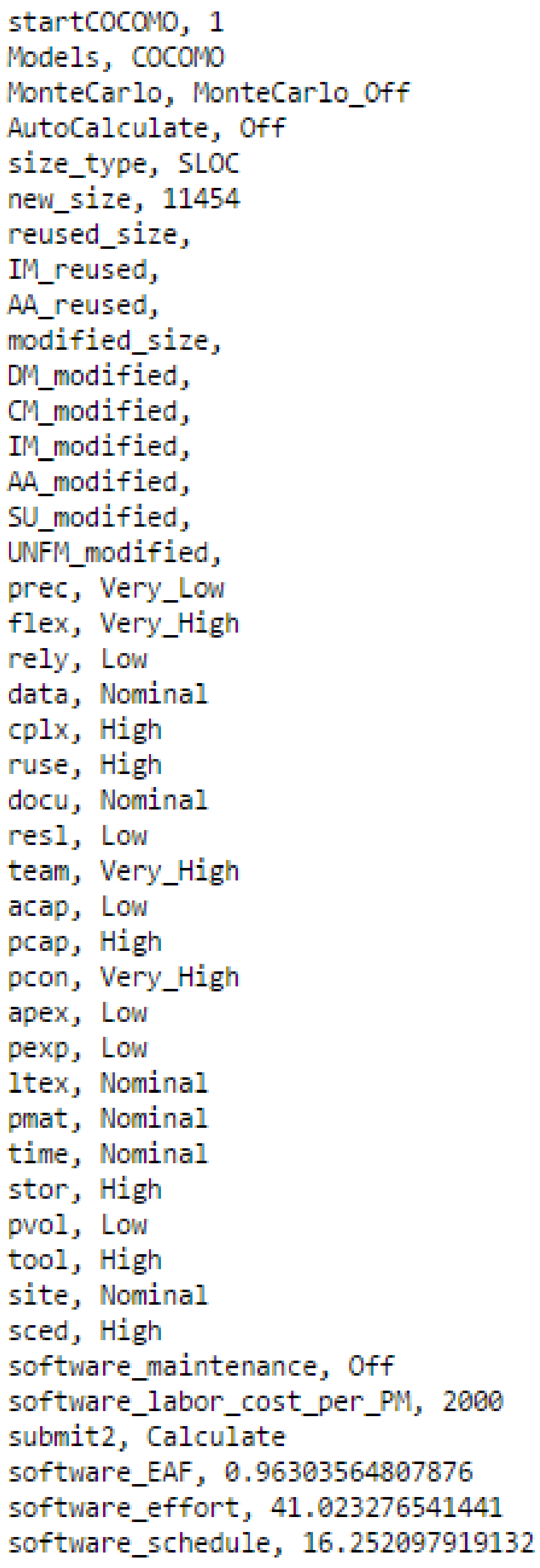
Phase	Effort (Person-months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	2.5	2.0	1.2	$4923
Elaboration	9.8	6.1	1.6	$19691
Construction	31.2	10.2	3.1	$62355
Transition	4.9	2.0	2.4	$9846

Staffing Profile



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

Phase/Activity	Inception	Elaboration	Construction	Transition
Management	0.3	1.2	3.1	0.7
Environment/CM	0.2	0.8	1.6	0.2
Requirements	0.9	1.8	2.5	0.2
Design	0.5	3.5	5.0	0.2
Implementation	0.2	1.3	10.6	0.9
Assessment	0.2	1.0	7.5	1.2
Deployment	0.1	0.3	0.9	1.5



```
startCOCOMO, 1
Models, COCOMO
MonteCarlo, MonteCarlo_Off
AutoCalculate, Off
size_type, SLOC
new_size, 11454
reused_size,
IM_reused,
AA_reused,
modified_size,
DM_modified,
CM_modified,
IM_modified,
AA_modified,
SU_modified,
UNFM_modified,
prec, Very_Low
flex, Very_High
rely, Low
data, Nominal
cplx, High
ruse, High
docu, Nominal
resl, Low
team, Very_High
acap, Low
pcap, High
pcon, Very_High
apex, Low
pexp, Low
ltex, Nominal
pmat, Nominal
time, Nominal
stor, High
pvol, Low
tool, High
site, Nominal
sced, High
software_maintenance, Off
software_labor_cost_per_PM, 2000
submit2, Calculate
software_EAF, 0.96303564807876
software_effort, 41.023276541441
software_schedule, 16.252097919132
```

2.2.1 Considerations About Choosen Drivers

Below are stated the main motivation for the COCOMO parameters chosen.

