# Integration Test Plan

# myTaxiService

AA 2015/2016

POLITECNICO
MILANO 1863

*Carlo Broggi*

*Nicola Ghio*

## *Summary*

# 1 Introduction

*In this document we are going to explain our project plan for myTaxiService.*

*First of all we are going to give an estimation of the project size using Function Points (FP) and then we will use this information to estimate the cost and the effort needed.*

*Then we will identify the tasks of the project and their schedule defining the allocation of every member that will take part in the work.*

*Finally we will define the risks for the project along with their relevance and the associated recovery actions.*

## 2 Function Points

*Function points are "a technique to assess the effort needed to design and develop custom software applications" (A.Albrecht IBM Technical Journal) that, basing on a combination of program characteristic is able to help us estimate the Source lines of Code (SLOC). There are 5 typologies of parameters to be taken into account, each of which has 3 levels of complexity, reported in table 1.1.*

| Function Types | Weight | | |
|---|---|---|---|
| | Simple | Medium | Complex |
| N. Inputs | 3 | 4 | 6 |
| N. Outputs | 4 | 5 | 7 |
| N. Inquiry | 3 | 4 | 6 |
| N.Internal logic Files | 7 | 10 | 15 |
| N. External interface file | 5 | 7 | 10 |

Tab. 1.1

## 2.1 Internal Logic File:

*With ILF we mean homogeneous set of data used and managed by the application. We identified the following ILFs for our project:*

| | | |
|---|---|---|
| User | Avarage | 10 |
| Ride | High | 15 |
| Queue | Low | 7 |
| Total: | | 32 |

## 2.2 External Interface File

*With EIF we mean homogeneous set of data used by the application but generated and maintained by other applications. We identified no EIs File.*

## 2.3 External Input:

*With EI we mean elementary operation to elaborate data coming form the external environment. We identified the following EIs for our project:*

| | | |
|---|---|---|
| *Login/Logout* | *Low* | *2 * 3 = 6* |
| *Request taxi* | *High* | *6* |
| *Reserve taxi* | *High* | *6* |
| *Insert/Update user* | *Low* | *2 * 3 = 6* |
| *Insert/Delete/Modify Driver* | *Low* | *3 * 3 = 6* |
| *Accept/Deny Ride* | *Low* | *2 * 3 = 6* |
| *Total:* | | *39* |

## *2.4 External Output*

*With EO we identify elementary operation that generates data for the*

*external environment, it usually includes the elaboration of data from logic files. We identified the following EOs for our project:*

| | | |
|---|---|---|
| *Notify Driver/Costumer* | *Low* | *2 * 3 = 6* |
| *Total:* | | *6* |

## *2.5 External Inquiry*

*With EIN we mean the requests of data from the external elementary operation that involves input and output, without significant elaboration of data from logic files We identified the following EINs for our project.*

| | | |
|---|---|---|
| *User Profile* | *Low* | *3* |
| *Notification Customer/Driver* | *Low* | *2 * 3 = 6* |
| *Queue* | *Average* | *4* |
| *Driver List visualized by Admin* | *Low* | *3* |
| *Driver Search by Admin* | *Average* | *4* |
| *Total:* | | *20* |

## *2.6 From FP to SLOC*

*To pass from FP to SLOC we need to multiply the number of FP that we have just found and multiply it by a constant that depends from the programming language we intend to use, found in the table situated at: http://www.qsm.com/resources/function-point-languages-table*

*For our project we think to use J2EE and therefore we have to multiply by 46, so we have:*

***SLOC** = 46 * FP = 46 * 97 = 4462 more or less **4500** lines*

*Remember that this value is just an estimation.*

# 3 Cocomo II estimation

*This estimation comes through a non linear model that takes in account the*

*characteristics of the product but also of people and process.*

## 3.1 Scale Drivers

*In COCOMO II, some of the most important factors contributing to a project's duration and cost are the Scale Drivers (SD) that determine the exponent used in the Effort Equation.*

*This Factors are:*

1. ***Precedentedness:*** *Reflects the previous experience of the organisation with this type of project. Very low means no previous experience, Extra high means that the organisation is completely familiar with this application domain.*
2. ***Development flexibility:*** *Reflects the degree of flexibility in the development process. Verylow means a prescribed process is used; Extra high means that the client only sets general goals.*
3. ***Architecture/risk resolution:*** *Reflects the extent of risk analysis carried out. Very low means little analysis, Extra high means a complete a thorough risk analysis.*
4. ***Team cohesion:*** *Reflects how well the development team know each other and work together. Very low means very difficult interactions, Extra high means an integrated and effective team with no communication problems.*
5. ***Process maturity:*** *Reflects the process maturity of the organisation. The computation of this value depends on the CMM Maturity Questionnaire but an estimate can be.*

