

# Project estimation with Use Case Points using Enterprise Architect (EA)

*Step-by-Step Guide: How to use Enterprise Architect (EA) as a CASE tool to facilitate calculating Use Case Points for software projects*

Abdollah Zawari  
skilledDeveloper@gmail.com  
<http://csharptips.wordpress.com>

## Introduction

Project estimation is one of the most challenging duties of project managers. Among various estimation methods, Use Case Points is the most reliable method. If you are not familiar with UCP please read this [article](#).

As you may know, Use Case Points method involves different factors and needs calculation. For most software project managers it seems difficult and time consuming to use it manually.

In this article I will show you how to estimate a software project using [Sparx Enterprise Architect \(EA\)](#) which is a famous [CASE](#) tool. (I used EA 7.5 in this article)

Based on Rational Unified Process (RUP), In order to estimate software projects duration, you need to recognize the project features and requirements first. This leads you to Use Cases that are cores of a software project analysis model and you cannot estimate any software project without recognizing its Use Cases. Requirements, Features and Use Cases recognition are usually done in the first phase of RUP, Inception.

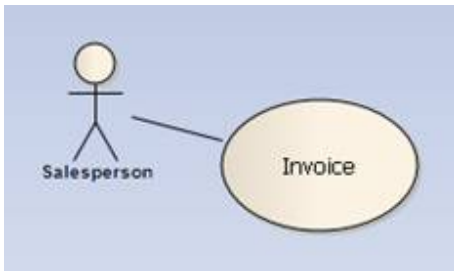
## Background

The main reference of this guide is the article written by [RoyClem](#), which describes Use Case Points method in detail. I suggest you to read that article first:

[Project Estimation with Use Case Points](#)

## Getting Started

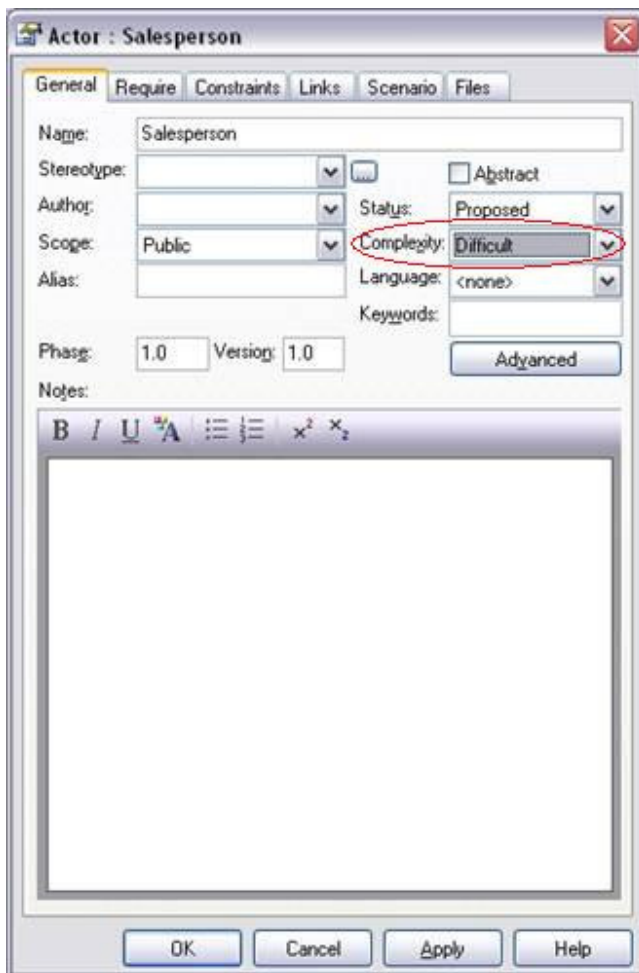
First of all you need to have EA installed and know how to model Use Cases. We assume that you have recognized the project Use Cases and actors and have Use Case model ready in EA.



