

EXTREME SOFTWARE ESTIMATION (XSOFT ESTIMATION)

E. Karunakaran¹, P.Radhakrishnan²

^{1,2}Department of Computer Science & Engineering, Pondicherry Engineering College, Pillaichavady, Puducherry – 605014, India.

Abstract

The management of software cost, development effort and project planning are the key aspects of software development. Throughout the sixty-odd years of software development, the industry has gone at least four generations of programming languages and three major development paradigms. Still the total ability to move consistently from idea to product is yet to be achieved. In fact, recent studies document that the failure rate for software development has risen almost to 50 percent. There is no magic in managing software development successfully, but a number of issues related to software development make it unique. The basic problem of software development is risky. Some example of risk is error in estimation, schedule slips, project cancelled after numerous slips, high defect rate, system goes sour, business misunderstanding, false feature rich, staff turnover. XSoft Estimation addresses the risks by accurate measurement. A new methodology to estimate using software COSMIC-Full Function Point and named as EXtreme Software Estimation (XSoft Estimation). Based on the experience gained on the original XSoft project development, this paper describes what makes XSoft Estimation work from sizing to estimation.

Keywords: -COSMIC function size unit, XSoft Estimation, XSoft Measurement, Cost Estimation.

1. INTRODUCTION

Software Measurement : It is fundamental to any engineering discipline and software engineering is not an exception. Measurements in the physical world can be categorized in two ways namely a direct measure (eg. The length of bolt) and indirect measurement (e.g. the quality of the bolts). Software metrics can be categorized similarly. Direct measures of the software engineering process include cost and effort applied. Direct measures of the product include lines of code (LOC) produced, execution speed, memory size and defects reported over some set period of time. Indirect measures of the product include functionality, quality, complexity, efficiency, reliability, maintainability and many other abilities. Throughout the industry, the size oriented or/and function oriented metrics are used. Size-oriented metrics make use of the line of codes as a normalizing factor so that the size is calculated only after finishing the product. The function point is derived from measures of the information domain and a subjective assessment of problem complexity. Since functionality cannot be measured directly, it must be derived indirectly using other direct measures. So, the modified function points method which is simple to understand, easy to evaluate is known as COSMIC-FULL Function Point (2).

Estimation : In the early days of computing, software cost comprised a small percentage of overall computer-based system cost. An order of magnitude error in estimates of software cost had relatively little impact. Today, software is the most expensive element in most computer-based systems.

A cost estimation error can make the difference between profit and loss. Software cost and effort estimation will never be an exact science. To achieve reliable cost and effort estimates, a number of options arises namely :

1. Delay estimation until late in the project.
2. Base estimation on similar projects that have already been completed.
3. Use relatively simple “decomposition techniques” to generate project cost and effort estimates.
4. Use one or more empirical models for software cost and effort estimation.

Unfortunately, the first option, however attractive, is not practical. Cost estimates must be provided “up-front”. The second option can work reasonably well if the current project is quite similar to past efforts and other project influences (eg. The customer, business conditions, deadlines) is equivalent and also past experience has not always been a good indicator of future results. The third option decomposition technique takes a “divide and conquer” approach to software project estimation. By decomposing a project into major functions and related software engineering activities, cost and effort estimation can be performed in a stepwise fashion. The fourth option “empirical estimation models” can be used to complement decomposition techniques and offer a potentially valuable estimation approach in their own right.

2. RELATED WORK

[1] The COSMIC Method defines a standardized measure of software Functional Size expressed in CFP units. The

measurement is carried out by mapping the FUR of the software onto the COSMIC Generic Software Model (shown in Figure 1). The purpose of the measurement and scope of the software to be measured the level of decomposition and level of granularity of the software. As can be seen in Fig.1, there are four different data movement types. Entry type move data across the boundary from the user to the functional process. Exit type move data across the boundary to the user. Read type move data from persistent storage in the functional process. Write type move data from the functional process to persistent storage.

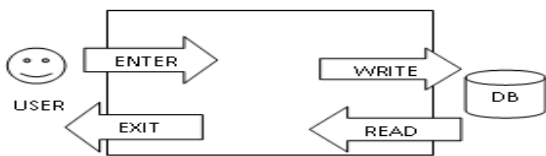


Fig 1 : Functional Process

[2] Research work on the quality of the documentation using a functional size method is recent. In this work documentation quality rating scale a, b, c, d, e is fixed based on facts, some of which can be listed as :

- The presence or absence of a data model.
- The presence or absence of information to identify the data movements (entry, read, write, exit).
- The presence or absence of documentation enabling identification of each functional process.