*Here is a table of the SD, taken from:*
*http://csse.usc.edu/csse/research/COCOMOII/cocomo2000.0/CII_modelman2000.0.pdf*

| Scale Factors | Very Low | Low | Normal | High | Very High | Extra High |
|---|---|---|---|---|---|---|
| PREC | Thoroughly unprecedented<br>6.20 | Largely unprecedented<br>4,96 | Somewhat unprecedented<br>3,72 | Generally familiar<br>2,48 | Largely familiar<br>1,24 | Thoroughly familiar<br>0,00 |
| FLEX | Rigorous<br>5.07 | Occasional relaxation<br>4,05 | Some relaxation<br>3,04 | General conformity<br>2,03 | Some conformity<br>1,01 | General goals<br>0,00 |
| RESL | Little (20%)<br>7.07 | Some (40%)<br>5.65 | Often (60%)<br>4.24 | Generally (75%)<br>2.83 | Mostly (90%)<br>1.41 | Full (100%)<br>0.00 |

| | | | | | | |
|---|---|---|---|---|---|---|
| *TEAM* | *Very difficult interactions 5.48* | *Some difficult interactions 4,38* | *Basically cooperative interactions 3,29* | *Largely cooperative 2,19* | *Highly cooperative 1,10* | *Seamless interactions 0,00* |
| *PMAT* | *SW-CMM Level 1 Lower 7.80* | *SW-CMM Level 1 Upper 6.24* | *SW-CMM Level 2 4.68* | *SW-CMM Level 3 3.12* | *SW-CMM Level 4 1.56* | *SW-CMM Level 5 0.00* |

*Tab. 3.1*

*And here is a table of our Scale Drivers:*

| *Scale Driver* | *Factor* | *Value* |
|---|---|---|
| *Precedentness* | *Low* | *4,96* |
| *Development flexbility* | *High* | *2,03* |
| *Risk resolution* | *High* | *2,83* |
| *Team Cohesion* | *Normal* | *3,29* |
| *Process Maturity* | *Normal* | *4,68* |
| *Total:* | | *14,79* |

*Tab. 3.2*

## *3.2 Cost Drivers*

*COCOMO II has 17 cost drivers, such as multiplicative factors that determine the effort required to complete the software project.*

*It defines each of the cost drivers, and the Effort Multiplier associated with each rating.*

*For the complete explaination of those parameters please consult the official cocomo guide here: http://csse.usc.edu/csse/research/COCOMOII/cocomo2000.0/CII_modelman2000.0.pdf*

*The following table contains our estimation of Cost Drivers:*

| *Cost Driver* | *Factor* | *Value* |
|---|---|---|
| *Required Software Reliability* | *Nominal* | *1.00* |

| | | |
|---|---|---|
| *Data Base Size* | *High* | *1.14* |
| *Product Complexity* | *High* | *1.17* |
| *Required Reusability* | *Low* | *0.95* |
| *Documentation match to life-cycle needs* | *Nominal* | *1.00* |
| *Execution Time Constraint* | *Nominal* | *1.00* |
| *Main Storage Constraint* | *Very High* | *1.17* |
| *Platform Volatility* | *Low* | *0.87* |
| *Analyst Capability* | *Nominal* | *1.00* |
| *Programmer Capability* | *Nominal* | *1.00* |
| *Application Experience* | *Nominal* | *1.00* |
| *Platform Experience* | *Nominal* | *1.00* |
| *Language and Tool Experience* | *High* | *0.91* |
| *Personnel continuity* | *Very High* | *0.81* |
| *Usage of Software Tools0* | *Nominal* | *1.00* |
| *Multisite development* | *High* | *0.93* |
| *Required development* | *High* | *1.00* |
| *Total product* | | *0.88415* |

*Tab. 3.3*

## *3.3 Effort Equation*

*Given these parameters we can finally calculate the effort (in Person Months):*

***Effort** = A * 0.88415 * KSLOC^E*

*Where A is equal to 2.94 for COCOMO.2000 .*

*E → exponent derived from Scale Drivers. Is calculated as:*

*B + 0.01 * sum{i} SF[i]*

*So: B + 0.01 * 14,79 = 0.91 + 0.1479 =* ***1.0579 = E***

*Where B is equal to: 0.91 for COCOMO.2000 .*

***KSLOC*** *: estimated lines of code using the FP analysis:* ***4.5***

*With this parameters we can compute the Effort value, that is equal to:*

***Effort*** *= 2.49 * 0.88415 * (4.5 ^ 1.0579) =* ***10.8*** *PM*

*Now we calculate the duration :*

***Duration*** *= 3.67 * (Effort ^ F)*

*F = 0.28 + 0.2 * (E – B) = 0.30958*

***Duration*** *= 3.67 * (10.8 ^ 0.30958) =* ***7.66*** *months*

# *4 Resource Allocation*

*The group is composed by two members: Nicola Ghio and Carlo Broggi. In this chapter we are going to explain how we intend to invest our time in order to complete the project. Before starting with the effective plan it is mandatory to remind that the group is composed by students and, therefore, it will be impossible to invest a whole day for the project unless it is strictly necessary.*

