TESTING ESTIMATION WITH USE CASE POINTS

An approach for testing Size and Effort Estimation

Prepared By

Vineet Singh Pangtey
vspangtey@gmail.com

DATE: 30Sept, 2008

TABLE OF CONTENTS

1. Purpose :-

The purpose of this white paper is to explain the testing size and effort estimation using the Use Case Point methodology.

The Use case methodology for estimation is widely accepted in the software industry. 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.

There is a lot of information available on estimating development effort using the Use Case point methodology but there is hardly any information available on estimating the testing activities using the same methodology. This paper provides the steps wise testing estimation process using the Use Case Points.

2. Introduction:-

Estimation of software testing effort is one of the most important aspects of the entire testing life cycle process as it is directly proportional to the cost of the project. Estimation has an impact on all the 3 most important aspects of a customer need – Time, Cost and Quality. A correct estimation helps in delivering the products in right time. If the estimation is not correct it might lead to delay in deliverables, increased cost and inappropriate results.

There are lots of methods available right now for testing estimation like Test Case Points, Function Point etc. Use Case methodology is also gaining popularity for estimating software effort. It can used 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.

3. Testing Size and Effort Estimation using UCP:-

3.1 Size Estimation

The first activity in any kind of estimation is to calculate the size of the activity to be done. In Use case methodology, there are 4 major components which are important for determining the size of a project -

1. Actor Weight
2. Use Case Weight
3. Technical Factors
4. Environmental Factors

3.1.1 Actor Weight:

The weight of actor has been classified into 3 categories – Simple, Medium and Complex. The *table A* describes the criteria for each category and also the factor 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.

Table A

ACTOR WEIGHT		
Weightage Type	**Factor**	**Criteria**
Simple	1	Represents another system with defined API
Medium	2	Can be another system that interacts through a protocol such as TCP/IP or a person interacting through a text based interface.
Complex	3	Can be person interacting through a GUI.

3.1.2 Use Case Weight:

The steps for calculating use case weight are similar as calculating actor weight. 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.

The *table B* has the criteria and factor for all the categories of Use Cases.

Table B

USE CASE POINTS		
Weightage Type	**Factor**	**Criteria**
Simple	1	3 or lesser transactions
Medium	2	4-7 transactions
Complex	3	More than 7 transactions

3.1.3 Unadjusted Use Case Points (UUCP):

UUCP = Actor Weight + Use Case Weight

3.1.4 Technical Complexity Factor:

There are 7 technical factors which impacts the overall estimation for a project. The *table* C shows the list of all the factors from testing perspective with their Weight. The weight of a technical factor indicates its impact on the overall estimation.

The 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

Table C

Techincal Factors		
S.N.	Description	Assigned Value
1	Test Tools	2
2	Documented Inputs	2
3	Test-ware Reuse	1
4	Distributed Systems	2
5	Performance Objectives	1
6	Security Features	1
7	Complex Interfacing	1

3.1.5 Environmental Complexity Factor

There are 7 environmental factors which impacts the overall estimation for a project. The *table D* shows the list of all the factors from testing perspective with their Weight. The weight of a technical factor indicates its impact on the overall estimation.

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

Table D

Environmental Factors		
S.N.	Description	Assigned Value
1	Test Environment	2
2	Test Data	1
3	Application Knowledge	1
4	Test Lead Capability	0.5
5	Motivation	1
6	Stable Requirements	2
7	Part Time Workers	-1

3.1.6 Calculate Final Use Case Points (UCP)

UCP = UUCP*TCF*ECF

3.2 Effort Estimation

3.2.1 Conversion Factor

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.

3.2.2 Calculating Final Effort

Final Effort = UCP * Conversion factor

4. Benefits of Use Case Point methodology:-

1. The method is very much scientific. It gives more accurate and precise results over the traditional method of effort estimation.

2. The detailed requirements are not required for estimation.

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

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

5. The size estimation is independent of test cases and provides a unit which can be used to draw productivity benchmarks across an organization.

5. Case Study:-

5.1 Actor Weight

S. N.	Actor Name	Weight	Factor
1	Actor 1	Simple	1
2	Actor 2	Medium	2
3	Actor 3	Complex	3
4	Actor 4	Simple	1
		Total	**7**

5.2. Use Case Weight

S. N.	Use Case Desription	Weight	Factor
1	Use Case 1	Simple	5
2	Use Case 2	Medium	10
3	Use Case 3	Simple	5
4	Use Case 4	Complex	15
		Total	**35**

5.3 Unadjusted Use Case Points (UUCP):

UUCP = 7 + 35 = **42**

5.4 Technical Complexity Factor Calculation

Techincal Factors				
S.N.	Description	Weight	Percieved Complexity	Calculated Factor
1	Test Tools	2	1	2
2	Documented Inputs	2	2	4
3	Test-ware Reuse	1	3	3
4	Distributed Systems	2	4	8
5	Performance Objectives	1	5	5
6	Security Features	1	1	1
7	Complex Interfacing	1	3	3
		TOTAL TF		26

TCF = 0.6 + (0.01)*Total TF
= 0.6 + (0.014)*26
= **0.96**

5.5 Environmental Complexity Factor Calculation

Environmental Factors				
S.N.	Description	Weight	Percieved Complexity	Calculated Factor
1	Test Environment	2	3	6
2	Test Data	1	1	1
3	Application Knowledge	1	5	5
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 EF		19.5

ECF = 1.14 + (-0.03)*Total EF
= 1.14 + (-0.0362)*19.5
= **0.69**

5.6 Calculate Final Use Case Points (UCP)

UCP = UUCP*TCF*ECF
= 42*0.96*0.69
= **28.82**

5.7 Calculating Final Effort

Final Effort (Hrs) = UCP*20 = 28.82*20 = **576.4**

6. References

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