Understanding TESTING ESTIMATION using Use Case Metrics

A loom for testing Estimation

Prepared By

Sanjeev Kumar
Sanjeevchauhan@qainfotech.net

Date:-08Sept, 2009

Table of Content:-

1. Purpose: - The purpose of this paper is to explain a new approach to the estimation of software testing efforts based on Use Case Points [UCP] as a fundamental project estimation measure. Object-oriented applications frequently rely on Use Cases to describe the business processes of a software application.

This methodology for estimation is in fact more reliable than functional point [FP].It works on the basic principle that more the complexity of the use case more is the time taken to design, develop, test and implement. The caveat here is that the V-model must be in use and use case generation must start becoming available right at the requirements gathering phase. This paper provides the steps wise testing estimation process using the Use Case Point [UCP].

2. Preamble: - Estimation effort for Software testing is one of the most important facets of the entire testing life cycle as it is directly proportional to the cost of the project. Estimation has great impact on all the most important aspects of a customer expectation – Time, Cost and Quality. A correct estimation helps in delivering the products at right time. If estimation is wrong it might lead to unwanted delay in deliverables, increased cost and inappropriate results.

There are several methods available for testing estimation like Test Case Points, Function Point etc. Use Case methodology is gaining far above the ground popularity for estimating software effort. It could be very useful in case of bidding projects as the use cases are one of the first or sometimes the only information available in the beginning of a software project.

Why estimate?

You cannot plan if you cannot measure, and if you fail to plan, you have planned to fail.

An omelet, promised in two minutes, may appear to be progressing nicely. But when it has not set in two minutes, the customer has two choices

Either wait & eat (or) eat it raw!

If the customer does not want to eat it raw as well as he does not want to wait, then the cook has too has a choice - he can turn the heat on and

Get the omelet burnt in one part and raw in another!

In software industry customers & vendors have also the same choices. One most important factor that requires consideration is setting the expectations right up front. This can be done first by getting the estimation right.

3. Effort Estimation using UCP:-

3.1 Size Estimation:-

The First step for every type of estimation is to calculate the size of activity to be performed. In Use case methodology, there are 4 major components which are important for determining the size of a project –

1. Actor Weight [AW]
2. Use Case Weight [UCW]
3. Technical complexity Factors [TCF]
4. Environmental complexity Factors [ECF]

3.1.1 Actor Weight [AW]:-

The Actor Weight [AW] is classified into 3 different types- Simple, Medium and Complex. *Figure A* describes the criteria for each type and also the factor (weight age) or numeric values.

The total Actor weight can be calculated by categorizing all the actors as simple, medium and complex and then adding up all the factors (weight age).

Figure A

Actor Weight		
Weight age Type	**Factor(Weight age)**	**Criteria/Description**
Simple	1	Represents another system with defined application programming interface(API)
Medium	2	The Actor represents another system interacting through a protocol, like TCP/IP.Or a person interacting through a text based interface.
Complex	3	Can be a person interacting via an interface(GUI)

3.1.2 Use Case Weight [UCW]:

Usage case weight can be calculated in the same way as Actor weight is calculated. All the use cases need to be categorized as simple, medium and complex. The final weight is the sum of the factors for all the categorized use cases (See *Figure B).*

Keep in mind the number of steps in a scenario affects the estimate. A large number of steps in a use case scenario will bias the UCW toward complexity and increase the UCPs.

Figure B

Use Case Weight

Weight age Type	Factor(Weight age)	Criteria/Description
Simple	5	Very simple interface, which touches a single data-base entity, having 3 or lesser transitions/steps.
Medium	10	More interface, which touches two or more data-base entities, having 4-7 transitions/steps
Complex	15	Very complex interface, which touches three or more data-base entities, having more than 7 transactions/steps

3.1.3 Unadjusted Use Case Points (UUCP):

UUCP = Actor Weight [AW] + Use Case Weight [UCW]

3.1.4 Technical Complexity Factor [TCF]:

There are mainly 7 standards which impacts the overall estimation for a project. The *Figure C* shows the list of all the factors from testing perspective with their Weight age. The weight of a technical factor indicates its impact on the overall estimation. Technical factors are evaluated by the development team and assigned a value between zero and five. A weight of 0 indicates the factor is irrelevant and the value 5 means that the factor has the most impact.

Technical complexity factor is calculated as below –

TCF = C1 + C2*(Total TF)

As per the Use Case Point method, the impact of TCF on use case points should vary from range of 0.6 (40% reduction) to 1.3 (30% increase). In order to achieve this range the coefficient C2 needs to be modified for the Testing Effort.

C2 for testing = (1.3 – 0.6)/50 = **0.014**

Where 50 is the max value of TCF for testing.

Where,
C1 = Constant with value 0.6
C2 = Constant with value 0.014
Total EF = Total Technical Factor

Figure C

Technical Factors		
S.No	**Value(Weight age)**	**Criteria/Description**
1	2	Test Tools.
2	2	Document Inputs
3	1	Test-ware Reuse
4	2	Distributed System
5	1	Performance Objectives
6	1	Security Features
7	1	Complex Interfacing

3.1.5 Environmental Complexity Factor [ECF]:

There are 7 environmental factors which impacts the overall estimation for a project. The *Figure D* shows the list of all the factors from testing perspective with their Weight. A rating of 0 means the environmental factor is irrelevant for this project; 3 is average; 5 mean it has strong influence. The weight of a technical factor indicates its impact on the overall estimation.

