eXtreme Software Development (XSoft)

E. Karunakaran, P.Radhakrishnan

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 addresses the risks mentioned above.. XSoft is a new methodology to manage software development successfully has been evolved and it is derived from Extreme Programming and COSMIC-Full Function Point and named as EXtreme Software Development (XSoft). XSoft covers the reasoning behind the XP process. Based on the experience gained on the original XP project, this paper describes what makes XSoft work, day to day and month to month from estimation to release. Successful software development is a team effort – not just the development team but the larger team consisting of customers, management and developers. XSoft is a simple process that brings these people together and helps them to succeed together. XSoft facilitates the needs of customers with the abilities of programmers and steering (managing the project to success). A complete case study gives the idea of developing software by XSoft methodology. A XSoft tool is developed to manage the XSoft projects.

Keywords- COSMIC function size unit, XSoft sizing, XSoft Estimation, XSoft Measurement, Cost Estimation, XSoft Practices, XSoft Planning, XSoft Construction, Pair Programming.

1 INTRODUCTION

XSoft is deliberate and discipline approach to software development. It was derived from Extreme Programming & COSMIC Full Function Points (CFFP). XSoft is successful because it stresses customer satisfaction. The methodology is designed so as to deliver the software to the customer when it is needed. It also emphasizes team work and implements a simple and effective way to enable groupware style development. XSoft improves a software project in four essential ways : communication, simplicity, feedback and courage. XSoft programmers communicate with their customers and fellow programmers. They keep their design simple and clean. They get feedback by testing their software starting from day one. XSoft consists of XSoft Measurement, XSoft Estimation, XSoft Practices, XSoft Planning and XSoft Construction.

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

- *E.Karunakaran is currently pursuing Ph.D. degree program in Computer Science and Engineering and working in Pondicherry Engineerig College, Pillaichavady, Puducherry, India.,. E-mail: ekaruna@pec.edu*
- *P.Radhakrishnan is currently pursuing Ph.D. degree program in Computer Science and Engineeringand working in Pondicherry Engineerig College, Pillaichavady, Puducherry, India. E-mail: priyanaveenpk@pec.edu*

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 one or more empirical models for software cost and effort estimation.
4. Use relatively simple "decomposition techniques" to generate project cost and effort estimates.

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.

Software Development : A software development methodology is a framework that is used to plan and control the process of developing. The existing common methodologies include waterfall, prototyping, iterative & incremental development, spiral development, rapid application development, and extreme programming. A methodology can also include aspects of the development environment (i.e. IDEs), model-based development, computer aided software development, and the utilization of particular frameworks (i.e. programming libraries or other tools).

Further the paper is organized as follows : Section2 describes the related work and the limitations observed. Section 3 narrates the proposed work; Section 4 is about the results and discussion. Section 5 details with a case study and Section 6 conclude the study with proposed future work.

2 RELATED WORK

The COSMIC [1] Method defines a standardized measure of software Functional Size expressed in CFP units. The measurement is carried out by mapping the Function User Requirement (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.

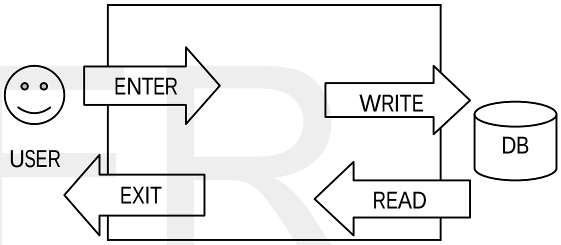

Fig. 1 Functional Process

Jean-Marc Desharnais [2] in his recent study on the quality of the documentation using a functional size method, 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. The quality rating are increasing from iteration to iteration. In the third iteration the documentation quality of a functional process is always equal or higher in the second and first iteration. At the end of third iteration only the highest rating like a or b will exists.

Kenneth Lind et. al [3] developed a tool 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 CFFP measurement and Code Size estimation to calculate estimated code size using linear regression and present estimation results.

