Software Test Estimation Tool: Comparable with COCOMOII Model

Shaiful Islam[1], Bishwajit B. Pathik[2], Manzur H. Khan[3], Mamun Habib[4]
[1] Department of Computer Science, American International University-Bangladesh, Dhaka, Bangladesh
[2]Department of Electrical and Electronic Engineering, American International University-Bangladesh, Dhaka, Bangladesh
[3]Department of Decision Science, Universiti Utara Malaysia, Kedah, Malaysia
[4]BRAC Business School, BRAC University, Dhaka, Bangladesh
[1]jshaiful@gmail.com, [2]bishwajit@aiub.com, [3]manzur@aiub.com, [4]mamunhabib@bracu.ac.bd

Abstract – **This study illustrates an estimation tool for software test that provides the estimated time and the cost of any sort of software test project. There are different well-recognized estimation tools for software development process [9], however, there remains a lack of standard tools for estimation of Software Test phase. Therefore, the authors developed a web base tool (www.4beats.net/tpet/) in order to determine the time and cost of projects. This paper highlights the comparable result with COCOMOII (Constructive Cost Model) model. More than 22 projects are already estimated by this open tool and from them 5 case projects, namely, Climate Resilient Ecosystems and Livelihood (CREL), Management Accounts Consolidation System for Line Director (LDMACS), PBI (Police Bureau of Investigations) Works Management System Software, PBI Mobile Apps and Human Resources Management System (HRMS) have selected. This paper would play a vital role to estimate time and cost for software test project and be beneficial for the software industry.**

Keywords – **Test Estimation, Use Cases, Functions, Cost Estimation, Time Estimation, User Story.**

I. INTRODUCTION

Software estimation is a crucial factor for resource, schedule and budget [4], [5]. Project failures can be avoid using proper estimation technique by means of proper value of the system and the proper payment for the developers. Software Development Life Cycle (SDLC) consists of requirements analysis, design, coding, testing, deployment and maintenance. Although actual test execution starts after coding but test planning and preparing test cases can be applied from the initial phase of the project. Depending on effective and efficient software test estimation by Quality analyst project cost can be estimated earlier with accurate output. Implementing better design for system development and maintenance we can reduce in cost. The authors identify the methodology of early estimation in software test projects. The steps and proofs of the system have defined in different sections with example. To understand the process sequentially details of fundamental theory, estimation practices, methods of calculation, advantages, disadvantages and others feature in the next phase with the earlier related study. Consequently, after analyzing the software test estimation's findings, results, comparative estimation with other models and conclusion discussed in research base.

II. RELATED STUDY

Last couple of decades it is observed that software project failures is common and important factors. Maximum project fails not because of the critical issue in the development but because of software planning and estimation. Around 30% to 40% of the software projects are completed with over schedule and cost and the others fail [11]. According to the Standish group's CHAOS reports, 70% software projects gets fail. Large number of projects cost overrun and it is 189% in 1994 CHAOS report [11]. Additionally, Ref. [11] signifies that the CHAOS report may be corrupted. It is easy to understand by the statistics data how the crisis concerned on future software development projects [11].

To identify the facts of software projects failure several studies have been conducted in the last decades [12]. Efforts were given to search between 2100 internet sites and found 5000 reasons for the software project failures. The reasons includes insufficient and improper requirements engineering, poor planning the project, rapidly decision changes in the requirements in the early phase and wrong estimations were the most vital reasons. The maximum failure of the projects found is inaccurate estimations and it is the root cause [11]. In spite of the indicated statistics might be cynical, incorrect estimation is a actual difficulty in the software production's world that would be resolved. We need to find out the efficient techniques and reliable models to resolve the issue.

However, lack of appropriate software test estimation tool in the industry, researched paid attention to build a tool that would satisfies the need of early estimation. In this article and later part can prove that with methodology and the techniques with projects statistics data break the bottleneck of the estimation issues and would lead to success in terms of time and costing.