In this process three iterations are made for documentation quality and the quality rating are never decreasing from iteration to iteration. In the third iteration the documentation quality of a functional process is always equal to or higher than in the second and first iteration. At the end of third iteration only the highest rating like a or b will exists.

[3] A tool is developed for automated estimation of Code Size based on UML profile. The main functionalities of the tool are importing information modeled using the UML Profile and store component data needed for CFP measurement and Code Size estimation to calculate estimated code size using linear regression and present estimation results.

In [1] the enter type and exit type of a function process is considered as one cfsu (cosmic function size unit) irrespective of number of fields. The development time will vary based on the software used. So, sizing and estimation will not be accurate. In [2] based on the quality of the documentation, the size and estimation are done, but it is not always sure that the software development will be made according to the documentation. In the third and final iteration of documenting all the stories are highly rated, and in general the documentation is prepared at the end of the software development. So, the software may not be improved.

In [3] it is required to convert from COSMIC concept to UML concept. The tool is required to calculate and it is not easy to use and it is also hard to convince the customer about the size and estimation.

To make the measurement and estimation more accurate, some research is applied based on practical experience using COSMIC FFU and it is named as XSoft Estimation.

3. PROPOSED WORK

XSoft Measurement Process : The purpose and scope is to size the functionality which corresponds to the effort interested in. For an application, written partly in Java and in VB, measurement is made separately for software size and development effort. First identify the software layers, boundary, users, trigger, functional processes, data groups and sub-processes. In each of the above identify the number of entry, exit, read and write.

XSoft Estimation : A model is based on experience and takes the form

$$D = S * V$$

Where

D is one of the estimated value (eg. Effort, Cost, Project duration)

S is the size of the project and V is one of the variable values depends upon effort, cost, time estimation.

The following five different projects were taken to measure the size of the software and the result are given in the table : 1,2,3,4, and 5

Table: 1 - Project-1 : Mobile Park

Layers	Functional Process	Cfsu
Telecommunication	MobilePark Owner	14
	City Entry	14
	Zone Entry	15
	Policy Entry	15
	Loading Credits	15
	Effectted Parking	4
	Client Account	4
	Parking Policy Database	4
	Town Parking Tax	4
	Credit Purchase	4
	Upload	4
	Download	4
	Total	101

Table : 2 - Project-2 : EX-CARGO

Layers	Functional Process	Cfsu
MIS	Shipment Details	29
	Consignor / Consignee Details	17
	Inbound/Outbound	22
	Report-Finance	22
	Report-Planning	10
	Booking Details	13
	Shipment Status	10
	Planning by Truck	10
	Status Report	21
	Shipment Sold	19
	Cosignee Reverse	14
	Account Details	12
	Total	199

Table : 3 - Project-3 : L-SALES

Layers	Functional Process	Cfsu
MIS	Estimation	22
	Sales	374
	Credit	12
	Cash	48
	File	67
	User	15
	Invoice	25
	Rate	87
	Total	313

Table : 4 - Project-4 : DDE

Layers	Functional Process	Cfsu
Tele Comm.	OPCGROUP	63
	EnumOPCItemAttributes	8
	OPCServer Object	27
	OPCBrowser Object	18
	OPCGroups Object	14

Table : 6 Man days requirement – Comparison between Expert and Skilled teams

Functional Process	Actual Size (Cfsu)	Allotted Size (Cfsu)		Actual Man Days (5 Cfsu)	
		E-TEAM	S-TEAM	E-TEAM	S-TEAM
MobilePark Owner	14	7	7	7	10
City Entry	14	7	7	7	10
Zone Entry	15	8	7	8	9
Policy Entry	15	8	7	9	9
Loading Credits	15	8	7	8	9
Parking details	20	10	10	11	14
Upload & Download	8	4	4	5	6
AVERAGE	14.42	7.42	7	7.86	9.57

Project Name : Mobile Parking,

Layer : Tele. Communication

Software Used : Java

	OPCGroup Object	12
	OPCItems Object	18
	OPCItem Object	5
	IOPCCCommon	15
	IOPCEventServer	45
	IOPCCConnectionPointContainer	5
	OPCHDAServer	24
	OPCHDABrowser Object	15
	OPCHDAItems Object	72
	Total	341

