POLITECNICO
MILANO 1863

Software Engineering 2 Project

MyTaxiService

PART 5:

Project Plan

Valeria Deciano 858479
Fabio Calabretta 852717

2/2/2016 **v 1.0**

SUMMARY

1. Introduction

1.1 Purpose and Scope

The aim of this paper is to estimate the cost of carrying out of the application MyTaxiService and provides a schedule of activities. Also in this phase it is important to also assign various tasks to team members.

1.2 Revision History

Version 1.0 : actual version

2. Function Points and COCOMO report

2.1 Function Points

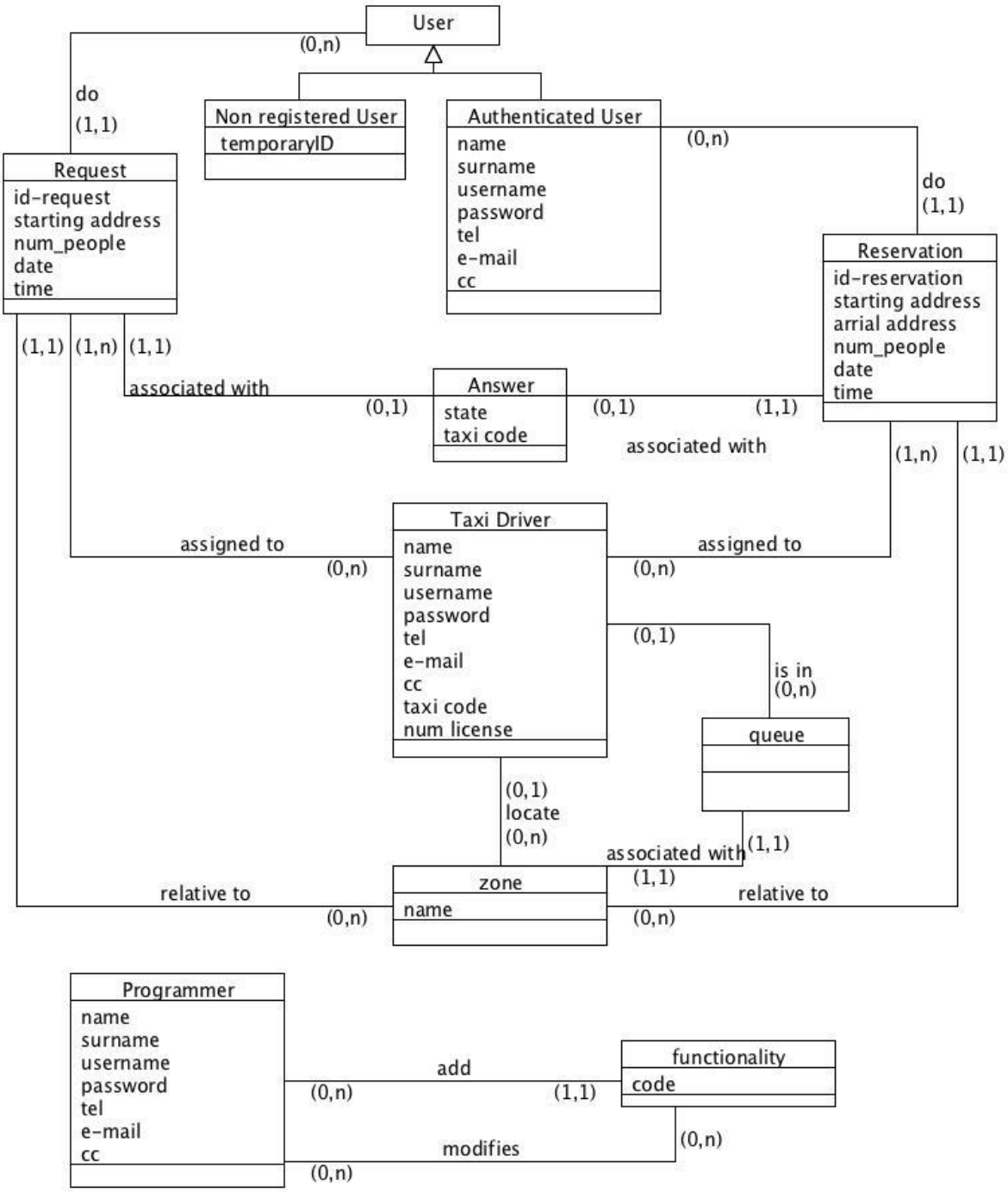
It is a method to estimate the size of the application, which is based on combination of the program characteristics. In particular we want to define:

a. **Internal Logic File (ILF)**
   They are homogeneous set of data used and managed by the application
   Like is shown in the following UML Class Diagram the application need to store information about:
   - Authenticated User
   - Non-Authenticated User
   - Taxi-driver
   - Programmer
   - Request
   - Reservation
   - Answer
   - Functionality
   - Zone
   - Queue

   All of this entity has a smaller number of fields so we can consider them as simple complexity

   The contribution of ILF in the FPs calculation is thus of 10 entities, which are multiplied by the weight, concerning their complexity. In our case we have a simple complexity therefore, as shown by the table the weight is 7.
   ILF contribution = 10 * 7 = **70**

FUNCTION TYPE	WEIGHT		
	Simple	Medium	Complex
N° input	3	4	6
N° output	4	5	7
N° inquiry	3	4	6
N° ILF	7	10	15
N° EIF	5	7	10



b. External Interfaces File (EIF)

They are homogeneous set of data used by application but generated by other application

A GPS system can cooperate with our application, in particular when a taxi driver communicates his availability, the GPS located him in a zone. We can consider this operation like a medium operation because involves 3 entities: Taxi-Driver, Zone and Queue

EIF contribution =1*7= **7**

c. External Input

They are elementary operation to elaborate data coming from the external environment.

The application interacts with a non authenticated user trough:

- Sign-up functionality
- Login functionality
- Enter without authentication functionality

We can consider these operations with a simple complexity because each of them interacts with only one entity.

3*3=9 (->the weight is derived from the table present in the point a.)

The system interactions with the Authenticated User are through these functionalities:

- Do a request
- Do a reservation

These two operations interact with only an entity, so they have a simple complexity.

2*3=6 (->the weight is derived from the table present in the point a.)

The application interacts with a Taxi driver trough:

- Availability functionality

This operation involves 3 entities so we can consider it with a medium complexity.

1*4=4 (->the weight is derived from the table present in the point a.)

The Programmer interactions with the system are through_

- Implementation of new functionality

This operation involves one entity so is consider it with simple complexity.

1*3=3 (->the weight is derived from the table present in the point a.)

External Input Contribution = 9+6+4+3= **22**

d. External Output

They are elementary operations that generate data for the external environment.

The application crates the queue relatives to the available taxi for each zones, and throw its manage the requests and reservations, which came by the user, assigning each of them to a taxy driver (first in the queue). This operation have a high complexity.

External Output Contribution = 1*7 =**7**

e. External Inquiry

They are elementary operation that involves input and output

The application gives the possibility to the authenticated user to see his reservations and we consider this operation of medium complexity

1* 4 =4 (->the weight is derived from the table present in the point a.)

Also a Programmer can choice to do some modification on an existing functionality.

We consider, also this operation with a medium complexity.

1* 4 =4 (->the weight is derived from the table present in the point a.)

External Inquiry contribution = 4+4 =**8**

IEF contribution	70
EIF contribution	7
External Input contribution	22
External Output Contribution	7
External Inquiry contribution	8
FPs	114

We can estimate the number lines of code that are: 114*46= 5244 (LOC)

2.2 COCOMO

COCOMO is a model used to estimate the Effort, the Duration and the Number of people that our project needed.