*The calculations above gives us a duration of 7 months and a half with 2 people working on it. This is problematic because we would like to finish this project in less then 5 months (starting at the beginning o October and finishing max at the first of february. For this specific reason we plan to add another person to our team, maybe someone who is more prepared than us in this kind of task (see cap. 5 for further details)*

*So, assuming that we are a group of 3 students, we schedule the following milestones:*

**Documentation:**

- ***Rasd** Submission deadline 6/11*
- ***Design** document submission deadline (4/12). We would, however, prefer to have completed this earlier since we also need to implement the system, let's say that a more informal document is ready for half november.*
- ***Integration test plan:** document submission deadline 21/01. Again, we intend to have a more informal document for us prepared earlier (since, of course, we are not going to start testing 10 days before the final deadline)*

**Implementation:**

- **Writing of Core application**
- **Writing of Mobile Application**
- **Writing of test functions**

# 5 Risks

*In this last section of the document we will analyse the possible risks for our project, their relevance and the associated recovery actions, we will proceed step by step.*

## 5.1 Identify risks

*We think that the possible risks may occur during the project execution:*

1. ***Personnel shortfall and experience**: Since this is our first time realizing such a complex application we think that there is the possibility that the members of our group may not be able to follow the deadlines properly. Moreover it could happen that one member leaves the project for any reasons (he's no more motivated or he has personal necessities) and this would cause a delay.*
2. ***Bad external components**: It could happen that the hardware provided for setting up our system may not be sufficient to efficiently perform all the tasks as we initially thought. This can be caused because the budget assigned for this project was too low or because the system as it has been finally implemented is too computationally expensive. This risks also refer to any kind of problem regarding the internet connection of our system that can be not enough fast for our need.*
3. ***Wrong Functionality**: Although we have already explained in the RASD which functionality our system is supposed to provide there is always the possibility of a misunderstanding between the stakeholders and us.*
4. ***Wrong Interface**: Since this is an application that is going to be use by a moderate amount of customers, the graphic user interface is for sure an important aspect of our system. Therefore there is the possibility that our realisation will not completely satisfy the stakeholders*

## 5.2 Analyze risks and rank

*Here we will give a short analysis of the risks listed above regarding their probability to occur and their possible impact.*

1. *The probability of this risk is high, if any unexpected issue appear the work can suffer a huge delay. Moreover, since this is a small group of students, the probability that one of us leaves the project is also high, for example he may need to concentrate more on university.*
2. *The probability of this risk is quite low, due to modern technologies it is not difficult to find the proper hardware for a reasonable cost moreover, since we are not beginners with algorithms, we think that the probability that the system becomes too computationally expensive is low. Finally, since this is a project for a big city we think that there will probably be a good internet connection available. However the possible impact of this occurrence is really high since it will compromise the very behaviour of our system.*
3. *The probability of this risk is normal. We think that we have done a good job in writing the RASD but the possibility that the customer misunderstood something or did not carefully read the document is not so low. The impact of this risk is normal if the proper precautions are taken if not, (such as if we don't ask to the stakeholders if they agree with our project ideas) this might bring us to rebuild the whole system and this would mean basically that the project failed.*

4. *The probability of this risk is high. Since it is what they directly see of our product, this is a crucial part for our stakeholders. Anyway, if the system has been built correctly, changing the GUI is a relatively easy job, therefore we might consider the impact of this risk as low.*

*We can now rank the risk we found, in order, starting from the most dangerous we have: 1-3-2-4*

## *5.3 Contingency Plan.*

*Here we will list for the most dangerous risks what are our plan to avoid the risk and what we inted to do in case it presents.*

*To try to prevent this risk we may see how the early stage of the development goes. If we'll see that we hardly hit the milestones in time then we will look for another member, maybe someone who has already experience in this kind of work and that can help us improving our skills. In order to prevent delays due to the leaving of a member we might think to involve an extra person right away at the beginning of the development.*

*If the hardware provided is too weak we firstly may try to rent a better one somewhere in order to wait for a good opportunity to buy our own hardware on the market. If we find out that the problem consists in our algorithmic implementation we will perform a proper code inspection in order to find which parts of the code can be lightened. In order to prevent this, during development we will carefully examine each algorithm we will implement by testing it with proper tools (like Jmeter). Finally, regarding the internet connection, we think that if the one provided to us is too low the most logic solution is to find a proper location in the town in which the system could be attached to a faster internet connection which will be easy in a big town in 2016.*

*In order to prevent this risk we plan to personally speak with our stakeholder periodically, mainly before starting to actually develop the project. This will bring us to have clear guidelines to develop the system, given by the customer itself.*

*For redacting and writing this document we spent more or less 6 hours per person.*