Release date 01-02-2016

Software Engineering 2: MyTaxiService

Project Plan

V1.0

Dimitar Anastasovski, Marco Colombo

Contents

Chapter 1

Introduction

1.1 Purpose

This project plan documents the project planning process and consists of the following basic tasks:

- Defining the sequence of tasks to be performed
- Identifying all deliverables associated with the project
- Defining the dependency relationship between tasks
- Estimating the resources required to perform each task
- Scheduling all tasks to be performed
- Identifying the known project risks
- Defining the process ensuring quality of the project product
- Defining the process specifying and controlling requirements

1.2 Definitions

1. FP: Function Points
2. COCOMO: Constructive Cost Model

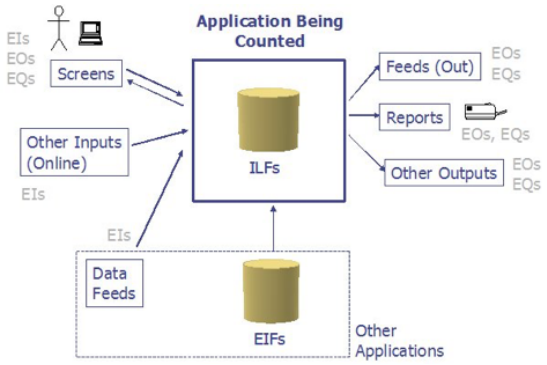
Chapter 2

Functional Points Approach

2.1 Introduction

Frequently the term end user or user is used without specifying what is meant. In this case, the user is a sophisticated user. Someone that would understand the system from a functional perspective — more than likely someone that would provide requirements or does acceptance testing.

Since Function Points measures systems from a functional perspective they are independent of technology. Regardless of language, development method, or hardware platform used, the number of function points for a system will remain constant. The only variable is the amount of effort needed to deliver a given set of function points; therefore, Function Point Analysis can be used to determine whether a tool, an environment, a language is more productive compared with others within an organization or among organizations. This is a critical point and one of the greatest values of Function Point Analysis.



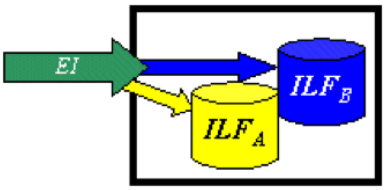
An ILF is a user-identifiable group of logically related data or control information maintained within the boundary of the application. The primary intent of an ILF is to hold data maintained through one or more elementary processes of the

application being counted." Furthermore, for data or control information to be counted as an ILF, both of the following IFPUG counting rules must also apply:

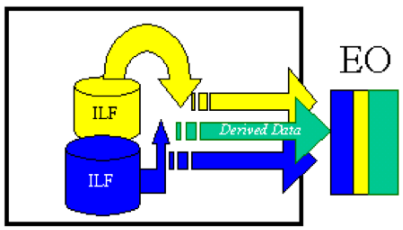
1. The group of data or control information is logical and user identifiable.
2. The group of data is maintained through an elementary process within the application boundary being counted.

An external interface file (EIF) is a user identifiable group of logically related data or control information referenced by the application, but maintained within the boundary of another application. The primary intent of an EIF is to hold data referenced through one or more elementary processes within the boundary of the application counted. This means an EIF counted for an application must be in an ILF in another application

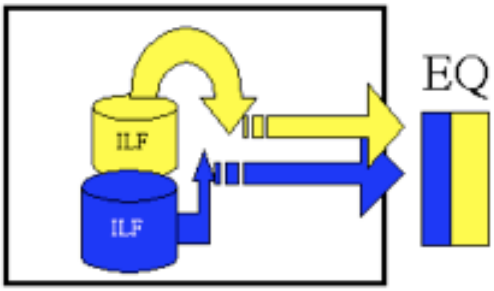
An external input (EI) is an elementary process that processes data or control information that comes from outside the application boundary. The primary intent of an EI is to maintain one or more ILFs and/or to alter the behavior of the system.



An external output (EO) is an elementary process that sends data or control information outside the application boundary. The primary intent of an external output is to present information to a user through processing logic other than, or in addition to, the retrieval of data or control information . The processing logic must contain at least one mathematical formula or calculation, create derived data maintain one or more ILFs or alter the behavior of the system.



An external inquiry (EQ) is an elementary process that sends data or control information outside the application boundary. The primary intent of an external inquiry is to present information to a user through the retrieval of data or control information from an ILF of EIF. The processing logic contains no mathematical formulas or calculations, and creates no derived data. No ILF is maintained during the processing, nor is the behavior of the system altered.



