
POLITECNICO
DI MILANO

# SmartCityAdvisor

## Project Management

*Hasancan Sayılan*

# Contents

# INTRODUCTION

## Purpose and Scope

This document is made with the purpose of planning the project SmartCityAdvisor. The main goal is to analyze and give detailed information about the complexity of the project with the information of estimated general costs, and efforts to help fixating the required budget and necessary resources.

This document will first describe the estimated size of the project with Function Points and COCOMO approaches. The lines of code, costs and efforts which are required for the project will also be explained.

In the next section, the elements in the first section will be reused to make a possible schedule for all activities mentioned in previous documents those are RASD , DD and ITPD.

In the final section, possible risks will be estimated and explained which may be encountered during the project.

## Definitions, Acronmys, Abbreviations

FP : Function Points

ILF : Internal Logic File

ELF : External Logic File

EI : External Input

EO : External Output

EQ : Extarnal Inquiries

RASD : Requirement Analysis and Specifications Document

DD : Design Document

ITPD : The Integration Test Plan Document

### Referance Documents

- RASD.pdf (RASD document of the same project)
- DD.pdf (Design document of the same project)
- Function Points Table
- COCOMO II Manuel
- Project Planning Exampel Document.pdf (as sample)

## PROJECT SIZE, COST and EFFORT ESTIMATION

In this section, a variety of estimations about the expected size, cost and required efforts of the project SmartCityAdvisor will be provided.

As mentioned in the previous section, the size estimation part will be made using FP approach by using the Function Points complexity evaluation table. This part will include all the main activities and estimation of the amount of lines of codes of the project SmartCityAdvisor.

The cost and effort estimation will be made using COCOMO approach with EO, EQ, ELF and ILF.

### Size Estimation : Function Points

The Function Points approach gives an estimation of the size of the project considering the functionality and complexity of the activities. By using COCOMO approach as a fundamental for the estimations the complexities can be generally calculated and they will be provided in the coming sections labeled as low, avarage and high which are related to the number of the elements, data and files in an activity.

The following tables in the coming sections are calculated using COCOMO II Manuel as a base.

## Internal Logic Files (ILFs)

The project SmartCityAdvisor can only be functional with the variety of data collected by sensors. Those sensors calculate the $CO_2$ level in the air, and function according to the relation between it and the tolerance. Also the number of cars entering the city center will be taking into account which is also observed by sensors. Those two activities will be considered differently in the tables. The system overally relies on the data collected by the sensors and those data are critical.

Also the sensors collect data from the parking areas to provide available slots in those parking sites to the clients.

The system requires no registration from the clients thus there are no datas to be collected in this area.

| ILF | Complexity | FPs |
|---|---|---|
| CO2 Data | Low | 7 |
| CarEntry Data | Avarage | 10 |
| Park Data | Low | 7 |
| Total | | 24 |

Here in this table $CO_2$ Data is considered with low complexity as it does not change heavily on a graphical base but the CarEntry Data is considered with avarage complexity since the traffic can be changed with ease and it also relies on the previous data changes.

## External Logic Files (ELFs)

There is only one external data source that is the GPS with the usage of normal meanings and also providing the alternative routes when the city center entrance is not allowed. This part is also crucial and very risky since a mistake in this calculation can create huge problems such as a very overloaded traffic which may devastate the city life of Milan. On the other hand, with the appopriate usage of this data will be very functional for the project SmartCityAdvisor.

| ELF | Complexity | FPs |
|---|---|---|
| GPS Data | Avarage | 10 |

## External Inputs (EIs)

As mentioned in the previous sections, the system does not require registration or login/logout operations from the clients as they are not a necessity but instead they could create data with no importance. The clients can only check the availability of the city center entrance, parking areas and alternative routes which are basically External Inquiries (Eqs)

For the administrators, they can update a city zone which means basically updating the sensors in the zone which will have a complexity of 6 and also update the actuators in the zones which will also have a complexity of 6.

| EI | Complexity | FPs |
|---|---|---|
| Update Sensors | High | 6 |
| Update Actuators | High | 6 |
| Total | | 12 |

All the external inputs require heavy and many activities to be accomplished. Thus their complexity is high.