2.2.1.1 Software Scale Drivers

- Precedentedness: I have set this value to low, becouse this is my first JavaEE application, so it seems quite reasonable to set it tho this value
- Development flexibility: Flexibility has to be high, so i have setted it to very high.

- Architecrue / Risk resolution: I have done the risk analysis very well so i set this value to low.
- Team cohesion: In the reality this project has been perform by one single person, but in an hypothetical work situation the team cohesion must be to an high level. So i srt it to very high.
- Process maturity: This was evaluated around the 18 Key Process Area (KPAs) in the SEI Capability Model. Because of the goals were consistently achieved these values will be set to normal

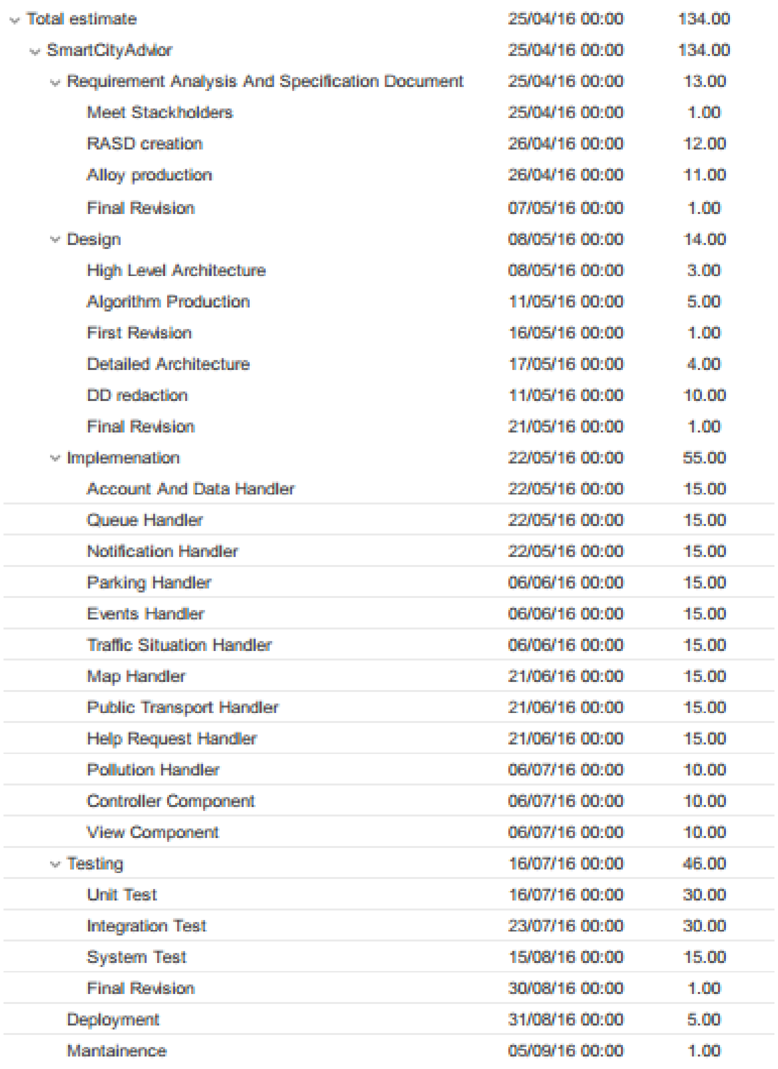
2.2.2 Software Cost Drivers

- Required Software Reliability: The application doesn't require any particular measure with respect to any other application, so it is setted to low
- Data Base Size: Here we have assumed to use the capacity of any other average application.
- Product Complexity: Set to high according to the nominal COCOMO II CPLEX rating scale.
- Developed for Reusability: SmartiCityAdvisor is designed with a SOA architecture, either for use or reuse. High is the value i choose.
- Documentation match to life-cycle needs: This driver its suitability set to nominal since each aspect of our system has been described in the RASD or in DD.
- Analyst Capability: Actually this my first analysis study, so i have decided to set our skill evaluation to low.
- Programmer Capability: i decide to set this value on high.
- Personnel continuity: i set this value to very high, since i am capable to work every day on this application.
- Application Experience: Being my first javaEE project this value is equal to low.
- Platform Experience: As above, for the same motivations i chose low.
- Language and Tool Experience: Here we have decided to keep in account of our java SE knowledge, so i set the value to nominal.
- Time Constraint: In our case this parameter is not relevant so it is reasonable to set it as nominal
- Storage Constraint: I have set it to high.
- Platform Volatility: The application platform shouldn't change too often, or rather we have forecast to keep it as stable as possible. so this value is set to low.
- Usage of Software Tools: I have used JavaEE, gitHub and several different frameworks to speed-up development so i set it on high.
- Multisite development: I set this driver to nominal, as actually is probably better to avoid a multisite development.
- Required development schedule: This is my first approach to a project of this dimension and i have preferred to produce a well documented development schedule. So I set this value to high.