2.2 Function types

- **Internal Logical File (ILF):** homogeneous set of data used and managed by the application
- **External Interface File (EIF):** homogeneous set of data used by the application but generated and maintained by other applications
- **External Input:** Elementary operation to elaborate data coming form the external environment
- **External Output:** Elementary operation that generates data for the external environment. It usually includes the elaboration of data from logic files
- **External Inquiry:** Elementary operation that involves input and output Without significant elaboration of data from logic files

The following table outline the number of Functional Point based on functionality and relative complexity:

2.3 Function points calculation

We perform the calculation step by step:

Function Type	Complexity		
	Simple	Medium	Complex
Internal Logic File	7	10	15
External Logic File	5	7	10
External Input	3	4	6
External Output	4	5	7
External Inquiry	3	4	6

- **Internal Logic Files (ILFs):** The application has ILFs, used to store and manage informations about users (Passenger or Taxi Driver), reservation, call and notification. For notification, call and information about users we can use a simple weight for the not so difficult operation involved. Different case for the reservation that we adopt a complex weight.
  So we have: $3 \times 7 + 1 \times 15 =$ **36 FPs**

- **External Logic Files (ELFs):** The application use two ELFs. One to manage the GPS, using the GoogleMap APIs and the second is the phone company for manage the calls. These are simple operation so in this case we have: $2 \times 5 =$ **10 FPs**

- **External Input:** The application uses these following inputs:

  - Login/Logout: simple operations so we have $2 \times 3 =$ **6 FPs**
  - Sign In: This is a simple operation too so $1 \times 3 =$ **3 FPs**
  - Create a taxi reservation: This is a complex operation because involved more than one entity, so $1 \times 6 =$ **6 FPs**
  - Change a taxi reservation: This is a complex operation because involved more than one entity, so $1 \times 6 =$ **6 FPs**
  - Delete a taxi reservation: This is a complex operation because involved more than one entity, so $1 \times 6 =$ **6 FPs**
  - Call a taxi: This is a simple operation, so $1 \times 3 =$ **3 FPs**
  - Insert taxi availability: This is a simple operation, so $1 \times 3 =$ **3 FPs**

- **External Output:** The application produces the following outputs:

  - Send a notification to the taxi driver for a desired reservation
  - Send a notification to the passenger to confirm the reservation
  - Call the taxi driver for a ride

The first two outputs can considers as medium weight, but the call we can consider is with a simple weight, so we have: $2 \times 5 + 1 \times 4 =$ **14 FPs**

- **External Inquiry:** The application allows user to:
  - View the available taxi for a reservation
  - Change the status of a taxi driver

  The first can consider as a medium weight, the second as a simple weight. $1 \times 4 + 1 \times 3 =$ **7 FPs**

In conclusion we have: $36 + 10 + 6 + 3 + 6 + 6 + 6 + 3 + 3 + 14 + 7 =$**100 FPs**

Chapter 3

COCOMO II Approach

3.1 Scale Drivers

These values are evaluated according to the following table:

Scale Factors	Very Low	Low	Nominal	High	Very High	Extra High
PREC	thorouhly unprecedented	largely unprecedented	somewhat unprecedented	generally familiar	largely familiar	thoroughly familiar
SF:	6.20	4.96	3.72	2.48	1.24	0.00
FLEX	rigorous	occasional relaxation	some relaxation	general conformity	some conformity	general goals
SF:	5.07	4.05	3.04	2.03	1.01	0.00
RESL	little {20%}	some {40%}	often {60%}	generally {75%}	mostly {90%}	full {100%}
SF:	7.07	5.65	4.24	2.83	1.41	0.00
TEAM	very difficult interactions	some difficult interactions	basically cooperative interactions	largely cooperative	highly cooperative	seamless interactions
SF:	5.46	4.38	3.29	2.19	1.10	0.00
PMAT	SW-CMM Level 1 Lower	SW-CMM Level 1 Upper	SW-CMM Level 2	SW-CMM Level 3	SW-CMM Level 4	SW-CMM Level 5
SF:	7.80	6.24	4.68	3.12	1.56	0.00

- **Precedentedness:** It reflects the teams previous experience with this kind of projects. Since for this team it was the first experience using this frame-

work and these development methodologies, such as J2EE and Android, this value will be low.

- **Development flexibility:** It reflects the degree of flexibility in the development process. The teachers and teaching assistants constructed the assignments giving the general specifications without going too much in details, making this project development very flexible, so, for this reason this value is going to be set to high - general conformity.

- **Risk resolution:** Reflects the extent of risk analysis carried out. This value will be high, considering this project.

- **Team cohesion:** Reflects how well the development team know each other and work together. Let?s assume that this is team's first project and that they didn't know each other previously. Also, let's assume that there are synchronization problems, such as different academic assignments during project for each of the members. So, this driver will be high - largely cooperative.