Now follow the steps:

## Step 1 - Weigh all actors:

Each actor can be determined as easy, medium or difficult. Double click on the actor to set its complexity:



The following table shows how to determine an actor's complexity:

Actor Type	Description
Easy	The actor represents another system with a defined Application Programming Interface (API)
Medium	The actor represents another system interacting through a protocol-driver Interface (TCP/IP)
Difficult	The actor is a person interacting via a Graphical User Interface (GUI)

### Step 2 - Weigh all Use Cases:

Each Use Case's complexity must be determined by double clicking on it:



The following table helps you determine the Use Case complexity:

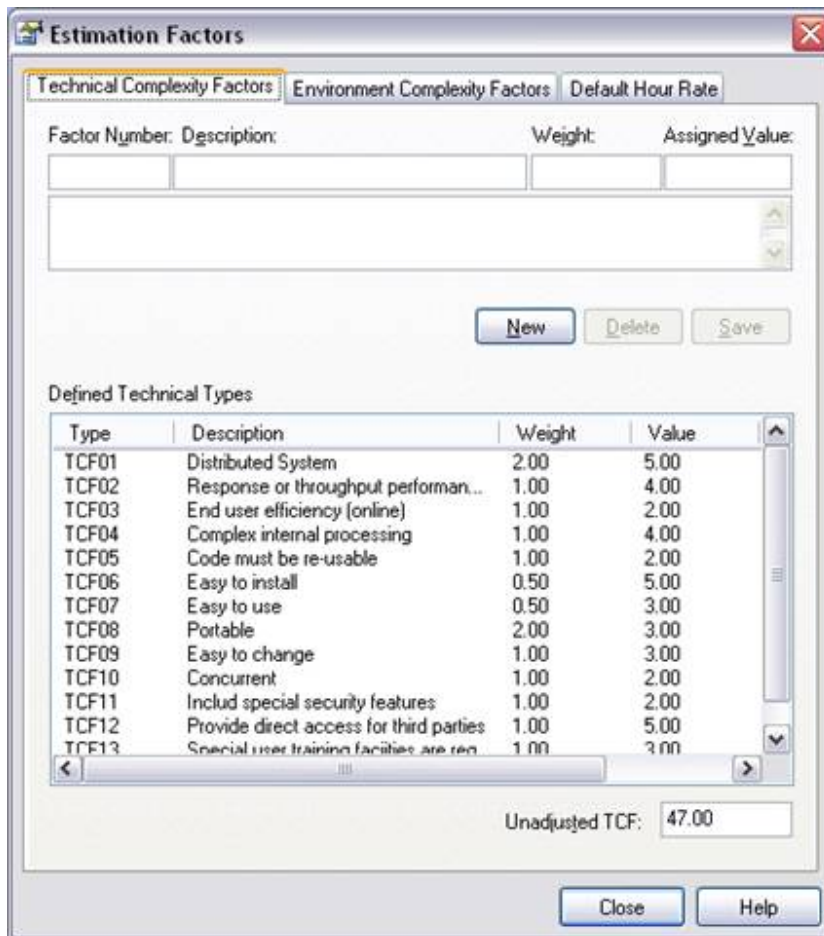
Use Case Type	Description
Simple	<ul style="list-style-type: none"> <li>• Touches only a single database entity</li> <li>• Its success scenario has 3 steps or less</li> <li>• Involves less than 5 classes.</li> </ul>
Average	<ul style="list-style-type: none"> <li>• Touches 2 or more database entities</li> <li>• Between 4-7 steps</li> <li>• Involves 5-10 classes.</li> </ul>
Complex	<ul style="list-style-type: none"> <li>• Touches 3 or more database entities</li> <li>• Over seven steps</li> <li>• Involves more than 10 classes.</li> </ul>

### Step 3 - Adjust Technical Factors:

To adjust technical factors, click *Estimation Factors* from *Settings* menu then click on *Technical Complexity Factors* tab.

Considering your own project, you can change the *Weight* and *Value* columns but for most purposes, the only table column requiring adjustment is Value, which indicates the degree of influence a particular factor has on the project. As a suggested gauge, a value of 0 indicates no influence, 3 indicates average influence and 5 indicates strong influence.

You can also add/remove factors regarding your project conditions:



The following table helps you find out the values of technical factors:

Technical Factor	Description	Weight	Explanation
T1	Distributed System	2	Informs whether distributed data processing is required in the system.
T2	Performance	1	Determines the system's efficiency with regard to response time to events, flow, etc.
T3	End User Efficiency	1	Defines efficiency for the final user in the context of his or her perception.
T4	Complex Internal Processing	1	Determines whether complicated operations related to data processing, use of advanced algorithms are required.
T5	Reusability	1	Informs whether elements or the code of the generated system will be used again.