III. TECHNIQUE FOR SOFTWARE TEST ESTIMATION [1], [3], [4], [5], [6]

This study illustrates software test estimation techniques stepwise with the method. The formulas, calculations, ratings, weight factor and the explanation of each area would be included here. It could be expressed as:

Total Hours = PT * PC, **Total MM** = (Total Hours / 7) / 22

Project Cost = Total Hours * Per Hour Cost + Overhead Cost

(a) Function-dependent (D_f) [1]

$$D_f = ((U_e + U_y + I + C) /16) * U$$

Where, D_f = weighting factor for the function-dependent factors, U_e = user-importance, U_y = usage-intensity, I = interfacing, C = complexity and U = uniformity

User Importance (U_e): Expert judgment techniques involve consulting with software. A useful rule of thumb is that about 75% of functions should be placed in the "high" category, 50 in the "normal" category and 25% in the "low" category.

Usage Intensity (U_y): The users and the size of the user group that uses the function have defined the usage intensity as the frequency with which a certain function is processed. As with user-importance the usage-intensity is being determined at a user-function level.

Interfacing (I): Interfacing is an expression of the extent to which a modification in a given function affects other parts of the system. The degree of interfacing is determined by ascertaining first the logical data sets (LDSs) which the function in question can modify, then the other functions which access these LDSs. A CRUD table (Table 1) is very useful for determining the degree of interfacing.

TABLE 1
TABLE FOR RATING AND WEIGHT OF CRUD

LDS\Functions	*1*	*2-5*	*>5*
1	L	L	A
2-5	L	A	H
>5	A	H	H

L: Low interfacing, A: Average interfacing, H: High interfacing

Complexity (C): Complexity of functions of any functions depends on condition of it. Rating and weight is totally depending on condition of the functions.

(b) Dynamic Quality Characteristics (TP_f)

$$TP_f = FP_f * D_f * Q_d$$

Where, TP_f = number of test points assigned to the function, FP_f = number of function points assigned to the function, D_f = weighting factor for the function-dependent factors, Q_d = weighting factor for the dynamic quality characteristics = per characteristic 0.02/4*Weighting Factor, then added together (explicitly measurable quality characteristics)

In Test Point Analysis (TPA), four dynamic, explicitly measurable quality characteristics are recognized. [1], [3-4].

Uniformity (U): Under the following circumstances 60% of the test points assigned to the function under analysis count towards the system total. A uniformity factor of 0.6 is assigned in cases of the kinds described above; otherwise 1 is allotted.

(c) Total number of Test Points

$$TP = \sum TP_f + (FP * Q_i) / 500$$

Where, TP = total number of test points assigned to the system as a whole, $\sum TP_f$ = sum of the test points assigned to the individual functions (dynamic test points), FP = total number of function points assigned to the system as a whole (minimum value 500), Q_i = weighting factor for the indirectly measurable quality characteristics.

(d) Primary Test Hours

$$PT = TP * P * E$$

Where, PT = total number of primary test hours, TP = total number of test points assigned to the system as a whole, P = productivity factor, E = environmental factor

$$\textit{Total test hours} = PT * PC$$

Where, PT= Primary test hours formula, PC= Total number of test hours.

(e) Static Test Points

One has to determine whether the static measurable quality characteristics are relevant for test purposes. A static test can be carried out using a checklist. In principle all ISO 9126 quality characteristics [4] can be tested using a checklist. E.g. security can therefore be measured dynamically, using a semantic test, and/or statically, by evaluating security measures with support of a checklist.