For each clarification on drivers meaning, we refer, once again to the official manual:
http://csse.usc.edu/csse/research/COCOMOII/cocomo2000.0/CII_modelman2000.0.pdf

3 Task Identification

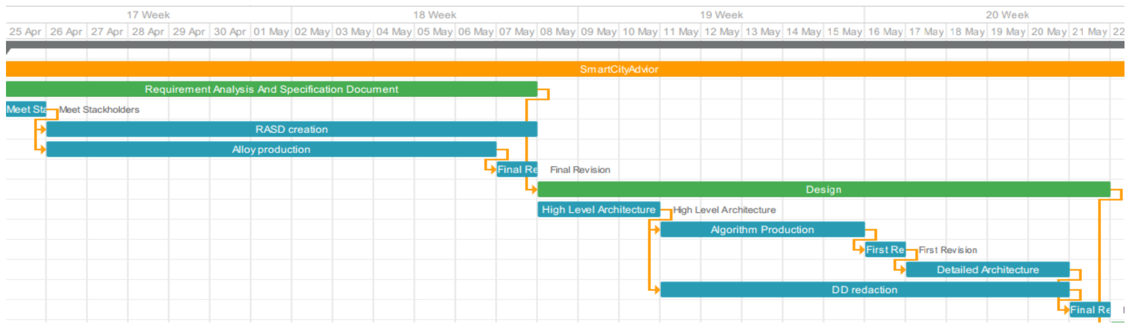
3.1 Task Scheduling

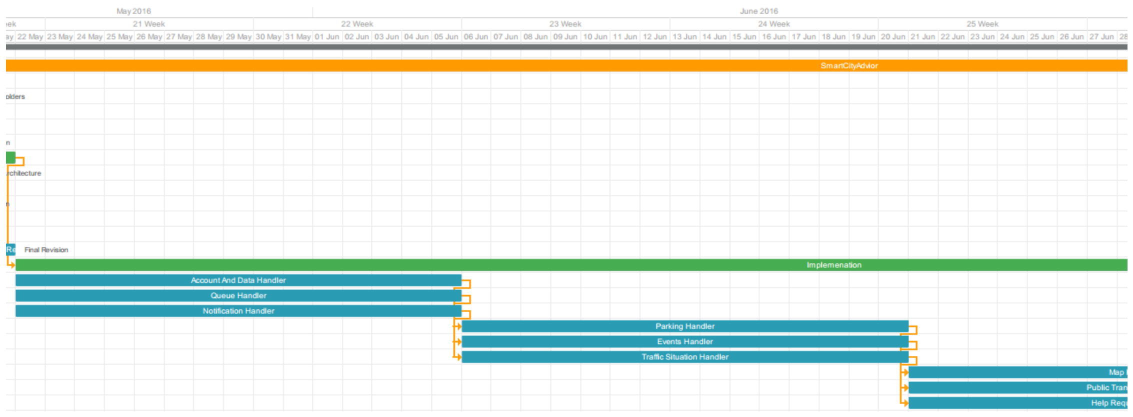


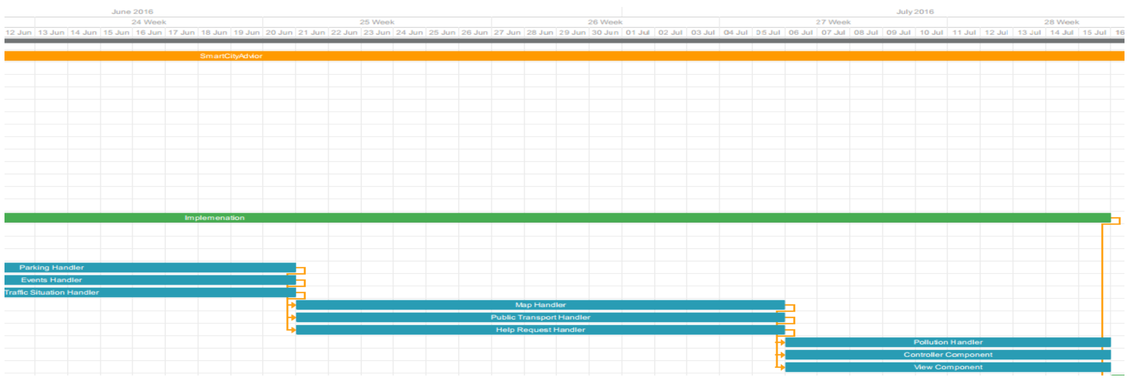
Total estimate	25/04/16 00:00	134.00
SmartCityAdvior	25/04/16 00:00	134.00
Requirement Analysis And Specification Document	25/04/16 00:00	13.00
Meet Stackholders	25/04/16 00:00	1.00
RASD creation	26/04/16 00:00	12.00
Alloy production	26/04/16 00:00	11.00
Final Revision	07/05/16 00:00	1.00
Design	08/05/16 00:00	14.00
High Level Architecture	08/05/16 00:00	3.00
Algorithm Production	11/05/16 00:00	5.00
First Revision	16/05/16 00:00	1.00
Detailed Architecture	17/05/16 00:00	4.00
DD redaction	11/05/16 00:00	10.00
Final Revision	21/05/16 00:00	1.00
Implemenation	22/05/16 00:00	55.00
