

Software Engineering 2 “myTaxiService”

Project Plan Version 0.1

1/21/2016

Politecnico di Milano A.A. 2015-2016

Milica Jovanovic (mat. 835953);Pavle Vidanovic (mat. 854472)

Contents

[1 Introduction 3](#_Toc441157752)

[1.1 Revision History 3](#_Toc441157753)

[1.2 Purpose and Scope 3](#_Toc441157754)

[1.3 Definitions and Abbreviations 3](#_Toc441157755)

[1.4 Reference Documents 4](#_Toc441157756)

[1.5 Document Overview 4](#_Toc441157757)

[2 Function Points 5](#_Toc441157758)

[2.1 Brief Introduction 5](#_Toc441157759)

[2.2 FP Estimation 5](#_Toc441157760)

[2.2.1 Internal Logic Files 5](#_Toc441157761)

[2.2.2 External Logic Files 6](#_Toc441157762)

[2.2.3 External Inputs 6](#_Toc441157763)

[2.2.4 External Inquiries 7](#_Toc441157764)

[2.2.5 External Outputs 7](#_Toc441157765)

[2.2.6 Resuming 7](#_Toc441157766)

[2.3 Evaluation of Estimation 7](#_Toc441157767)

[3 Effort Estimation COCOMO II 7](#_Toc441157768)

[3.1 Brief Introduction 7](#_Toc441157769)

[3.2 Scale Drivers 7](#_Toc441157770)

[3.3 Cost Drivers 7](#_Toc441157771)

[3.4 Effort Equation 7](#_Toc441157772)

[3.5 Schedule Estimation 7](#_Toc441157773)

[4 Task Allocation 7](#_Toc441157774)

[5 Resources Allocation 7](#_Toc441157775)

[6 Risk Management 7](#_Toc441157776)

[7 References 7](#_Toc441157777)

# Introduction

## Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Data | Authors | Summary |
| 0.1 | 21/01/2016 | Pavle Vidanovic  Milica Jovanovic | Initial Draft |

## Purpose and Scope

The purpose of this document is to define plan for testing, integration testing and verifying that system development during the project complies with the requirements of Requirement document and Design Document. This document also presents test results in order to determinate if the application meets predetermined requirements and functionalities.

The aim of this project is to develop and implement myTaxiService, an application similar to Uber, which makes the process of assigning an available taxi vehicle to possible passengers.

The developed system should allow new users to register. Users, once logged in, should be able to:

* request a taxi
* reserve a taxi
* cancel a ride
* check taxi availability around him
* receive a confirmation with information about the assigned vehicle and ETA once taxi is requested
* create/maintain user profile
* report a taxi driver

The developed system should allow new taxi drivers to register. Drivers, once logged in, should be able to:

* inform the system about their availability
* confirm/decline that they are going to take care of a certain call
* create/maintain taxi driver profile
* report a passenger

The system should keep information about new arrived requests, as well as the confirmed rides. A ride should have and id number, information about the passenger that requested the ride, as well as the code of the assigned vehicle and ETA. System should also keep information about taxi queues connected to particular zone of the city and ensure fair management of the queues. Developed system should keep information about the list of reservations made by passengers, such as id number of the reservation, information about the passenger that made the reservation and the time of reservation and time of the ride.

## Definitions and Abbreviations

|  |  |
| --- | --- |
| *FP* | FunctionPoints |
| *COCOMO* | Constructive Cost Model |
| *PM* | Person-Month |
| *UFP* | |  | | --- | | Un-adjusted Function-Points | |

## Reference Documents

* RASD - RASD myTaxiService - final v2.0
* DD - DD myTaxiService - final
* Assignment 5 – Project Plan
* Example of usage FP and COCOMO for Assignment 5

## Document Overview

The document is essentially structured in six parts:

* Chapter 1: Introduction, gives description of document and some basic information about the software
* Chapter 2: Integration Strategy, gives an overview of entry criteria for the integrating components and how the elements will be integrated as well as used testing strategy and  sequences of component/function integration
* Chapter 3: Individual Steps and Test Description, description of type of tests for verifying elements defined in one step, verifying the results are as expected
* Chapter 4: Tool and Test Equipment Required, overview on tools and equipment used to support integration test
* Chapter 5: Program Stubs and Test Data Required, gives an overview of how the requirements defined in RASD map into the design elements defined in DD.
* Chapter 6: References