TABLE 2
TABLE FOR FUNCTION DEPENDENT AND DYNAMIC QUALITY CHARACTERISTICS

D_f & TP_f	*Rating*	*Factors*	*Weight*
User Importance	3	Low: the importance of the function relative to the other functions is low	0.5
	6	Normal: the importance of the function relative to the other functions is normal	0.75
	12	High: the importance of the function relative to the other functions is high	1
Usage Intensity	2	Low: the function is only used a few times per day or per week	0.1
	4	Normal: the function is being used a great many times per day	0.2
	12	High: the function is used continuously throughout the day	0.3
Interfacing	2	The degree of interfacing associated with the function is low	0.5
	4	The degree of interfacing associated with the function is normal	0.75
	8	The degree of interfacing associated with the function is high	1
Complexity	3	Function contains no more than 5 conditions	0.5
	6	Function contains between 6 and 11 conditions	0.75
	12	The function contains more than 11 conditions	1
Dynamic Quality Characteristics	3	Suitability: Quality requirements are not important and are therefore disregarded for test purposes.	0.05
	4	Security: Quality requirements are relatively unimportant but do need to be taken into consideration for test purposes.	0.10
	5	Usability: Quality requirements are of normal importance.	0.25
	6	Efficiency: Quality requirements are very important.	0.50

Method of Calculation (Q_i): The factor Q_i obtained 16 values if the quality characteristic is tested by means of a checklist (static test). Every quality characteristic add statics test, Q_i rating increased by 16.

Primary Test Hours (PT): The formula gives the total number of test points assigned to the system as a whole. This total number of test points is a measure of the volume of the primary test activities. The primary test hour count is the number of hours required for carrying out the test activities involved in the test life cycle phases Preparation, Specification, Execution and Completion.

Productivity Factor (P): The productivity factor indicates the number of test hours required per test point. The productivity factor is a depends on experience, skill and knowledge of the test team. Once project completed we can calculate the productivity. The higher the productivity factor, the greater the number of test hours required. In practice, the productivity factor has shown to have a value between 0.7 and 2.0.

Environmental Factor (E): The number of test hours required for each test point is also influenced by the. To calculate of the environmental factor different environment variables are used.

(i) Test Tools: The test tools variable reflects the testing tools and automated test used in the testing. To calculate this it is important which tools used in the primary test activities.

(ii) Development Testing: Details description is available in reference [4], [5]

(iii) Test basis: Details description in reference [4], [5]

TABLE 3
TABLE FOR UNIFORMITY

	Sl. No.	Cases
Uniformity	1	In the case of a second occurrence of a virtually unique function: in such cases, the test specifications can be largely reused
	2	In the case of a clone function: the test specifications can be reused for cloned functions
	3	In the case of a dummy function (provided that reusable test specifications have already been drawn up for the dummy).
Tools	1	Testing involves the use of a query language such as SQL; a record and playback tool is also being used.
	2	Testing involves the use of a query language such as SQL, but no record and playback tool is being used.
	4	No test tools are available.
Development	2	A development testing plan is available and the test team is familiar with the actual test cases and test results.
	4	A development testing plan is available.
	8	No development testing plan is available.
Test Basis	3	During the system development documentation standards are being used and a template, in addition the inspections are organized
	6	During the system development documentation standards are being used and a template.
	12	The system documentation was not developed using a specific standards and a template.
Development Environment	2	The system was developed using a 4 GL (Generation Language) programming language with an integrated DBMS containing numerous constraints.
	4	The system was developed using a 4 GL programming language, possibly in combination with a 3 GL language.
	8	The system was developed using only a 3 GL programming language such as COBOL, PASCAL or RPG.
Test Environment	1	Environment has been used for testing several times in past
	2	The test is to be conducted in a newly equipped environment similar to other well-used environments within organization.
	4	The test is to be conducted in a newly equipped environment which may be considered experimental within organization.
Testware	1	A usable general initial data set and specified test cases are available for the test.
	2	A usable general initial data set is available for the test.
	4	No usable testware is available.