Account And Data Handler	22/05/16 00:00	15.00
Queue Handler	22/05/16 00:00	15.00
Notification Handler	22/05/16 00:00	15.00
Parking Handler	06/06/16 00:00	15.00
Events Handler	06/06/16 00:00	15.00
Traffic Situation Handler	06/06/16 00:00	15.00
Map Handler	21/06/16 00:00	15.00
Public Transport Handler	21/06/16 00:00	15.00
Help Request Handler	21/06/16 00:00	15.00
Pollution Handler	06/07/16 00:00	10.00
Controller Component	06/07/16 00:00	10.00
View Component	06/07/16 00:00	10.00
Testing	16/07/16 00:00	46.00
Unit Test	16/07/16 00:00	30.00
Integration Test	23/07/16 00:00	30.00
System Test	15/08/16 00:00	15.00
Final Revision	30/08/16 00:00	1.00
Deployment	31/08/16 00:00	5.00
Mantainence	05/09/16 00:00	1.00

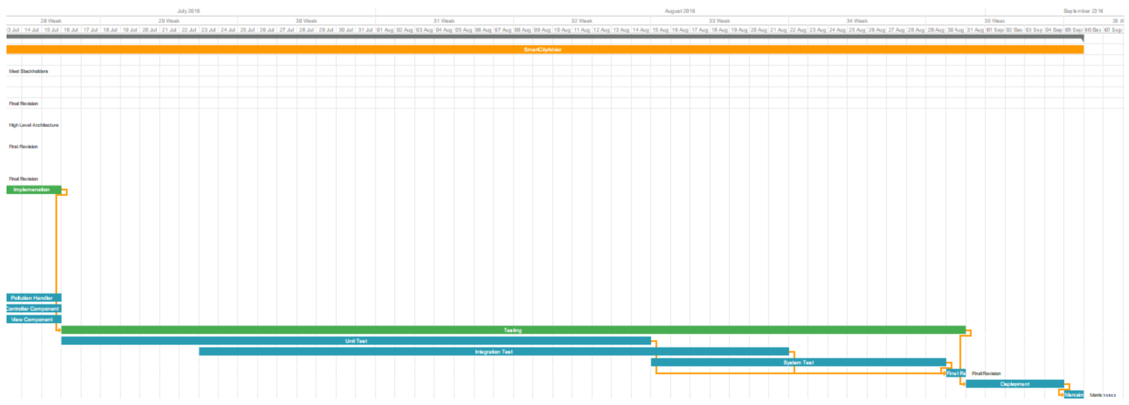
3.2 Gantt Diagrams

In this section are reported some gantt diagrams, they have been divided but is already clear the work-flow and the sequence of event.