Environmental complexity factor is calculated as below –

$$\mathbf{ECF = C1 + C2*(Total\ EF)}$$

As per the Use Case Point method, the impact of ECF on use case points is more than TCF and it should vary from range of 0.0425 (57.5 % reduction) to 1.4 (40% increase). In order to achieve this range the coefficient C2 needs to be modified for the Testing Effort.

C2 for testing = (1.4 – 0.0425)/37.5 = **0.0362**
Where 37.5 is the max value of ECF for testing

Where,

C1 = Constant with value 1.4
C2 = Constant with value (-0.0362)
Total EF = Total Environmental Factor

Figure D

Environmental Complexity Factors

S.No	Value(Weight age)	Criteria/Description
1	1	Application Knowledge
2	2	Test Environment
3	1	Test Data
4	0.5	Test Lead Capability
5	1	Motivation
6	2	Stable Requirements
7	-1	Part Time Workers

3.1.6 Calculate Final Use Case Points (UCP):-

$$UCP = UUCP*TCF*ECF$$

3.2 Effort Estimation:-

3.2.1 Conversion Factor (CF):-

The Conversion Factor (CF) is a ratio of the number of man hours per use case point based on past projects. Once the size of a project has been calculated in terms of Adjusted Use Case Points, the total size needs to be converted to effort by multiplying it with a Conversion factor. The Conversion factor (Effort/Size) is defined as the total testing time required to test one Use Case Point.

The Conversion factor can be derived by reverse engineering technique i.e. by putting the historical project data in the estimation template for different technologies. It is 20(hrs) for Java based applications. If no historical data has been collected, a figure between 15 and 30 is suggested by industry experts.

3.2.2 Calculating Final Effort:-

Final Effort = UCP * Conversion factor

4. Benefits of Use Case Point methodology:-

1. This scientific method gives more accurate and precise results over any traditional method available for effort estimation.

2. No any detail requirements are required for estimation.

3. This method is very well suited for bidding projects as most of the time use case is the only information available at the beginning of a project.

4. The Use Case point method considers the technical and environmental factors which can be refined further to achieve more accurate estimates.

5. This can be used to illustrate productivity benchmarks across an organization since it is independent of test cases.

5. Case Study:-

5.1 Actor Weight [AW]:-

Actor Weight			
S. No	**Actor Name**	**Weight age Type**	**Factor**
1	**Actor 1**	**Simple**	**1**
2	**Actor 2**	**Medium**	**2**
3	**Actor 3**	**Complex**	**3**
4	**Actor 4**	**Simple**	**1**
5	**Actor 5**	**Medium**	**2**
		Total	**9**

5.2 Use Case Weight [UCW]:-

Use Case Weight			
S. No	**Use Case Description**	**Weight age Type**	**Factor**
1	Use Case 1	Simple	5
2	Use Case 2	Medium	10
3	Use Case 3	Complex	15
4	Use Case 4	Simple	5
5	Use Case 5	Medium	10
		Total	45

5.3 Unadjusted Use Case Points (UUCP):-

$$
\begin{aligned}
UUCP &= AW + UCW \\
&= 9 + 45 = \mathbf{54}
\end{aligned}
$$

5.4 Technical Complexity Factor Calculation [TCF]:-

Technical Factors				
S. No	**Description**	**Weight age**	**Perceived Complexity**	**Calculated Factor**
1	Test Tools	2	1	2
2	Document Inputs	2	2	4
3	Test-ware reuse	1	3	3
4	Distributed System	2	4	8
5	Performance Objectives	1	5	5
6	Security Features	1	1	1
7	Complex Interfacing	1	3	3
			Total Factor	26

$$\begin{aligned}\textbf{TCF} &= C1 + C2*\text{Total TF}\\ &= 0.6 + (0.014)*26\\ &= \mathbf{0.96}\end{aligned}$$

A perceived complexity of 0 means the technical factor is irrelevant for the project; 3 is average; 5 means it has strong influence.

Each factors weight is multiplied by its perceived complexity to produce its calculated factor. The calculated factors are summed to produce the Total Factor.

5.5 Environmental Complexity Factor Calculation [ECF]:-

Environmental Factors				
S. No	**Description**	**Weight age**	**Perceived Complexity**	**Calculated Factor**
1	**Application Knowledge**	**1**	**5**	**5**
2	**Test Environment**	**2**	**3**	**6**
3	**Test Data**	**1**	**1**	**1**
4	**Test Lead Capability**	**0.5**	**1**	**0.5**
5	**Motivation**	**1**	**5**	**5**
6	**Stable Requirements**	**2**	**2**	**4**
7	**Part Time Workers**	**-1**	**2**	**-2**
			Total Factor	**19.5**

$$\begin{aligned}\textbf{ECF} &= C1 + C2*\text{Total EF}\\ &= 1.14 + (-0.0362)*19.5\\ &= \mathbf{0.43}\end{aligned}$$

5.6 Calculate Final Use Case Points (UCP):-

UCP = UUCP*TCF*ECF
= 54*0.96*0.43= **22.29**

5.7 Calculating Final Effort:-

Final Effort = UCP * Conversion factor

Final Effort (Hrs) = UCP*20 = 22.29*20 = 445.8

6. References:-

http://www.codeproject.com/KB/architecture/usecasep.aspx
http://www.stsc.hill.af.mil/crosstalk/2006/02/0602Clemmons.pdf

"a stitch in times saves nine"