Paul Rodrigues et al. [4] developed a software using extreme programming practices as a discipline because of simplicity, error free, communication flow, focus on programming and unit testing. Naturesoft applied XP to the project named "PWAP" a wireless application. The actual development consists of requirement gathering, estimation, iteration planning, standup meeting and release plan.

Limitation in the related work : In ISO/IEC 2003 standard for COSMIC FFP [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 and number of fields, in which the sizing and estimation may 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 according to the document iteration. 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. It is also hard to convince the customer about the size and estimation based on this method. In [4] XP concept is applied on a single project and the experience is shared.

To address the above limitations a study is proposed to define a generic software development method by integrating COSMIC FFP and XP.

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. Mod of 5 fields take one Cfsu for each group. Number of Cfsu is calculated for different layers and the same is used for estimation.

XSoft Estimation : A model is developed based on the experience obtained from XP projects which 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 value 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
TC	MobilePark Owner	14
	City Entry	14
	Zone Entry	15
	Policy Entry	15
	Loading Credits	15
	Effected 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
	OPCGroup Object	12
	OPCItems Object	18
	OPCItem Object	5
	IOPCCommon	15
	IOPCEventServer	45
	IOPCConnectionPointContainer	5
	OPCHDAServer	24
	OPCHDABrower 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 allotted to the pair programmers of skilled team and expert team. The actual man days required to finish the stories by the teams between various layers like Management Information System (MIS) and Telecommunication (TC) are measured and the results are given in the table : 6, 7, 8 & 9. Also Fig. 2 to Fig. 5 are the charts drawn using the data from the table 6 to table 9 respectively. The figure gives more clear view of the man days taken to develop stories between expert Vs skilled with various layers and softwares.

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

(* The stories Effected Parking, Client Account, Parking Policy Database, Town Parking Tax, Credit Purchase are merged and made as one story namely Parking details)

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

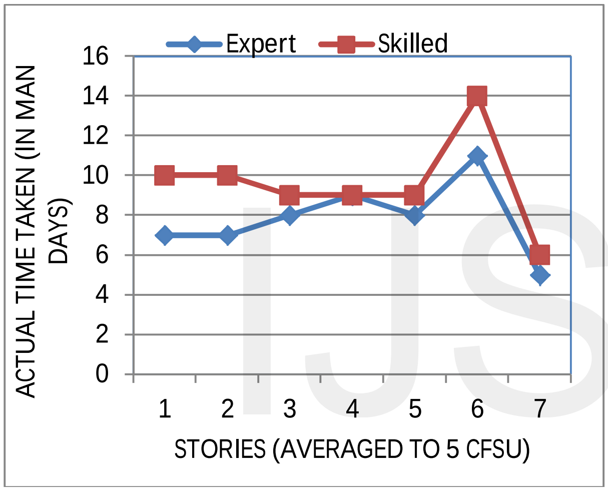

Fig. 2 : Expert Vs Skilled (TC)

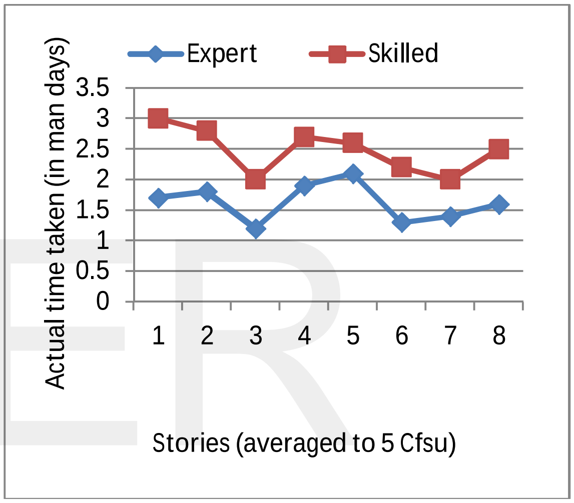

Fig.3 : Expert Vs Skilled (MIS)