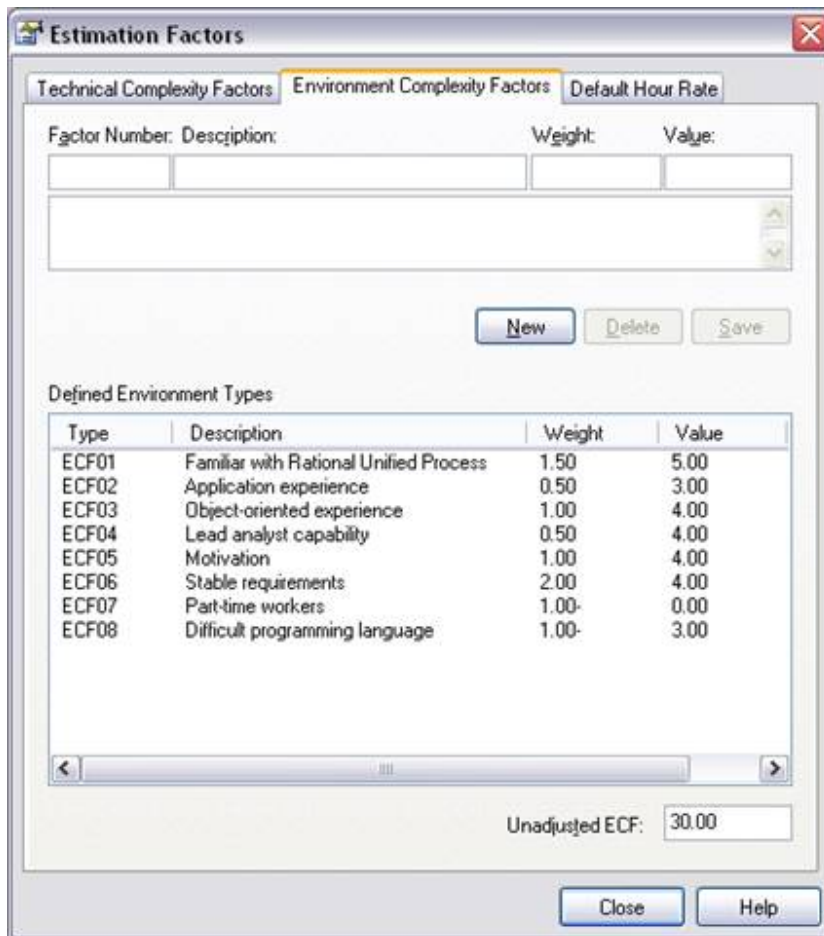
T6	Installability	0.5	Specifies the method of installation, ease of installation and whether it is user-friendly, indicates whether the participation of specialists will be required for the installation and initial setup on the part of the software supplier.
T7	Usability	0.5	Determines the adjustment of user's interface to his or her needs, convenience in use and whether it is easy to learn how to use the system.
T8	Portability	2	Informs whether application should operate in certain environments.
T9	Modifiability	1	Determines whether the system is to be built in such a way that it will be easy to further develop it in the future.
T10	Concurrency	1	Informs whether the concurrent processing will take place in the system.
T11	Includes special security requirements	1	Defines whether the system will require the use of mechanisms related to securing access to the system or data.
T12	Provides direct access by third parties	1	Determines to what degree other external systems or actors will use the system.
T13	Special User training facilities are required	1	Determines the need to organize trainings for users.

Technical factors can increase the project duration. The bigger the value of each factor, the longer the estimation.

## Step 4 - Adjust Environmental Factors:

To adjust environmental factors, in *Estimation Factors* form, click on *Environmental Complexity Factors* tab and change the values. Like technical factors, the value can be 0 to 5.

You can also add/remove factors regarding your project conditions:



The following table helps you find out the values of environmental factors:

Environment Factor	Description	Weight	Explanation
E1	Familiarity with Life-Cycle model used	1.5	Informs if the team is familiar with problem domain and technical aspects of the solution to a customer's problem. Attention should be also paid to knowledge of methodology in which the project is implemented e.g. RUP (Rational Unified Process), as well as knowledge of system modeling languages e.g. UML
E2	Application domain experience	0.5	Generally understood experience of the team in developing the software.

E3	Experience with development methodologies used	1	Experience in designing object-oriented applications related to the ability to design object-oriented applications and the ability to use support tools for designing IT systems.
E4	Analyst capability	0.5	Determines the ability of the analyst to properly acquire requirements from the client and the possession of knowledge related to the problem which is being solved.
E5	Team motivation	1	Assesses the ability of the team to engage in the assigned task.
E6	Stability of requirements	2	Defines if the requirements are not exposed to frequent changes.
E7	Use of part-time team members	-1	determines whether there is a big number of part-time staff in the team (e.g. interns, students)
E8	Use of difficult programming language	-1	Specifies how difficult it is to learn the programming language in which the future IT system will be implemented.