## External Inquiries (Eqs)

These are the activities which are done by clients with ease but could be complex since they also rely on the functions and operations of the main system.

As mentioned in the previous sections, the clients can check availability of the city entrance, check availability of the parking areas and find alternative routes. All of these operations can be done with one click on the button by clients but they rely on heavy activities done by the main system of the software project.

| EQ | Complexity | FPs |
|---|---|---|
| Check Availability | High | 6 |
| Check Parking Areas | Avarage | 4 |
| Find Alternative Routes | High | 6 |
| Total | | 16 |

## External Outputs (Eos)

Beside of the External Inquiries (EQs) the main system of SmartCityAdvisor also interacts with the clients, giving them basic notifications about the availability of the city center entrance and parking areas.

| EO | Complexity | FPs |
|---|---|---|
| Availability Notification | Low | 4 |
| ParkingAreas Notification | Low | 4 |
| Total | | 8 |

## Overall Estimation

By summarizing the previous tables, the overall estimation of the complexity of the project SmartCityAdvisor is given in the following table

| Function Type | Value |
|---|---|
| Internal Logic Files | 24 |
| External Logic Files | 10 |
| External Inputs | 12 |
| External Inquiries | 16 |
| External Outputs | 8 |
| Total | 70 |

Hence by using the following table and the formula

Sloc = Avc * FP

The estimation of the number of codes can be calculated considering the Java Enterprise Edition as the code development platform base. Here we select the lower bound as 46 and upper bound as 67. Hence the number of lines in the code are :

| |
|---|
| Lower Bound : SLOC = 70 * 46 = 3220 |
| Upper Bound : SLOC = 70 * 67 = 4690 |

## Cost And Effort Estimation : COCOMO II

This section will give information about the costs and efforts which are required by the project SmartCityAdvisor using COCOMO II approach.

## Scale Driver

To be able to evaluate the values of scale drivers, COCOMO II table will be used during this process. Hence the following table gives the evaluated values of the scale drivers :

| Scale Driver | Factor | Value |
|---|---|---|
| Precedentedness (PREC) | Low | 4.96 |
| Development Flexibility(FLEX) | Low | 4.05 |
| Risk Resolution (RESL) | Very High | 1.41 |
| Team Cohesion (TEAM) | Very High | 1.10 |
| Process Maturity (PMAT) | Level 3 | 3.12 |
| Total | | 14.64 |

The elements in the previous table is explained as :

- Precedentedness : It represents the experience of the team working on the project which means if the team has an experience similar to this project before the PREC factor is high. Since this is the first time this team is making a project like this the PREC factor is low with the value equal to 4.96 with respect to the COCOMO II table.

- Development Flexibility : It represents the flexibility of the development process which means if there are no boundries and constraints the FLEX factor is high. For this project, the functions and operations relies heavily on some constraints such as the $CO_2$ level tolerance and some other spesified in RASD document. Thus FLEX factor is low with the value equal to 4.05.
- Risk Resolution : It represents the level of awareness and reactiveness considering the risks which means if the risk documentation gives clear and definite information then the RESL factor is high and in this situation it is very high regarding the information given in the risk documents. Hence the RESL factor value is equal to 1.41
- Team Cohesion : It represents the relation between the team members which means if the team members know each other well, can coordinate completely and work together in a good state then TEAM factor is high. For this situation it is very high since the team consist of only one member. Hence the TEAM factor value is equal to 1.10
- Process Maturity : It represents the goal achievements and developments of the process. Again the team consist of only one member and this is the first project ever to be made by the team the PMAT factor is level 3 with the value equal to 3.12.