Table : 5 - Project-5 : FORGING

Layers	Functional Process	Cfsu
MIS	Receipts	16
	Issues	16
	Stock	11
	Ledge & Opening Balance	20
	RM details	16
	Queries – Ledger details	38
	Inventory Maintenance	9
	RM, TR Entry Details	10
	TDC & TC Entry	14
	Forging Entry & Backup	18
	Queries – Forging details	20
	Transfer Heatcode, ICIN, DN Modification & ReIndexing	18
	IOPCEventServer	45
	Reports	50
	Total	260

4. RESULTS AND DISCUSSION

The five different projects were used and find the man days required to finish the functional process of each project between the skilled and expert teams. As well as man days required between various layers like MIS and Tele Communication. The results are given in the table : 6, 7, 8 & 9.

Table : 7 Man days requirement – Comparison between Expert and Skilled teams

Functional Process	Actual Size (Cfsu)	Allotted Size (Cfsu)		Actual Man Days (5 Cfsu)	
		E-TEAM	S-TEAM	E-TEAM	S-TEAM
Estimation	22	12	10	1.7	3
Sales	32	17	15	1.8	2.8
Credit	12	7	5	1.2	2
Cash	48	26	22	1.9	2.7
File	67	37	30	2.1	2.6
User	15	9	6	1.3	2.2
Invoice	25	15	10	1.4	2
Rate	87	47	40	1.6	2.5
AVERAGE	38.5	21.25	17.25	1.625	2.475

Project Name : LIPS,

Layer : MIS,

Software Used : VB6

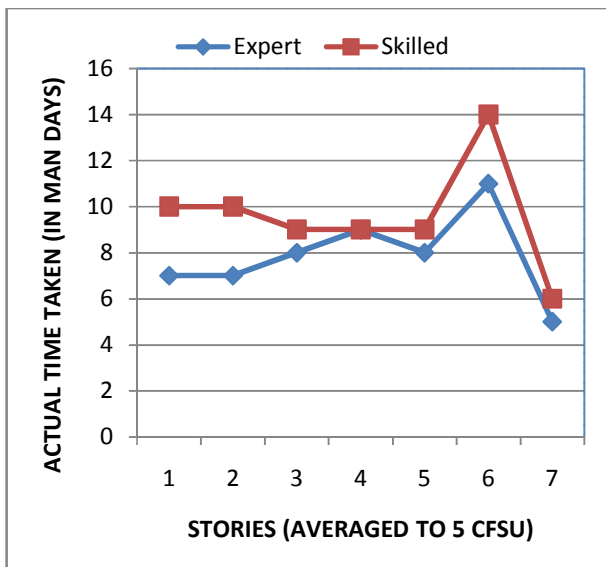


Fig. 2 : Expert Vs Skilled (Telecommunication)

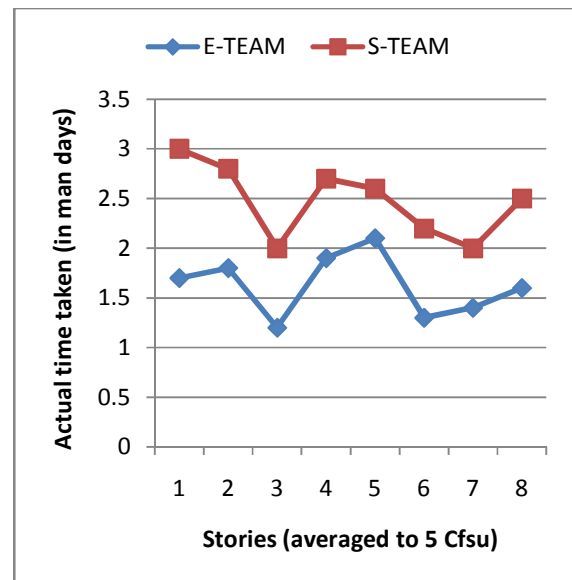


Fig.3 : Expert Vs Skilled (MIS)

Table : 8 Man days requirement – Comparison between Layers

SOFTWARE	Actual Size (Cfsu)		Actual Man Days (5 Cfsu)		Actual Time Taken (in Man Days)	
	MIS	T.C.	MIS	T.C.	MIS	T.C.
STORY - 1	22	14	2.95	10.71	13	30
STORY - 2	37	14	2.43	10.00	18	28
STORY - 3	12	15	2.92	9.33	7	28
STORY - 4	48	15	2.81	8.67	27	26
STORY - 5	67	15	2.76	8.67	37	26
STORY - 6	15	20	4.33	9.00	13	36
STORY - 7	25	8	2.60	8.75	13	14
STORY - 8	87		2.76		48	
AVERAGE	38.5		3.37	9.30	22	26.86

Project Name : LIPS & MOBI LE PARK

Layer : MIS & T.C.