All environmental factors except E7, E8, will decrease the project duration. It means, the bigger the value of E1 to E6, the shorter the estimation but it is vice versa for E7 and E8.

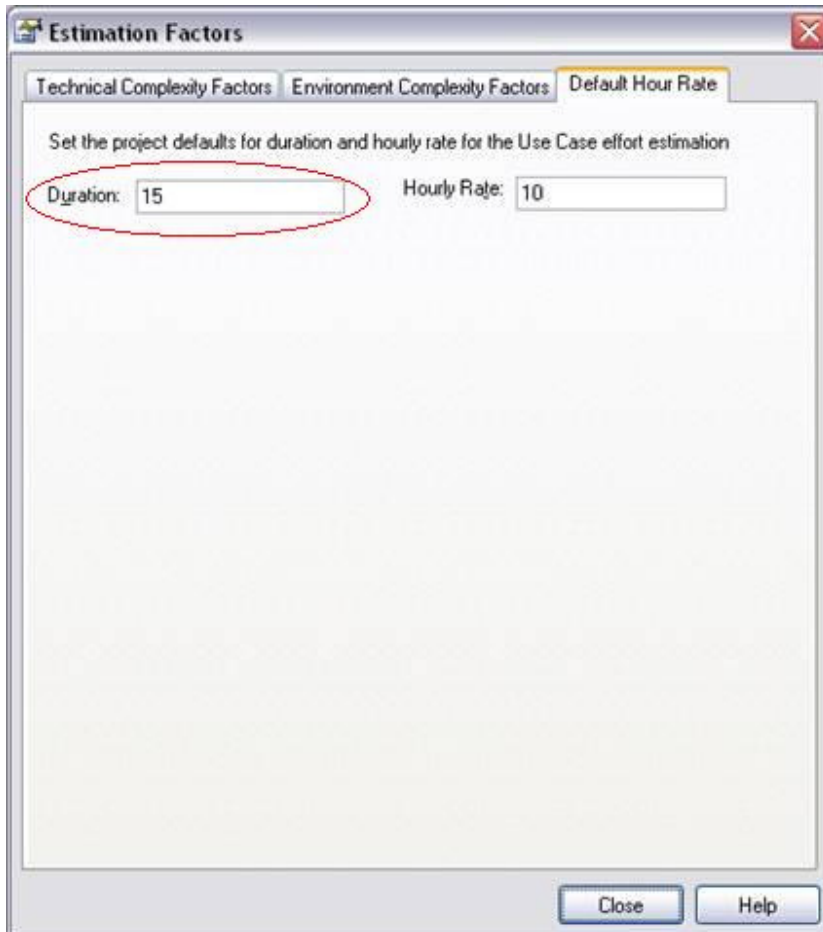
If you don't know how to determine the value of E8, take a look at this table:

<b>0</b>	<b>Easy.</b> Within one week the language can be picked up.
<b>1</b>	At least two week is needed to pick up the language.
<b>2</b>	At least one month is needed to pick up the language.
<b>3</b>	Special training needed for the language.
<b>4</b>	Special training needed for the language and need help during the project.
<b>5</b>	<b>Difficult.</b> Needs only experienced people.



## Step 5 – Set Productivity Factor (PF):

After adjusting technical and environmental factors, you should set productivity factor which is in *Default Hour Rate* tab of *Estimation Factors* form:



The screenshot shows a dialog box titled "Estimation Factors" with three tabs: "Technical Complexity Factors", "Environment Complexity Factors", and "Default Hour Rate". The "Default Hour Rate" tab is selected. Below the tabs, there is a text label: "Set the project defaults for duration and hourly rate for the Use Case effort estimation". There are two input fields: "Duration:" with the value "15" and "Hourly Rate:" with the value "10". The "Duration:" field is circled in red. At the bottom of the dialog box, there are "Close" and "Help" buttons.

Duration (productivity factor), can varies from 10 to 30.

Setting an hourly rate is the most difficult factor in an accurate estimation. Typical ranges can vary from 10 to 30 hours per Use Case point. Studying the *Use Case Points Method*, from which this variable is defined, can help you to understand its role in the estimation and facilitate selection of a suitable initial value. The best way to estimate this value is through analysis of previous completed projects. By calculating the project estimation on a completed project for which the Use Cases and environment are configured within EA, you can adjust the hour rate to render an appropriate value for your unique work environment.

## Step 6 – Calculate UCP:

The final step is to calculate Use Case Points and project man-hours.