# Function Points

## Brief Introduction

The Function Point estimation approach is based on the amount of functionalities in software and their complexity. FP estimators are useful since they are based on information that is available early in the project life cycle. To perform this estimation we've based our parameters on the following tables, taken from COCOMO II.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Function Types |  | Weight | | |
|  |  | Low | Average | High |
| N.Inputs |  | 3 | 4 | 6 |
| N.Outputs |  | 4 | 5 | 7 |
| N.Inquiry |  | 3 | 4 | 6 |
| N.ILF |  | 7 | 10 | 15 |
| N.EIF |  | 5 | 7 | 10 |

Table 1. UFP – Weighting Function Points

After calculating the UFP we use this data into the COCOMO II effort equation. Following the COCOMO II algorithm at the end we get the PM effort.

## FP Estimation

### Internal Logic Files

The myTaxiService system stores information about:

* Users
* Drivers
* Reservations
* Reports
* Requests
* Zones
* Admin
* Taxi vehicles

System stores information about users using myTaxiService system. It stores id, name, surname, phone, email, password, image path. Considering the fields stored about the user we will assume its complexity is ***Average***. Information stored about the driver includes user’s fields as well as drive license number and its availability indicator. Taking this into account we will assume that its complexity is as well ***Average***. Information stored about requests is id, ETA, origin, valid; its complexity is ***Low***. Information stored about reservations is the fields of request object with additional four fields; time of reservation, time of ride, destination and description. We will assume complexity of reservation object is Average. Report object has four fields; id, description, user id and driver id. We will assume it’s complexity as Low. Information stored about zone is id, radius, longitude and latitude; its complexity is as well Low. Admin object has two fields, username and password; its complexity is Low. Taxi vehicle has three parameters id, type and number of seats; thus its complexity is Low.

Taking into consideration above description our system handles with three Average and five Low complexity objects.

|  |  |  |
| --- | --- | --- |
| ILF | Complexity | FP |
| User | Average | 10 |
| Driver | Average | 10 |
| Request | Low | 7 |
| Reservation | Average | 10 |
| Report | Low | 7 |
| Taxi Vehicle | Low | 7 |
| Zone | Low | 7 |
| Admin | Low | 7 |
| Total: | | **65** |

### External Logic Files

myTaxiService system user three external APIs: GoogleMaps API, GooglePlaces API and Gmail API. Invoking GoogleMaps API happens often, at least once when user requests a taxi, or when checking on the map if there are taxi vehicles available near him. Google Places API is invoked when user request or reserve a taxi; when he specify his origin or destination address. Thus Google Places API is invoked often. Gmail API is invoked only once per user, when he creates account for myTaxiService.

|  |  |  |
| --- | --- | --- |
| ELF | Complexity | FP |
| GoogleMaps API | High | 10 |
| Google Places API | Average | 7 |
| Gmail | Low | 5 |
| Total: | | **22** |

### External Inputs

OPIS!!!

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Clients |  | Function | Complexity | FP |
| Guest |  | signUp()  signIn() | Low  Low | 3  3 |
| User |  | makeRequest()  makeReservation()  report()  manageProfile()  checkTaxisAvailable()  checkReservation()  cancelRide() | High - ? Inquiry  High - ? Inquiry  Low  Average  Low - ? Inquiry  Low - ? Inquiry  Low | 3  4  3 |
| Driver |  | confirmDeclineRide()  setAvailable()  manageProfile()  report()  cancelRide()  checkRides() | Average - ? Inquiry  Low  Average  Low  Low  Average - ? Inquiry | 3  4  3  3 |
| Admin |  | banUser()  viewReports()  signIn() | Average  Low - ? Inquiry  Low | 4  3 |
| Total: | | | | **36** |

### External Inquiries

OPIS!!!

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Clients |  | Function | Complexity | FP |
| User |  | makeRequest()  makeReservation()  checkTaxisAvailable()  checkReservation() | High  High  Low  Low | 6  6  3  3 |
| Driver |  | confirmDeclineRide()  checkRides() | Average  Average | 4  4 |
| Admin |  | viewReports() | Low | 3 |
| Total: | | | | **29** |

### External Outputs

### Resuming

## Evaluation of Estimation

# Effort Estimation COCOMO II

## Brief Introduction

## Scale Drivers

## Cost Drivers

## Effort Equation

## Schedule Estimation

# Task Allocation

# Resources Allocation

# Risk Management

# References