(iv) Development Environment: Details description in reference [4], [5]

(v) Test Environment: Details description in reference [4], [5]

(vi) Testware: Details description in reference [4], [5]

(vii) Method of Calculating Environment Factor: Details description in reference [4], [5]

Total Number of Test Hours (PC): Details description in reference [4], [5]

- *Team Size:* Details description in reference [4], [5]
- *Management Tools:* Details description in reference [4], [5]

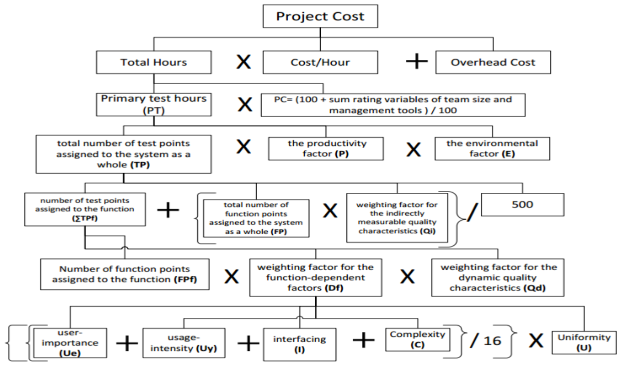


Fig. 1. Diagram of the Estimation Methodology

Method of Calculation: Planning and management percentage is adding with ratings for the two influential factors (team size and planning and control tools). Multiplying the primary test hour with the count by this percentage generated expected test hours. Primary test hours with Planning and control generate the total number of test hours.

Breakdown between Phases: Total test point is the estimation of complete test process excluding test plan. In the structured test process [7], [8] are of five life cycles; depending on clients demand it require individual cycles or full cycle's estimation. Estimation of planning and control phase will equally treated as planning and control allowance, i.e. the primary test hour count multiplied by the planning and control percentage. The primary test hours are then divided between the Preparation, Specification, Execution and Completion phases. Different phases of the breakdown can be differ from one organization to another; even one unit to another of similar organization. Experts of TPA technique suggest the following percentages: preparation (10%), specification (40%), execution (45%) and completion (5%).

Total Hours = [{{∑(FPf * (((Ue + Uy + I + C)/16) * U) * Qd) + ((FP * Qi) / 500)} * P * E} * PC]
Total MM = [[{{∑(FPf * (((Ue + Uy + I + C)/16) * U) * Qd) + ((FP * Qi) / 500)} * P * E} * PC] / 7] / 22
Project Cost= [{{∑(FPf * (((Ue + Uy + I + C)/16) * U) * Qd) + ((FP * Qi) / 500)} * P * E} * PC] * Per Hour Cost + Overhead Cost

IV. PROJECTS CALCULATION

In this paper, we have taken 5 projects out of 22 and compared with COCOMOII. Both *functions* and *use cases* considered for the calculation. First project calculated with *functions* [2] and the second one considered with *use cases* [10]. Full system implemented in a such way so that any changes would be reflect immediately to the next to understand the calculation. We have measured the maximum to avoid any kind of fractional calculation error.

Case 1: LDMACS

*Project Name: Management Accounts Consolidation System for Line Director

*Number of Functions/Use Cases : 59

Function Dependent	Function/Use Cases	RATING	WEIGHT	POINTS	Total
Ue	14.75	3	0.5	7.38	44.25
	29.5	6	0.75	22.13	
	14.75	12	1	14.75	
Uy	30	2	0.10	3	9.70
	20	4	0.20	4	
	9	12	0.30	2.7	
I	24	2	0.50	12	40.50
	26	4	0.75	19.5	
	9	8	1.00	9	
C	21	3	0.50	10.5	44.00
	18	6	0.75	13.5	
	20	12	1.00	20	
U	59	1	0.60	35.4	35.40
	0	3	1.00	0	
Df					306.32

Dynamic Test Point	Function/Use Cases	RATING	WEIGHT	POINTS	Total
FPf				1	1
Df				306.32	306.32
Qd	39	3	0.05	1.95	4.70
	15	4	0.10	1.5	
	5	5	0.25	1.25	
	0	6	0.50	0	
TPf					1439.71