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

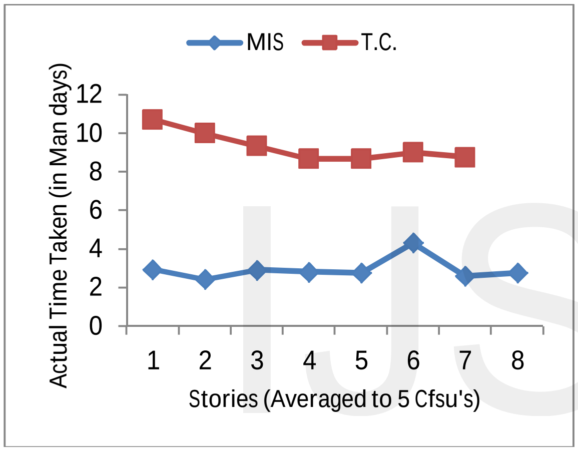

Fig. 4 MIS Vs Telecommunication

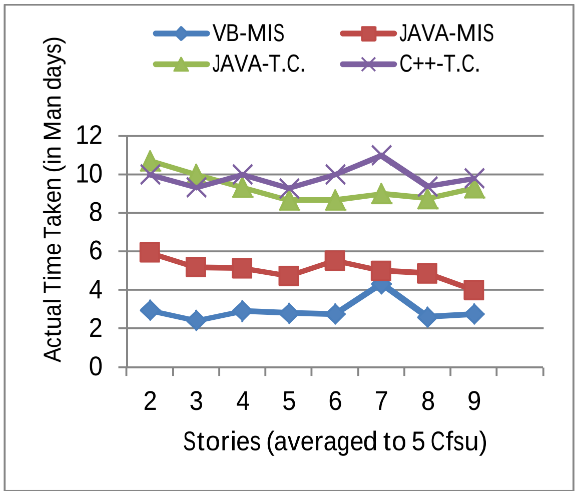

Fig. 5 Between Softwares (VB, Java, C++)

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.2 and Fig. 3 clearly indicates Expert Programmer out performed over Skilled Programmer. The time effort for 1 Cfsu for different software and different layers are estimated and furnished in the table:10.

TABLE 10
Time Effort – Comparison Between Different Software / Layers.

Layers/ Software	VB 6	C++	JAVA
MIS	0.5	1.2	0.9
TC	2.2	2.0	1.9

It is evident from the above (Fig. 5) and table 10, 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. Finally, a new simple formula is proposed to determine the effort estimation to arrive at optimum cost to develop the software.

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 have been calculated and 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	0.3
Telecommunication	1.4
SU ---	
VB	.2
Java	.4
C++	.6
DU ---	
Oracle	.100
MS-Acess	.050
My-SQL	.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

XSOFT PRACTICES
The following practices were under taken based on XP projects[6] :

a) ON SITE CUSTOMER : A XSoft project needs a full-time customer to provide guidance. Programmers will be able to program anything. The trick is to tell them what's needed. The customer really is too valuable to be on-site.

b) PAIR PROGRAMMING : Two programmers sitting together at the same machine write all production code and better code than the same two programmers working separately. They can work longer without getting tired, and when they're finished, two people understand everything, instead of understanding just half of everything.

c) CODING STANDARDS : One can always read their own code. But wait, it's all their own code". All will code to an agreed coding standard. This ensures that the code communicates as cleary as possible and supports our shared responsibility for quality everywhere. Some of the key topics one has to think about in our standard are Indentation, Capitalizaiton, Commenting, Method Size and Names.

XSOFT PLANNING
The following planning are done under XSoft [7]:

a) MAKING STORIES : Customers have the right to get the most possible value out of every programming moment by asking for small atomic bits of functionality. Programmers have the right to know what is needed. These two rights come together in the user story.

b) SETTING PRIORITIES : In each release cycle, the customer controls scope deciding what to do and what to defer to provide the best release by the due date.

c) SYSTEM METAPHOR : It helps to keep the design simple an clear and helps programmers guess what needs to be done and how. When working without a clear metaphor (which probably half of all project do), expect to need more diagrams, more design discussions, and more documents.

d) RELEASE PLAN : In each release cycle, the customer controls scope, deciding what to do and what to defer, to provide the best possible release by the due date.

e) ITERATION PLAN : Inside each release, an Extreme team plans just few weeks at a time withclear objective and solid estimates.