## Cost Drivers

The following table gives information and values of the cost drivers.

| Cost Driver | Factor | Value |
|---|---|---|
| Required Software Reability (RELY) | High | 1.10 |
| Database Size (DATA) | Nominal | 1.00 |
| Product Complexity (CPLX) | Very High | 1.34 |
| Required Reuseability (RUSE) | Nominal | 1.00 |
| Documentation Match to Life-Cycle Needs (DOCU) | Nominal | 1.00 |
| Execution Time Constraint (TIME) | Very High | 1.29 |
| Main Storage Constraint (STOR) | Nominal | 1.00 |
| Platform Volatility (PVOL) | Nominal | 1.00 |
| Analyst Capability (ACAP) | High | 0.85 |
| Programmer Capability (PCAP) | Nominal | 1.00 |
| Application Experience (APEX) | Low | 1.10 |
| Platform Experience (PLEX) | Low | 1.09 |
| Language and Tools Experience (LTEX) | Nominal | 1.00 |
| Personnel Continuity (PCON) | Very Low | 1.29 |
| Usage of Software Tools (TOOL) | Nominal | 1.00 |

| | | |
|---|---|---|
| Multisite Development (SITE) | Extra High | 0.80 |
| Required Development Schedule (SCED) | High | 1.00 |
| Total | | 1.3231 |

The elements of the previous table is explained as :

- Required Software Reliability : The RELY factor is considered as high for this project because a failure in the system can cause devastating problems to the clients life such as if the availability is set so FALSE when it is supposed to be true, it can cause a traffic overload. Thus the value is equal to 1.10.
- Database Size : Given an estimation of the required database as 300 KB (not necessarily to be high since no client data will be stored) and using the SLOC boundaries and D/P relation the DATA factor is nominal and the value is equal to 1.00
- Product Complexity : Considering the COCOMO II rating scale , CPLEX is set to very high with the value equal to 1.34
- Required Reuseability : The required reuseability is limited in scope to project itself hence RUSE factor is nominal with the value equal to 1.00
- Documentation Match to Life-Cycle Needs : This factor represents the status of the explanation of the application requirements in the document hence its factor is nominal with the value 1.00.
- Execution Time Constraint : This factor represents the CPU usage to complete the hardware functionalities. As the system mostly relies on hardware such as sensors which do the critical operations of the project SmartCityAdvisor the factor is very high with the value 1.29.
- Main Storage Constraint : This factor represents the storage usage considering the hardware operations. The system does not require massive storage thus the factor is nominal and the value is equal to 1.00.
- Platform Volatility : The base of the system is not considered to be changed during the process of the project SmartCityAdvisor hence the factor is set to nominal with the value 1.00.
- Analyst Capability : During the development and documentation of SmartCityAdvisor, the analysis of the project has been explained, described and documented in a good state thus the factor is set to high with the value 0.85.
- Programmer Capability : The programming process of the project has not been made so far and the programming knowledge of the team is not very high hence the factor is set to nominal with the value 1.00.

- Application Experience : The team has some knowledge about Java Programming and applications but it is not high. Also the team has never done a project like this and never worked with Java EE system of this kind hence the factor is low with the value 1.10.
- Platform Experience : The team has some experience with user interfaces and databases but not with Java EE platform and server side development. Hence thef actor is low with the value 1.09.
- Language and Tools Experience : Considering the platform experience this factor is set to nominal with the value 1.00
- Personnel Continuity : The time which can be spent on this project is very limited because of the school process such as other projects, exams etc. Hence this factor is set to very low with the value 1.29.
- Usage of Software Tools : The application environment is in a good state but not completed hence the factor is set to nominal with the value 1.00.
- Multisite Development : Since the team consist of only one person this factor is set to extra high with the value 0.80.
- Required Development Schedule : The efforts and works of the first phase of the project which are RASD and DD preperation is much larger than for the current preperations this factor is set to high with the value 1.00.

## Effort Equation

We need a final equation to calculate the estimation of the efforts in Person-Month (PM) that is

$$Effort = A * EAF * KSLOC^{E}$$

Here :

A = 2.94 (for COCOMO II)

EAF = product of all cost drivers

E = exponent derivation from the scale drivers. It can be computed as the following :

$$E = B + 0.01 * \sum_{i} SF(i) = B + 0.01 * 14.64 = 0.91 + 0.1464 = 1.0564$$

Here SF(i) is the sum of the values of scale drivers.