Productivity Factor	Total
ΣTP	1439.71
FP	500
Qi	16
TP	1455.71

Environmental Factors	
Test Tools	2
Development Testing	4
Test Basis	3
Development Environment	2
Test Environment	1
Testware	2
SUM OF ENVIRONMENTAL FACTORS	14

Team Size and Management Tools	
Team Size	3
Management Tools	2

Primary Test Hours		Total
TP		1455.71
P		1.20
E		0.67
PT	Total	1164.57

Total Number of Test Hours		Total
PT		1164.57
PC		1.05
Total Test Hours	Person-Hour	1222.79
	Man-Month	7.94

Finding Cost: BDT

Person Hour Cost	Total Hours	Total Cost	Overhead Cost 60 %	Total Gross Cost
400	1222.79	489117.53	293470.52	782588.05

Actual Man Month 8

Fig. 2. Final MM and Cost of LDMACS

Case 2: CREL

*Project Name: CLIMATE RESILIENT ECOSYSTEMS AND LIVELIHOOD

*Number of Functions/Use Cases : 78

Function Dependent	Function/Use Cases	RATING	WEIGHT	POINTS	Total
Ue	19.5	3	0.5	9.75	58.50
	39	6	0.75	29.25	
	19.5	12	1	19.5	
Uy	70	2	0.10	7	8.80
	6	4	0.20	1.2	
	2	12	0.30	0.6	
I	68	2	0.50	34	42.00
	8	4	0.75	6	
	2	8	1.00	2	
C	72	3	0.50	36	41.00
	4	6	0.75	3	
	2	12	1.00	2	
U	78	1	0.60	46.8	46.80
	0	3	1.00	0	
Df					439.63

Dynamic Test Point	Function/Use Cases	RATING	WEIGHT	POINTS	Total
FPf				1	1
Df				439.63	439.63
Qd	71	3	0.05	3.55	4.35
	3	4	0.10	0.3	
	2	5	0.25	0.5	
	0	6	0.50	0	
TPf					1912.38

Productivity Factor	Total
ΣTP	1912.38
FP	500
Qi	16
TP	1928.38

Environmental Factors	
Test Tools	1
Development Testing	2
Test Basis	3
Development Environment	2
Test Environment	1
Testware	1
SUM OF ENVIRONMENTAL FACTORS	10

Team Size and Management Tools	
Team Size	6
Management Tools	2

Primary Test Hours		Total
TP		1928.38
P		1.20
E		0.48
PT	Total	1101.93

Total Number of Test Hours		Total
PT		1101.93
PC		1.08
Total Test Hours	Person-Hour	1190.09
	Man-Month	7.73

Finding Cost: BDT

Person Hour Cost	Total Hours	Total Cost	Overhead Cost 50 %	Total Gross Cost
500	1190.09	595042.86	297521.43	892564.28

Actual Man Month 6

Fig.3. Final MM and Cost of CREL

Case 3: HRMS

*Project Name: POLICE BUREAU OF INVESTIGATIONS (PBI) OF Human Resources Management System

*Number of Functions/Use Cases : 123

Function Dependent	Function/Use Cases	RATING	WEIGHT	POINTS	Total
Ue	30.75	3	0.5	15.38	92.25
	61.5	6	0.75	46.13	
	30.75	12	1	30.75	
Uy	80	2	0.10	8	18.60
	23	4	0.20	4.6	
	20	12	0.30	6	
I	90	2	0.50	45	73.00
	20	4	0.75	15	
	13	8	1.00	13	
C	115	3	0.50	57.5	64.00
	6	6	0.75	4.5	
	2	12	1.00	2	
U	123	1	0.60	73.8	73.80
	0	3	1.00	0	
Df					1143.21

Dynamic Test Point	Function/Use Cases	RATING	WEIGHT	POINTS	Total
FPf				1	1
Df				1143.21	1143.21
Qd	85	3	0.05	4.25	8.80
	33	4	0.10	3.3	
	5	5	0.25	1.25	
	0	6	0.50	0	
TPf					10060.23