4 Resources Allocation

The team work for this project is formed by only one person. For this reason i wil simulate to have 2 collegues with whom to share the work.Team members can be considered as resources of the project. In this chapter it is wanted to assign at each task individuated before a corresponding person. In order to increase the document's comprehensibility, in next sections i will use the following actors:

- Francesco Verdi
- Antonio Rossi
- Marco Bianchi

4.1 Requirement analysis

Task	**Resource**	**Motivation**
Meet Stackholders	All team members	In this phase there is the need to have the better understanding of the future requirements possible
RASD Creation	Antonio Rossi, Marco Bianchi	It's important to have different point of view in order to undersand each requirements of the project
Alloy Production	Francesco Verdi	This phase is complex, but a single person can better manage it
Final revision	All team members	Having all the members working on the revision of this document allows the better cohesion in the future phases

4.2 Design

Task	**Resource**	**Motivation**
High level architecture	All the members	Fair job division
Algorithm design	Marco Bianchi	Fair job division
First Revision	All the members	Fair job division
Detailed Architecture	Antonio Rossi	Fair job division
DD production	Sequentially: • Antonio Rossi • Marco Bianchi • Fancesco Verdi	Francesco Verdi works only on DD production, the other two guys will work on it with the redaction of them assigned part as alghorithm design and detailed architecture
Final Revision	All team members	Having all the members working on the revision of this document allows the better cohesion in the future phases. Each of them can check and understand the parts done by the others

4.3 Implementation

Task	**Resource**	**Motivation**
Account And Data Handler Handler	Francesco Verdi	This component is one of the first that must be implemented. It is the component that manage the interaction with DB
Queue Handler Handler	Marco Bianchi	This component has an high interaction with the DB so it can be develop with the previous one in order to directly develop a correct integration between them.
Notification Handler	Antonio Rossi	As the previous one, this component has an high interaction with the DB so it can be develop with the previous one in order to directly develop a correct integration between them
Parking Handler	Francesco Verdi	This component is directly connected with one of the previous already implemented, but it's not in conflict with the two that will be developed in parallel. This should guarantee a better work-flow continuity
Event Handler	Marco Bianchi	This component is directly connected with one of the previous already implemented, but it's not in conflict with the two that will be developed in parallel. This should guarantee a better work-flow continuity
Traffic Situation Handler	Antonio Rossi	This component is directly connected with one of the previous already implemented, but it's not in conflict with the two that will be developed in parallel. This should guarantee a better work-flow continuity
Map Handler	Francesco Verdi	This component is directly connected with the previous already implemented. But it is not in conflict with the two that will be developed in parallel. This should guarantee a better work-flow continuity
Public Transport Handler	Marco Bianchi	This component is directly connected with the previous already implemented. But it is not in conflict with the two that will be developed in parallel. This should guarantee a better work-flow continuity
Help Request Handler	Antonio Rossi	This component is directly connected with the previous already implemented. But it is not in conflict with the two that will be developed in parallel. This should guarantee a better work-flow continuity

Task	Resource	Motivation
Pollution Handler	Francesco Verdi	This component is directly connected with the previous already implemented. But it is not in conflict with the two that will be developed in parallel. This should guarantee a better work-flow continuity
Controller	Marco Bianchi	The controller is composed by two similar component. Two team member can done the task in parallel
View	Antonio Rossi	The view is composed by two similar component. Two team member can done the task in parallel

4.4 Testing

Task	Resource	Motivation
Unit tests over modules	All team members	Each team member should dedicate part of his time in doing tests over his specific component
Integration Tests	All the members	Members are employed to perform the implementation of other components. After completion, integration tests must proceed in parallel with implementation and they involve part of each member's time
System Tests	All team members	This tests are done when the implementation phase is over. All the members are available and can work together. These tests will include also security test and usability test.

4.5 Deployment

Task	Resource	Motivation
Deployment	All team members	At this point the application could be considered 'on work' and the project is tecnically finished. From this point will start the 'Manteinace Phase', that hasn't a really end date.

4.6 Maintenance

Task	Resource	Motivation
Maintenance	All team members	Every possible maintenance action, because of the complete project knowledge of each member, can be handled by who is available

5 Appendix

5.1 Used Tools

We used the following tools to make the ITP document:

- LYX 2.1: to redact and format the document
- GANTT PRO: to produce the Gantt chart. Available at: https://ganttpro.com
- COCOMO II simulator. Avabile at http://csse.usc.edu/tools/COCOMOII.php

5.2 Time Spent

For redact, correct and review this document spent almost 15 hours.