Software Used : VB6 & Java

Table : 9 Time taken – Comparison between different Software / Layers

SOFTWARE	Actual Size (Cfsu)				Actual Time Taken (Man Days for 5 Cfsu)			
	VB	JAVA	JAVA	C++	VB	JAVA	JAVA	C++
	MIS	MIS	T.C.	T.C.	MIS	MIS	T.C.	T.C.
STORY - 1	22	16	14	16	2.95	5.94	10.71	10.00
STORY - 2	37	27	14	15	2.43	5.19	10.00	9.33
STORY - 3	12	36	15	14	2.92	5.14	9.33	10.00
STORY - 4	48	38	15	14	2.81	4.74	8.67	9.29
STORY - 5	67	19	15	17	2.76	5.53	8.67	10.00
STORY - 6	15	32	20	10	4.33	5.00	9.00	11.00
STORY - 7	25	39	8	16	2.60	4.87	8.75	9.38
STORY - 8	87	50			2.76	4.00	9.31	9.80
AVERAGE	44.71	36.71	14.43	14.57	2.46	4.27	9.31	9.80

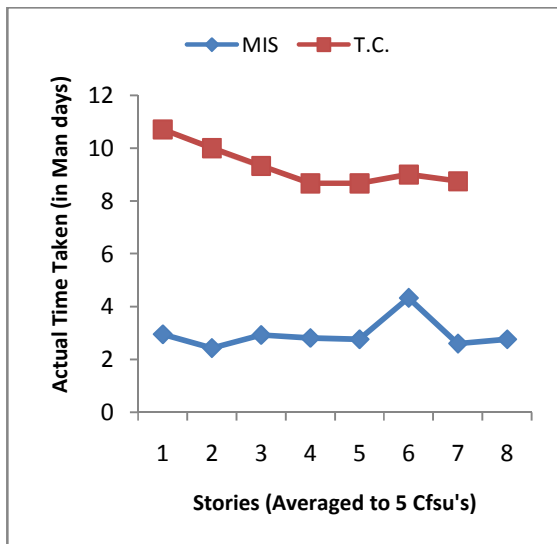


Fig. 4 : MIS Vs Telecommunication

The results show that the VB requires half the time to complete the project of size 5 Cfsu than the time required for Java. Also it is very clear that the time required for Java in MIS is almost half of the time required in telecommunication system. (Fig. 4).

The time effort for 1 Cfsu for different software and different layers are estimated and furnished with table : 10.

Table : 10 Time effort – Comparison between different Software / Layers

Layers/ Software	VB 6	C++	JAVA
MIS	0.5	1.2	0.9
Tele Communication	---	2.0	1.9

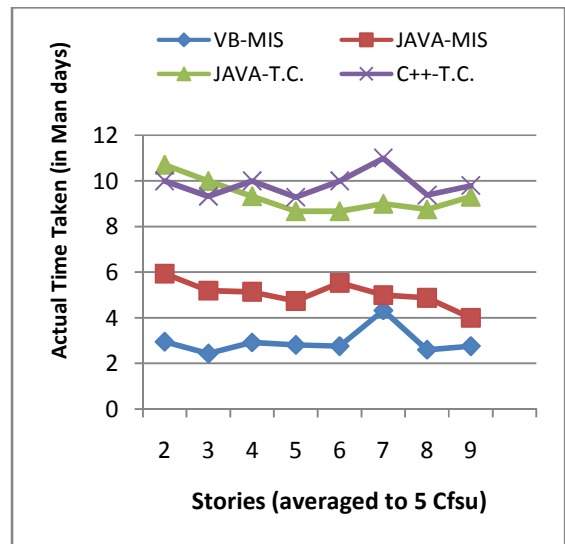


Fig. 5 : Between Softwares (VB, Java, C++)

It is evident from the above table that the VB software requires very less effort time to develop a Management Information System followed by Java. In case of Tele Communication C++ may be effective software language to optimize time effort. (Fig. 5).