Scale Drivers:
We need to set these parameters in order to calculate the exponent E used to the estimation of effort

1. **Precedentedness**: Reflects the previous experiences of organization in this kind of project

Very Low	Low	Nominal	High	Very High	Extra High
6.20	4.96	3.72	2.48	1.24	0.00

   It's the first time we uses this kind of approach

2. **Development flexibility**: reflects the degree of flexibility in development process

Very Low	Low	Nominal	High	Very High	Extra High
5.07	4.05	3.04	2.03	1.01	0.00

   The given specification aren't so much detailed

3. **Risk Resolution**: reflects the extent of risk analysis carried out

Very Low	Low	Nominal	High	Very High	Extra High
7.07	5.65	4.24	2.83	1.41	0.00

   We choses high value because we think we have at least 75% of hedging risk

4. **Team Cohesion**: reflects how well the team know each other

Very Low	Low	Nominal	High	Very High	Extra High
5.48	4.38	3.29	2.19	1.10	0.00

   We know well each other and we do other projects together

5. **Process Maturity**: this value depends on CMM Maturity Questionnaire

Very Low	Low	Nominal	High	Very High	Extra High
7.80	6.24	4.68	3.12	1.56	0.00

We choses to set this with an intermediate value

Scale Drivers	**Factor**	**Value**
Precedentedness	Low	4.96
Development Flexibility	High	2.03
Risk Resolution	High	2.83
Team Cohesion	Very High	1.10
Process Maturity	High	3.12
Total: (sum)		14.04

Cost Driver: We need to set these parameters in order to calculate the EAF used to the estimation of effort

1. **Required Software Reliability**

Very Low	Low	Nominal	High	Very High	Extra High
0.82	0.92	1.00	1.10	1.26	n/a

It's possible that same requests or reservations are lost

2. **Data Base Size**

Very Low	Low	Nominal	High	Very High	Extra High
n/a	0.90	1.00	1.14	1.28	n/a

We think that a database with a medium size is enough for our application

3. **Product Complexity**

Very Low	Low	Nominal	High	Very High	Extra High
0.73	0.87	1.00	1.17	1.34	1.74

According to the next COCOMO II COMPLEX rooting scale

4. **Required Reusability**

Very Low	Low	Nominal	High	Very High	Extra High
n/a	0.95	1.00	1.07	1.15	1.24

We have not focused too much on modularity but we believe we have still reached an average level.

5. Documentation match to life-cycle need

Very Low	Low	Nominal	High	Very High	Extra High
0.81	0.91	1.00	1.11	1.23	n/a

All system aspects are described in RASD and DD document, but there aren't any parts of the document directly related to the system development.

6. Execution Time Constraints

Very Low	Low	Nominal	High	Very High	Extra High
n/a	n/a	1.00	1.11	1.29	1.63

No relevant

7. Main Storage Constraints

Very Low	Low	Nominal	High	Very High	Extra High
n/a	0.90	1.00	1.14	1.28	n/a

No relevant

8. Platform Volatility

Very Low	Low	Nominal	High	Very High	Extra High
n/a	0.87	1.00	1.15	1.30	n/a

The platform doesn't often change

9. Analyst Capability

Very Low	Low	Nominal	High	Very High	Extra High
1.42	1.19	1.00	0.85	0.71	n/a

The requirements and scenarios are studied in detail even if we cannot guarantee that the implementation of the requirements is correct