Here B = 0.91 (for COCOMO II)

Hence for the lower bound, the effort is

$$Effort = A * EAF * KSLOC^{E} = 2.94 * * 3220^{1.0564} = 2.94 * 1.3231 * 5078.073 = 19.7532 = 20\ PM$$

And for the upper bound, the effort is

$$Effort = A * EAF * KSLOC^{E} = 2.94 * * 4690^{1.0564} = 2.94 * 1.3231 * 7554.869 = 29.3878 = 29\ PM$$

## Schedule Estimation

We also need a final equation to calculate the estimation of schedule which is

$$Duration = 3.67 * Effort^{F}$$

Here F is :

$$F = 0.28 + 0.2 * (E - B) = 0.28 + 0.2 * 0.1464 = 0.30928$$

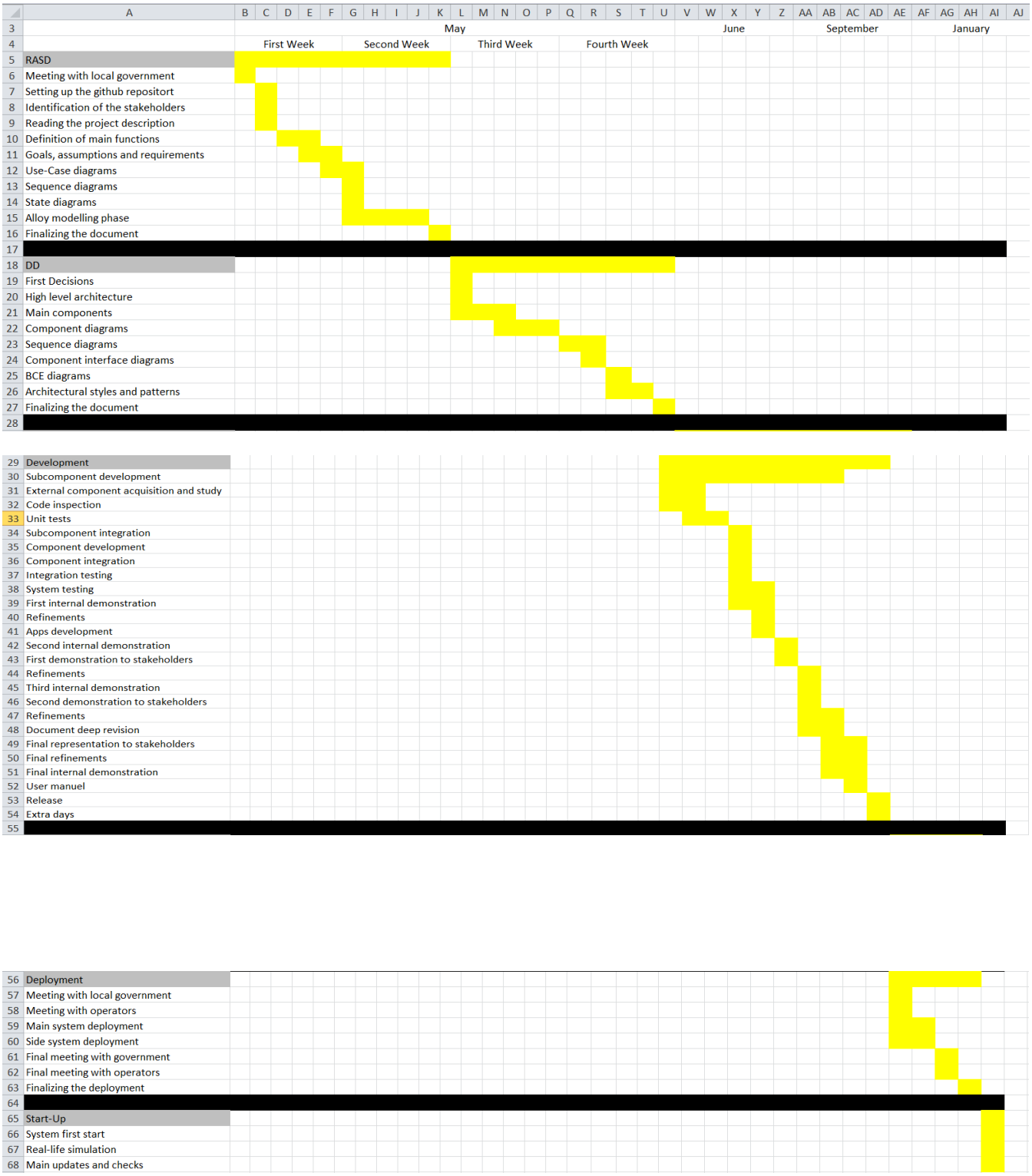
Thus for the lower bound the value is

$$Duration = 3.67 * Effort^{F} = 3.67 * 2.5257 = 9.269 = 9\ Months$$

For the upper bound the value is

$$Duration = 3.67 * Effort^{F} = 3.67 * 10.398 = 10\ Months$$

# SCHEDULE


May
June
September
January
First Week
Second Week
Third Week
Fourth Week
RASD
Meeting with local government
Setting up the github repositort
Identification of the stakeholders
Reading the project description
Definition of main functions
Goals, assumptions and requirements
Use-Case diagrams
Sequence diagrams
State diagrams
Alloy modelling phase
Finalizing the document
DD
First Decisions
High level architecture
Main components
Component diagrams
Sequence diagrams
Component interface diagrams
BCE diagrams
Architectural styles and patterns
Finalizing the document
Development
Subcomponent development
External component acquisition and study
Code inspection
Unit tests
Subcomponent integration
Component development
Component integration
Integration testing
System testing
First internal demonstration
Refinements
Apps development
Second internal demonstration
First demonstration to stakeholders
Refinements
Third internal demonstration
Second demonstration to stakeholders
Refinements
Document deep revision
Final representation to stakeholders
Final refinements
Final internal demonstration
User manuel
Release
Extra days
Deployment
Meeting with local government
Meeting with operators
Main system deployment
Side system deployment
Final meeting with government
Final meeting with operators
Finalizing the deployment
Start-Up
System first start
Real-life simulation
Main updates and checks

## RESOURCE ALLOCATION

Since the team has ony one member, resource allocation is the same with the schedule.

## RISK MANAGEMENT

This section explains the risks that can be occured during the project development. This part includes all the major risks which are dangerous and may effect human life.

1. The first and main issue is a political one which is related to the city city administration. Since this project is related to the city of Milan, the government may refuse to have SmartCityAdvisor because of the following reasons :