Finally, a new formula is proposed to determine the effort estimation to arrive at optimum cost to develop the software.

$$\text{Effort Estimation} = SS * (SL + SU + DU)$$

Where SS – Software Size (using COSMIC – FFP)

SL – Software Layer

SU – Software Used

DU – Database Used

The effort value for different software, layers and databases are furnished in the table : 11. The effort values given in the table are based on the effort valued of the five projects used for the present study.

Table : 11 Time effort – Comparison between different Software / Layers

SL/SU/DU	Effort (for 1 Cfsu)
SL --- MIS Tele Communication	0.3 1.4
SU --- VB Java C++	.2 .4 .6
DU --- Oracle MS-Acess My-SQL	.100 .050 .075

Cost Estimation = Effort Estimation * Cost per Man day.

The cost per man day will vary from company to company and the cost estimation for the present study is based on the cost per man day fixed by this Company is given in table :12.

Table : 12 Cost Estimation Details

Sl.no.	Description	Hourly	Monthly
1	Developer, Tester, Technical Writer	400	50,000
2	Consultant, System Analyst, Module Leader, System Administrator	600	75,000
3	System Architect	800	1,00,000
4	Project Manager, Senior Consultant	1200	1,50,000

5. IMPLEMENTATION – A CASE STUDY

A Case study is undertaken to explain the implementation process which is shown in Fig 6. The various stages using XSOFT Measurement are detailed below

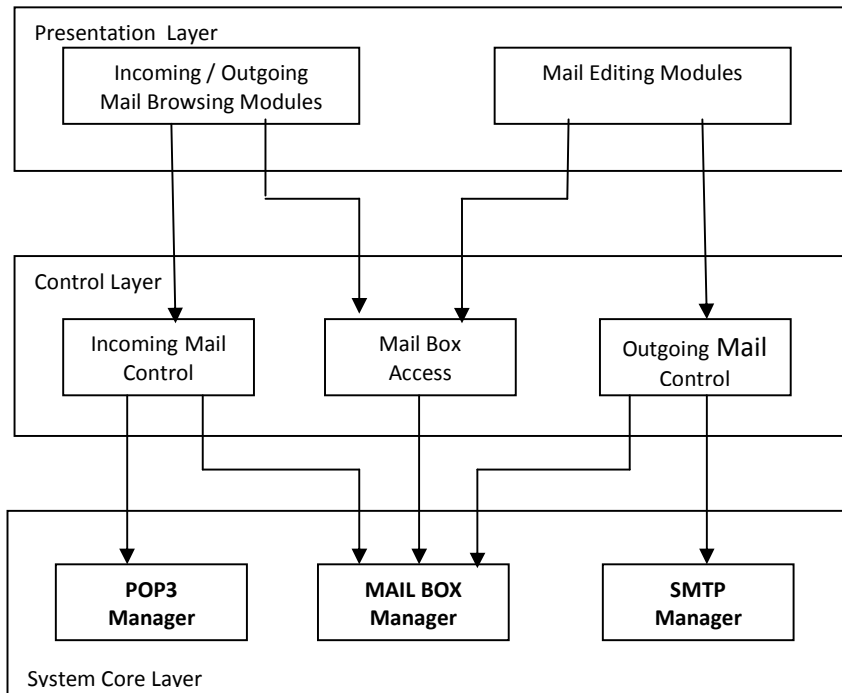


Fig. 6 E-Mail Server Block Diagram

5.1 Purpose and Scope

E-mail component is the core of all Linux-based e-mail applications. The main functionality of the component is to process conversations with remote mail servers by implementing the most popular email protocols, POP3, SMTP, MIME, etc. in internet. The component also contains the functionality to store local mail boxes for both incoming and outgoing messages.

MAPPING PHASE I

Triggering Events & Functional Processes

SMTPObject
 POP3 Object, Message List Object, Recipient List Object,
 Attachment List Object & Mail Box Object
 Data Group
 Senders Details & Receivers Details

MAPPING PHASE II

Sub-Processes

SMTPObject - Add Attachment, Add Recipient, Cancel & Send
POP3 Object - Cancel, Connect, Disconnect, Download Header, GetDownloadableCount, GetDownloadableSize & Download Messages.
Message List Object - Get, Delete & DeleteAll.
Recipient List Object -Get Recipient, Attachment List Object & Get Attachment.
Mail Box Object - Open, Close, GetHeaders, GetMessage, GetCount , Append & Delete.