Productivity Factor	Total
ΣTP	10060.23
FP	500
Qi	16
TP	10076.23

Environmental Factors	
Test Tools	1
Development Testing	2
Test Basis	3
Development Environment	2
Test Environment	1
Testware	1
SUM OF ENVIRONMENTAL FACTORS	10

Team Size and Management Tools	
Team Size	6
Management Tools	2

Primary Test Hours		Total
TP		10076.23
P		1.20
E		0.48
PT	Total	5757.85

Total Number of Test Hours		Total
PT		5757.85
PC		1.08
Total Test Hours	Person-Hour	6218.47
	Man-Month	40.38

Finding Cost: BDT

Person Hour Cost	Total Hours	Total Cost	Overhead Cost 50 %	Total Gross Cost
500	6218.47	3109237.15	1554618.57	4663855.72

Actual Man Month 39

Fig. 4. Final MM and Cost of HRMS

Case 4: PBI Website

*Project Name: POLICE BUREAU OF INVESTIGATIONS (PBI) OF PBIwebsite

*Number of Functions/Use Cases : 70

Function Dependent	Function/Use Cases	RATING	WEIGHT	POINTS	Total
Ue	17.5	3	0.5	8.75	52.50
	35	6	0.75	26.25	
	17.5	12	1	17.5	
Uy	65	2	0.10	6.5	7.80
	2	4	0.20	0.4	
	3	12	0.30	0.9	
I	63	2	0.50	31.5	37.75
	3	4	0.75	2.25	
	4	8	1.00	4	
C	59	3	0.50	29.5	39.00
	6	6	0.75	4.5	
	5	12	1.00	5	
U	70	1	0.60	42	42.00
	0	3	1.00	0	
Df					359.76

Dynamic Test Point	Function/Use Cases	RATING	WEIGHT	POINTS	Total
FPf				1	1
Df				359.76	359.76
Qd	64	3	0.05	3.2	3.95
	5	4	0.10	0.5	
	1	5	0.25	0.25	
	0	6	0.50	0	
TPf					1421.04

Productivity Factor	Total
ΣTP	1421.04
FP	500
Qi	16
TP	1437.04

Environmental Factors	
Test Tools	1
Development Testing	2
Test Basis	3
Development Environment	2
Test Environment	1
Testware	1
SUM OF ENVIRONMENTAL FACTORS	10

Team Size and Management Tools	
Team Size	6
Management Tools	2

Primary Test Hours		Total
TP		1437.04
P		1.20
E		0.48
PT	Total	821.16

Total Number of Test Hours		Total
PT		821.16
PC		1.08
Total Test Hours	Person-Hour	886.86
	Man-Month	5.76

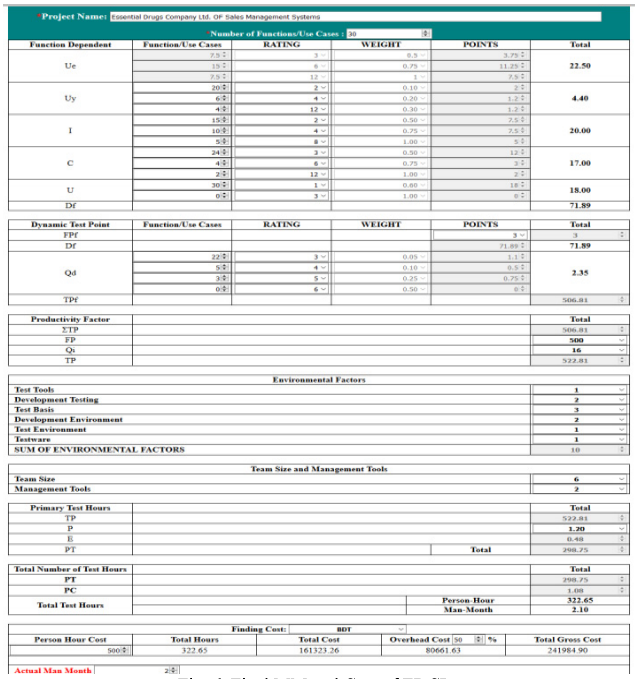
Finding Cost: BDT

Person Hour Cost	Total Hours	Total Cost	Overhead Cost 50 %	Total Gross Cost
500	886.86	443428.62	221714.31	665142.93