- **Process maturity:** This was evaluated around the 18 Key Process Area (KPAs) in the SEI Capability Model.

  There are five levels of process maturity, level 1 (lowest half) to level 5 (highest). The CMM specifies "what" should be in the software process rather than "when" or "for how long".

  The CMM level 1 is for organizations that don't focus on processes or documenting lessons learned. The CMM level 1 is for organizations that have implemented most of the requirements that would satisfy CMM level 2. In CMM's published definition, level 1 (lower half) and (Upper half) are grouped into level 1.

3.2 Risk

A risk is an event or condition that, if it occurs, could have a positive or negative effect on a project?s objectives. Risk Management is the process of identifying, assessing, responding to, monitoring and controlling, and reporting risks. This Risk Management Plan defines how risks associated with the MyTaxiService project will be identified, analyzed, and managed. It outlines how risk management activities will be performed, recorded, and monitored throughout the lifecycle of the project and provides templates and practices for recording and prioritizing risks by the Risk Manager and/or Risk Management Team.

3.2.1 Risk Assessment

Risk assessment is the act of determining the probability that a risk will occur and the impact that event would have, should it occur. This is basically a ?cause and effect? analysis. The ?cause? is the event that might occur, while the ?effect? is the potential impact to a project, should the event occur. Assessment of a risk involves two factors. First is the probability which is the measure of certainty that an event, or risk, will occur. This can be measured in a number of ways, but for the MyTaxiService project will be assigned a probability as defined in the table below.

Probability of Occurrences		
Definition	**Meaning**	**Value**
Frequent	- Occurs frequently - Will be continuosly experienced unless action is taken to change event	5
Likely	- Occur less frequently if process is corrected - Issues identified with minimal audit activity - Process performance failures evident to trained auditors or regulators	4
Occasional	- Occurs sporadically -Potential issues discovered during focused review	3
Seldom	- Unlikely to occur - Minimal issue identification during focused review	2
Improbable	Highly unlikely to occur	1

Risk	Probability	Effects	Strategy
Requirements change	Moderate	Serious	Derive traceability information to access requirements to change impact; work on software flexibility
Requirements have compliance issues (conflicts) with law and legal regulations - as this software deals with personal data which need to be authentic)	Low	Serious	Mention in the beginning what could be legal issues related to requirements and try to reformulate them in order to avoid conflicts with law and legal regulations
Requirements fail to align with strategy - Requirements conflict with the firm's strategy (fair taxi managment)	Moderate	Serious	Involve stakeholders as much as possible during requirements analysis and formulation
Design lacks flexibility - A poor design makes change requests difficult and costly.	Moderate	Serious	Work on training and improving less experienced team members but, in general, let only the most experienced members do this job, as design is critical part of the project and is heavily base on experiene with similar projects.
Technology components aren't scalable - Components that can't be scaled to meet performance demands	High	Serious	Consider different technlogies (as alternatives) and start performance testing as soon as possible
Technology components aren't compliant with standards and best practites - Non-standard components that violate best practices	Moderate	Marginal	Use technology that is already has been approved as suitable for similar systems.

Project team lack authority to complete	Low	Catastrophic	Make clear in the beginning what is the authority needed (access to government databases in this case) to complete the project and work on achieving it as early as possible
Database perfomance	Moderate	Serious	Consider using higher perfomance database or cloud
Junior member not able to work on mobile app	Moderate	Catastrophic	Consider switching other team member to this task(either expert or project manager who is not so experienced in mobile apps development but is experienced and educated enough as an engineer to accept such challenge)

3.3 COCOMO II calculation

To pass from FP(Function Points) to SLOC(Source Lines of Code) we use an average conversion factor of 46 lines of code for each function point:

$$100FPs * 46 = 4600SLOC$$

Consider a project with all "Nominal" (i.e., normal) Cost Drivers and Scale Drivers would have an EAF of 1.00 and exponent, E, of 1.0997.

EAF: Effort Adjustment Factor derived from Cost Drivers.
E:Exponent derived from Scale Drivers.

The formula to calculate the effort is the following:

$$effort = 2.94 * EAF * (KSLOC)^E$$

so we have

$$effort = 2.94 * (1.0) * (4.6)^{1.0997} = 15.74 \text{ Person-Months}$$

Now we try to calculate the schedule (duration) of project in month with the following formula

$$Duration = 3.67 * (effort)^E$$

we use a new value for E, that is 0.3179 (calculated from new scale drivers for schedule). So we have

$$Duration = 3.67 * (15.74)^{0.3179} = 8.81 \text{ Months}$$

Now we can estimate the number of people needed to complete the project with the following formula:

$$N_{people} = effort/Duration$$

$$N_{people} = 15.74/8.81 = 1.78 \rightarrow 2 \text{ people}$$