Table : 13 Aggregate Result (Measurement)

Layers	Functional Process	Total Cfsu
Tele Communication	SMTPObject	6
	POP3 Object	21
	Message List Object	8
	Recipient List Object	2
	Attachment List Object	2
	Mail Box Object	13
	Total	52

5.2 Estimation

The size of the Software = 52 Cfsu
 Total Man days = 52 * 2 =104 (since 1 Cfsu = 2 man days)
 Developers = 2 Skilled and 2 Beginners
 Human Resource = 2 * 1 + 2 * .5 = 3

Estimated time to finish the project with 2 teams
 (2 developer in one team) = 104 / 3 = 35 Working days.
 Cost Estimation = 104 * 3200 = 3,32,800

CONCLUSIONS AND FUTURE WORK

In the present study an attempt has been made to address the risk involved in software development with the help if the proposed methodology which is developed with the features of COSMIC FFP. Based on the extensive studies carried out during the present, the following conclusion are drawn. It is evident from the results that the man days required for the expert team to complete either the MIS or Tele Communication project are significantly less than the required by the skilled team.

The software layer is also one the factors influencing the man days required to complete the project. The man days required to complete MIS project is almost three times less than the telecommunication project. Another factor influencing the man days is the type of software used for the project. VB requires half the time to complete the project than the time required for Java.

The effort time required to complete the MIS using VB is significantly less than Java and C++. In case of telecommunication systems C++ may be effective to optimize time effort than VB and Java. A formula was developed to determine the effort estimation to arrive at the optimum cost to develop software.

Using the previous information about the developers (i.e.. the man days to complete one Cfsu in various Layers / Software) the company can estimate the accurate cost estimation and time taken to finish the project.

REFERENCES

[1]. ISO/IEC. 2003 19761: Software Engineering COSMIC FFP: A Functional Size Measurement Method, 2003.
 [2]. Jean-Marc Desharnais, Computer Engineering Department, Bogazici University, Istanbul, Turkey, Using the COSMIC Method to Evaluate the Quality of the Documentation of Agile User Stories : 2011 Joint Conference of the 21st International Workshop on Software Measurement, pp. 269-272.
 [3]. Kenneth Lind, Electrical Systems Engineering, Saab Automobile AB, Trollhattan, Sweden, CompSize : Automated Size Estimation of Embedded Software Components, 2011 Joint Conference of the 21st International Workshop on Software Measurement, pp. 86-95.
 [4]. Joseph C.Giarratano, Gary D.Riley, *Expert Systems Principles and Programming*. Beijing: China Machine Press, 2006.
 [5]. Z. Li, L.X.Lin, Y.B. Cao, *Method and Practice of Function Point Analysis*. Beijing: Tsinghua University Press, 2005.
 [6]. J.D. Zhu, C.H. Qi, X.M. Hou, "Estimating software cost based on grey theory", *Computer Engineering and Applications*, vol. 46, no. 16, pp.71-73, 2010.

Albrecht, A.: Measuring application development productivity. Proc. of the IBM Applications Development Symposium, Monterey, CA, 83-92, Oct. 1979.

[7]. Albrecht, A, and Gaffney, J.: Software function, source lines of code, and development effort prediction: A software science validation. IEEE Trans. Softw. Eng. SE-9, 6, 639–648, 1983.

[8]. A. Abran, J.M. Desharnais, A. Lesterhuis, B. Londeix, R. Meli, P.M.Jorgensen, “A Review of Studies on Expert Estimation of Software Development Effort”, *Journal of Systems and Software*, vol. 70, no.1/2, pp. 58-65, 2004.

[9]. E. Miranda, *Sizing User Stories Using Paired Comparisons*, Information and Software Technology (IST) Journal, vol. 51, issue 9, September 2009, pp. 1327-1337.

[10]. Haugen N.C., *An Empirical Study of Using Planning Poker for User Story Estimation*, AGILE 2006 Conference (AGILE 2006), 23-28 July, 2006, Minneapolis, Minnesota, USA. IEEE Computer Society 2006, pp. 23-34.

[11]. Extreme Programming on PWAP (fictitious) a wireless application, Paul Rodrigues, E.Karunakaran, Naturesoft Eolution Provider, Chennai, India. Agile International Conference, Italy, 2001, pp. 73.

[12]. COSMIC – Common Software Measurement International Consortium 2007. The COSMIC Funtional Size Measurement Method – version 3.0 Measurement Manual (The COSMIC Implementation Guide for ISO/IEC 19761:2003), September 2007.