Actual Man Month 4

Fig. 5. Final MM and Cost of PBI website

Case 5: EDCL



Project Name: Essential Drugs Company Ltd. OF Sales Management Systems

Number of Functions/Use Cases : 30

Function Dependent	Function/Use Cases	RATING	WEIGHT	POINTS	Total
Ue	7.5	3	0.5	3.75	22.50
	15	6	0.75	11.25	
	7.5	12	1	7.5	
Uy	20	2	0.10	2	4.40
	6	4	0.20	1.2	
	4	12	0.30	1.2	
I	15	2	0.50	7.5	20.00
	10	4	0.75	7.5	
	5	8	1.00	5	
C	24	3	0.50	12	17.00
	4	6	0.75	3	
	2	12	1.00	2	
U	30	1	0.60	18	18.00
	0	3	1.00	0	
Df					71.89

Dynamic Test Point	Function/Use Cases	RATING	WEIGHT	POINTS	Total
FPf				3	3
Df				71.89	71.89
Qd	22	3	0.05	1.1	2.35
	5	4	0.10	0.5	
	3	5	0.25	0.75	
	0	6	0.50	0	
TPf					506.81

Productivity Factor	Total
ΣTP	506.81
FP	500
Qi	16
TP	522.81

Environmental Factors	
Test Tools	1
Development Testing	2
Test Basis	3
Development Environment	2
Test Environment	1
Testware	1
SUM OF ENVIRONMENTAL FACTORS	10

Team Size and Management Tools	
Team Size	6
Management Tools	2

Primary Test Hours		Total
TP		522.81
P		1.20
E		0.48
PT	Total	298.75

Total Number of Test Hours		Total
PT		298.75
PC		1.08
Total Test Hours	Person-Hour	322.65
	Man-Month	2.10

Finding Cost: BDT

Person Hour Cost	Total Hours	Total Cost	Overhead Cost 50 %	Total Gross Cost
500	322.65	161323.26	80661.63	241984.90

Actual Man Month 2

Fig. 6. Final MM and Cost of EDCL

V. COST ESTIMATION COMPARISON

COCOMOII (Constructive Cost Model) [13-15] cost models and sizing methods like Function Point Analysis (FPA) are well known and in widespread use in software engineering. As the authors are attempting to establish estimation methods, therefore, they have taken the best and proven method in the field of software engineering. From table 4, we can easily understand that the estimation by tool is more accurate if it compares to other estimation.

TABLE 4
COMPARISON OF TIME ESTIMATION

Project Name	*Estimated Time by Project Manager [MM]*	*Estimated Time with Research Tool [MM]*	*Estimated Time with COCOMOII Model [MM]*	*Actual Time Required [MM]*
LDMACS	6	7.94	10.68	8
CREL	4	7.73	9.76	6
HRMS	35	40.38	36.7	39
PBI Website	3	5.76	11.2	4
EDCL	2	2.1	4.1	1.5

VI. CONCLUSION

Though, the war against lowering the cost has not been over yet, however, the researchers has implemented the estimation tool to decrease the software cost. This paper demonstrates the software test estimation tool, which is clearly defined with project statistics, that demonstrates the time and cost of several projects. The authors have successfully developed the tools and 22 real projects were calculated. Web base test estimation tool is easy to access and more projects could be estimated with times with the guidelines. More projects statistics would contribute to make the tool remarkable and prove the necessity like other tools (Use Cases Point estimation). In the next phase, the researchers have planned to cover the Agile User Story Points [16] in order to estimate test projects.

