



Politecnico di Milano, A.A. 2015/2016

Software Engineering 2: My Taxi Service  
**Project Plan Document**

Belotti Nicola 793419  
Chioso Emanuele 791621  
Colombo Andrea 853381

February 1, 2016

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Function points</b>	<b>3</b>
2.1	Internal Logic files . . . . .	3
2.2	External interface files . . . . .	4
2.3	External inputs . . . . .	4
2.4	External outputs . . . . .	4
2.5	External Inquiry . . . . .	5
2.6	Summary . . . . .	5
2.7	Source lines of code . . . . .	5
<b>3</b>	<b>Cocomo</b>	<b>6</b>
3.1	Effort estimation . . . . .	6
3.2	Time estimation . . . . .	6
<b>4</b>	<b>Tasks</b>	<b>7</b>
<b>5</b>	<b>Risks</b>	<b>8</b>
<b>6</b>	<b>Appendix</b>	<b>9</b>

# 1 Introduction

## 2 Function points

The Function Point estimation approach is based on the amount of functionalities in a software and their complexity. We will now provide a detailed description on the function points related to our application.

There are five estimators to take in account:

- Internal logic files
- External interface files
- External inputs
- External outputs
- External inquiries

This table defines the weights values that we've to use to perform the FP value.

<b>Table 3. UFP Complexity Weights</b>			
<b>Function Type</b>	<b>Complexity-Weight</b>		
	<b>Low</b>	<b>Average</b>	<b>High</b>
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

Figure 1: Weights for all the different estimators

We will now proceed to identify all the elements and assign a value to them

### 2.1 Internal Logic files

An internal logic files is simply a data structure used in the application. The ILFs of our system are the following:

- User
- Passenger
- TaxiDriver

- Request
- Ride

User, passenger and taxi drivers can be assumed to be simple data structure. Request and ride are a little more complex so we'll treat them as average complexity.

## 2.2 External interface files

External Interface Files are a homogeneous set of data used by the application but generated by other applications.

The only EIF of our system is the interface of a map service we use for navigation. It can be considered as average complexity.

## 2.3 External inputs

External Inputs are operations to elaborate data coming from the external environment. The EIs of our system are:

- Registration
- Login
- Simple request
- Taxi reservation
- Driver availability

Login, request, reservation and driver availability are inputs with low complexity. Registration is more complex (average complexity) since it involves a bigger amount of data and components.

## 2.4 External outputs

External Output is an elementary operation that generates data for the external environment. The EOs of our system are:

- Notifications to users
- Notifications to taxi drivers

Both of these outputs can be considered of low complexity.

## 2.5 External Inquiry

External Inquiry is as an operation that involves input and output, without elaboration of data. The EQs of the system are:

- Taxi driver ride request
- Ride sharing

Ride request to taxi driver is simple (low complexity). On the other hand, ride sharing requires a lot of operations since it can be considered of high complexity.

## 2.6 Summary

**Internal logic files score:** 3 low complexity + 2 average complexity:  $3 * 7 + 2 * 10 = 41$

**External onterface files score:** 1 medium complexity: 7

**External inputs score:** 4 low complexity + 1 average complexity:  $4 * 3 + 1 * 4 = 16$

**External outputs score:** 2 low complexity: 4

**External inquiries:** 1 low complexity + 1 high complexity:  $1 * 3 + 1 * 6 = 9$

The total is of  $41 + 7 + 16 + 4 + 9 = 77$

## 2.7 Source lines of code

The conversion multiplicator from function points to SLOC is 53 for the Java language: hence we have  $77 * 53 = 4081$  lines of code.

### 3 Cocomo

We use cocomo model to provide an approximate cost estimation on the project and the time needed to develop the software.

#### 3.1 Effort estimation

COCOMO formula for calculating the effort is  $ffort = 2.94 * EAF * KSLOC^E$   
Where:

- KSLOC is kilo source lines of code calculated in the previous section.
- EAF: effort adjustment factor derived from cost drivers
- E: exponent derived from scale drivers
- 2.94 a parameter

We consider our project with all nominal Cost Drivers and Scale Drivers, thus EAF will be equal to 1 and E will be equal to 1.0997. So  $ffort = 2.94 * 1 * 4.081^{1.0997} = 15.3709809$  person-months

#### 3.2 Time estimation

The duration of the project is calculated with the formula:  $Duration = 3.67 * ffort^E$  Where:

- ffort is the effort calculated in the previous point.
- E: exponent derived from scale drivers
- 3.67 a parameter

So  $Duration = 3.67 * 15.3709809^{0.3179} = 3.60379248$  months

#### 3.3 Conclusion

Now the we have effort (expressed in person-months) and duration (expressed in months) we can calculate the number of people necessary to develop the software  $N = \frac{ffort}{duration} = \frac{15.3709809}{3.60379248} = 4.26522365 \approx 5people$

## 4 Tasks



## 5 Risks

## 6 Appendix