10. Programmer Capability

Very Low	Low	Nominal	High	Very High	Extra High
1.34	1.15	1.00	0.88	0.76	n/a

Reflects our experiences

11. Application Experience

Very Low	Low	Nominal	High	Very High	Extra High
1.22	1.10	1.00	0.88	0.81	n/a

We did not experience longer than six months because it's the first experience

12. Platform Experience

Very Low	Low	Nominal	High	Very High	Extra High
1.19	1.09	1.00	0.91	0.85	n/a

We have experiences in the past year

13. Language and tool experience

Very Low	Low	Nominal	High	Very High	Extra High
1.20	1.09	1.00	0.91	0.84	n/a

We have experiences in the past year

14. Personnel Continuity

Very Low	Low	Nominal	High	Very High	Extra High
1.29	1.12	1.00	0.90	0.81	

The time available is less than a half-year

15. Usage of Software tools

Very Low	Low	Nominal	High	Very High	Extra High
1.17	1.09	1.00	0.90	0.78	n/a

We don't use many tools

16. Multi site development

Very Low	Low	Nominal	High	Very High	Extra High
1.22	1.09	1.00	0.93	0.86	0.80

We use 2 platforms: Web and Mobile

17. Required development schedule

Very Low	Low	Nominal	High	Very High	Extra High
1.43	1.14	1.00	1.00	1.00	n/a

Effort well distributed on available time

Estimation of Effort, Duration and Number of people:

Effort := $A*EAF*KSLOC^E$

A → 2.94

EAF: product of all cost drivers = 1.007595877

E: derived from Scale Driver

B+0.01 * sum{All Scale Drivers}

B → 0.91

E=1.0504

KSLOC = 5.244 derived from FPs

Effort =**16.88757903 PM**

Duration =3.67 * effort ^{F}=**8.767081163 months**

F= 0.28+0.2*(E-B)=0.30808

Number Of People=Effort/Duration =1.9 -> **2 people**

By our estimates, we found a too high value for the Duration, that should be about 4 months.
This value could be wrong for several reasons:
- Estimates may not be suitable to the problem
- The values attributed to the drivers may not be fair
- The model does not fit perfectly with our situations

3. Tasks schedule and resources allocation

Tasks	From	To	Member1	Member2
RASD				
1. Overall Description	24/10/2015	24/10/2015	2h	
2. Requirements	25/10/2015	31/10/2015	6h	
3. Scenarios	24/10/2015	25/10/2015		3h
4. UML models	1/11/2015	6/11/2015	5h	
5. Alloy	29/10/2015	4/11/2015		6h
DESIGN DOCUMENT				
1. Architectural design	20/11/2015	26/11/2015	4h	2h
2. Algorithm design	27/11/2015	29/11/2015		1h
3. User interface design			Made in RASD	
4. Requirements traceability	1/12/2015	4/12/2015	4h	
CODE INSPECTION				
1. Functional role of assigned set of classes	18/12/2015	20/12/2015		3h
2. List of issues found by applying the checklist	21/12/2015	23/12/2015		5h
3. Any other problem you have highlighted	3/12/2015	3/12/2015		1h
TEST PLAN DOCUMENT				
1. Integration strategy	11/01/2016	12/01/2016	2h	
2. Individual steps and test description	13/01/2016	16/01/2016	3h	
3. Tools and test equipment required	17/01/2016	17/01/2016	0.5h	
4. Program stubs and test data required	20/01/2016	21/01/2016	2h	
PROJECT PLAN				
1. Function Points and COCOMO report	23/01/2016	24/01/2016	3h	
2. Tasks schedule and resources allocation	23/01/2016	24/01/2016		2h
3. Risk for the project	30/01/2016	31/01/2016	1h	0.5h

The color of the boxes reglects the difficulty level of the task:

4. Risks for the project

We want to understand what are the risks of our system and the preventive actions that can be applied to avoid it.

First of all it may be that the customer who wants to make a request, via mobile interface for instance, hasn't network coverage and for this reason he's not be able to use the service. We believe that an application ables to operate even when there is no network coverage is too expensive. We think that at least in the areas with hard coverage circumstances, it can be a solution installing some totems that work just like a mobile phone and are always connected with the system. They are fixed to the same point, thus we can make available to the users a map that shows where the totems are located.

There is another problem in case there is no available taxi at the same time in all zones and users have made one or more requests. Since we do not know if or when taxis will be available, it would be appropriate to communicate this problem to the user and allow him to choose whether to cancel the request or wait for the arrival of the taxi (which it may take a long time)

Another risk for the system could be in case of two requests occur simultaneously from the same user account. In this case the system may not take into consideration neither of the two reservations. To avoid this, the system should provide to verify that the same user could access to the service only by one device at a time.