REFERENCES

[1] E.P.W.M. Veenendaal, T. Dekkers, “Testpointanalysis: a method for test estimation”, *Project Control for Software Quality,* Shaker Publishing BV, The Netherlands, 1999.

[2] Y. Ahn, J. Suh, S. Kim and H. Kim, “The software maintenance project effort estimation model based on function points”, *Journal of Software Maintenance and Evolution: Research and Practice*, Vol. 15, pp.71–85, 2003.

[3] Function Point Counting Practices, Release 4.1, International Function Point User Group (IFPUG) 1994, Internet:http://www.bilkent.edu.tr/~csevgi/courses/ctis359/IFPUG_Counting_Practices_Manual_4.1.pdf [Access: May, 2013]

[4] Pathik, B. B, Islam, S., Khan, M. H. and Habib, M. M., “Software Test Estimation Tools using Use Cases and Functions”, Proceedings of the IEEE International Conference on Industrial Engineering and Engineering Management (IEEE IEEM), Bangkok, Thailand, 10-13 December 2013.

[5] Shaiful Islam, Bishwajit B. Pathik, Manzur H. Khan, Dr. Mamun Habib., “A Novel Tool for Reducing Time and Cost at Software Test Estimation: An Use Cases and Functions Based Approach”, Proceedings of the IEEE International Conference on Industrial Engineering and Engineering Management (IEEE IEEM), Malaysia, 9-12 December 2014.

[6] Bevan Nigel, “Quality in Use: Meeting User Needs for Quality”, *Journal of System and Software*, 1999.

[7] ISO/IEC TR 9126-2 (1997), Information Technology - Software Engineering Product Quality - Part 2: External metrics, International Organization of Standardization, Internet:http://www.cse.unsw.edu.au/~cs3710/PMmaterials/Resources/9126-2%20Standard.doc, [Access: April, 2013]

[8] The Application of Function Point Analysis in the Early Phases of the Application Life Cycle A Practical Manual, Theory and Case Study, Version 2.0, Manual of the Netherlands Software Metrics Users Association, Internet: http://www.nesma.nl/download/boeken_NESMA/N20_FPA_in_Early_Phases_(v2.0).pdf [Access: May, 2013]

[9] Leung, Hareton and Fan, Zhang, “Software Cost Estimation”, *Handbook of Software Engineering and Knowledge Engineering*, World Scientific Pub. Co, River Edge, Nj, 2002

[10] Chris F. Kemerer, “An empirical validation of software cost estimation models”, *Communications of the ACM*, Volume 30, Number 5, May 1987.

[11] Schneider, G. and Winters, J. "Applying Use Cases – A Practical Guide", Addison-Wesley. 1998.

[12] Software Estimation Best Practices. Internet: http://www.galorath.com/wp/estimating-best-practices.php, [Access: April, 2013]

[13] Grimstad, S. and Jorgensen, M., “A framework for the Analysis of software cost estimation accuracy”, *ISESE '06* Proceedings of the 2006 ACM/IEEE international symposium on Empirical software engineering, pp. 58 – 65.

[14] Lalit V. Patil, Rina M. Waghmode, S. D. Joshi, V. Khanna, “Generic model of software cost estimation: A hybrid approach”, 2014 IEEE International Advance Computing Conference (IACC),

[15] B. Boehm, C. Abts, A.W. Brown, S. Chulani, B.K. Clark, E. Horowitz, R. Madachy, D. Reifer, and B. Steece, “Software Cost Estimation with COCOMO II”. Prentice Hall, 2000.

[16] Pierre Bourque, Alain Abran, Eduardo Miranda, “Sizing User Stories Using Paired Comparisons”, Institute for Software Research, Carnegie-Mellon University and École de technologie supérieure – Université du Québec