To calculate UCP for the entire EA project, Click *Project Metrics...* from *Project* menu.

You can also calculate UCP for a specific package. To do so, right click on the package and from *Documentation* select *Package Metrics...*

In this form you can see the partial result calculation such as TCF and ECF as well as total hours and cost of project. You are able to manipulate the values and re-calculate but don't forget to check *Include Actors* anyway.

**Use Case Metrics**

Use Cases  
 Root Package: Package1 [Reload]  
 Phase like: \*  
 Keyword like: [ ]  
 Bookmarked: All [v]  
 Use Cases: 1 [ ]  Include Actors

Package	Name	Type	Complexity	Phase
Package1	Salesperson	Actor	3	1.0
Package1	Invoice	UseCase	10	1.0

Technical Complexity Factor  
 Unadjusted TCF Value (UTV): 47  
 TCF Weight Factor (TWF): 0.01  
 TCF Constant (TC): 0.6  
 TCF = TC + (TWF x UTV): 1.07

Environment Complexity Factor  
 Unadjusted ECF Value (UEV): 21.5  
 ECF Weight Factor (EWF): -0.03  
 ECF Constant (EC): 1.4  
 ECF = EC + (EWF x UEV): 0.755

Unadjusted Use Case Points (UUCP) = Sum of Complexity: 13  
 Ave Hours per Use Case: Easy: 57 Med: 115 Diff: 173

Total Estimate  
 Use Case Points (UCP) = UUCP \* TCF \* ECF = 13 \* 1.07 \* 0.755 = 10 UCP  
 Estimated Work Effort (hours) = 15 \* 10 = 150 Hours  
 Estimated Cost = EWE \* Default hourly Rate = 150 \* 10 = 1500 Cost

[Re-Calculate] [Report] [View Report] [Default Rate] [Close] [Help]

In our case the estimated duration will be 420 man-hour and if the cost of an hour is \$10, it would cost \$4200.

As you can see, it also says that estimated time for developing a medium Use Case is 115 hours. At first it may look too much but if you divide it to other smaller tasks, it would be more reasonable.

The following table shows estimated tasks for developing a Use Case in Elaboration and Construction phases of RUP:

<b>Category</b>	<b>Task</b>	<b>%</b>	<b>Medium Use Case (hours)</b>
<b>Environment:</b>			
	<b>Planning, Integration, Change management, etc.</b>	<b>6</b>	<b>6.9</b>
<b>Analysis:</b>			
	<b>Learning, Research and Realizing</b>	<b>9</b>	<b>10.35</b>
	<b>Documentation, Modeling</b>	<b>10</b>	<b>11.5</b>
	<b>Review</b>	<b>5</b>	<b>5.75</b>
	<b>Re-work after review</b>	<b>7</b>	<b>8.05</b>
	<b>Prototyping</b>	<b>6</b>	<b>6.9</b>
	<b>Test Case</b>	<b>4</b>	<b>4.6</b>
<b>Development:</b>			
	<b>Design</b>	<b>5</b>	<b>5.75</b>
	<b>Coding</b>	<b>23</b>	<b>26.45</b>
	<b>Unit testing</b>	<b>7</b>	<b>8.05</b>
	<b>Re-work after test</b>	<b>10</b>	<b>11.5</b>
<b>Test:</b>			
	<b>Intra-team testing</b>	<b>8</b>	<b>9.2</b>
<b>Total</b>		<b>100</b>	<b>115</b>

As you have noticed, Use Case Points estimates 115 hours for a medium Use Case. However, its coding time would be about 27 hours.

## Summary

To estimate a software project with **Use Case Points** method using [Spax Enterprise Architect](#):

- a) Recognize system actors
- b) Recognize system Use Cases
- c) Determine Use Cases complexity
- d) Determine actors' complexity
- e) Adjust technical factors
- f) Adjust environmental factors
- g) Set Estimation factors (PF)
- h) Calculate UCP

After calculating, UCP gives you a rough estimation of entire project and needed time for each Use Case that can be easy, medium or difficult. You can then divide estimated Use Case hours to smaller tasks and prepare to plan them.

## Resources

- Gautam Banerjee, Use Case Points - An estimation approach (Paper)
- Nguyen Mai, Himanshu Saxena and Lucas Grossi, Use Case Point Estimation (Presentation)
- RoyClem, Project Estimation with Use Case Points (<http://www.codeproject.com/KB/architecture/usecasep.aspx>)
- Enterprise Architect User Guide