    a. There might be a change in the government.
    b. The failure of the project can cause mistrust of the citizens to the government.
    c. There might be a financial crisis.
    d. A solution for all the problems mentioned, the government itself can be concluded during the development of the project. If the government can witness the process even work as a consultant, this may ease the threats to both citizens and the administration. The financial crisis also can be solved with this duration.
2. Second issue is with the tourists visiting the city. When the city center entrance is set to FALSE during a day with high tourism activity, this may cause the tourists to feel unsatisfied since they wont be able to visit the landmarks in the city center such as Duomo. This can also be solved working with the government, scheduling the days with the highest rated tourism activities and trying to balance the $CO_2$ level with additional activities during those days.
3. Third issue is with the clients. This issue can be divided as the following :
    a. They might refuse to use the system.
    b. Even if they use the system, some may try to enter the city center when the availability is set to FALSE.
    c. In fact, all of these issues mentioned in this section are related. The control over the system and the city center can solve the problem mentioned in (3.b). First one can be solved through appopriate advertising and with the collaboration of the city government to help easing the citizens.
4. Fouth issue is related to the development of the technologies. If the tools used for the project is developt to another form, this may cause a problem. The change in the GPS system may harm finding alternative routes or any error

related to this state can be an issue during the development of the project. But this issue can be easily solved by choosing the correct and solid tools and also observing the development of the environment of the project Smart City Advisor.

5. Fifth issue is related to the loss of source code. This is a problem and a threat to all software projects but also can be easily solved by using backup systems and source coudes.
6. The final issue is the development of the project itself, or in another term the schedule of the development. During this process, the team or the staff may get ill, or have some problems in their life. To solve this final issue, the schedule should be made considering the life of the ones working on the project.

## WORKING HOURS

### HASANCAN SAYILAN

- 27.06.2017 : 5h
- 28.06.2017 : 6h
- 29.06.2017 : 5h
- 30.06.2017 : 10h