Pair Programming / Making Stories :
To find the effort and quality of the programmers, when they work in pair and sole two real projects are taken and tested. In this method almost equal capable programmers were taken for development. Using XP concept [1] the project is made as stories and each story is splited into tasks. Now using COSMIC Full Function Point, Cfsu (COSMIC function size unit) [2] were found. The stories and size of the stories of the projects are listed in table : 13 and table : 14. Now story by story the tasks were allotted to the pair and solo programmers

equally on their choice. The time taken by them to finish the tasks of the each story are listed down in the table 15 and table 16. Using the data the charts are created which are fig. 6 and fig. 7.

TABLE 14
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 15
Man Days Requirement – Comparison Between Pair and Solo Programmer

Functional Process	Actual Size (Cfsu)	Allotted Size (Cfsu)		Actual Man Days (5 Cfsu)	
		PAIR PROGRAMMERS	SOLO PRORAMMER	PAIR PROGRAMMERS	SOLO PRORAMMER
MobilePark Owner	14	9	5	9	11
City Entry	14	9	5	8	10
Zone Entry	15	10	5	8	10
Policy Entry	15	10	5	9	11
Loading Credits	15	10	5	8	11
Parking details	20	13	7	10	14
Upload & Download	8	5	3	5	7
AVERAGE	14.42	9.43	7	8.14	10.57

Project Name : Mobile Parking, Layer : Tele. Communication
Software Used : Java

TABLE 16
Man Days Requirement – Comparison Between Pair and Solo Programmer

Functional Process	Actual Size (Cfsu)	Allotted Size (Cfsu)		Actual Man Days (5 Cfsu)	
		PAIR PROGRAMMERS	SOLO PRORAMMER	PAIR PROGRAMMERS	SOLO PRORAMMER
Estimation	22	15	7	3	4
Sales	32	21	11	5	6
Credit	12	8	4	3	3
Cash	48	32	16	8	9
File	67	44	23	11	13
User	15	10	5	2	3
Invoice	25	17	8	2	4
Rate	87	58	29	11	15
AVERAGE	38.5	25.63	12.88	5.63	7.13

Project Name : LIPS, Layer : MIS, Software Used : VB6

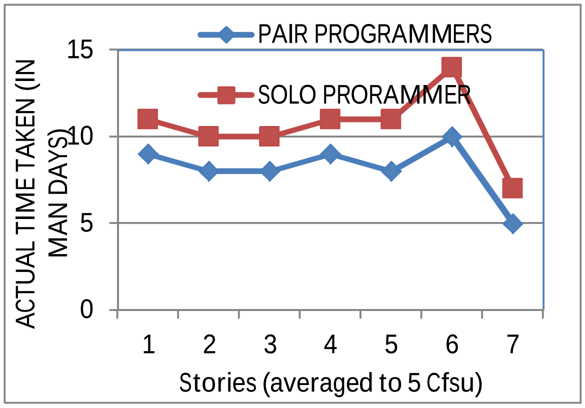

Fig. 6 Pair Vs Solo (TC)

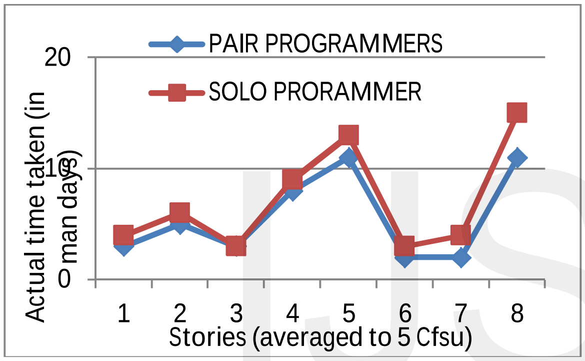

Fig. 7 Pari Vs Solo (MIS)

The Fig. 6 and Fig. 7 clearly shows that pair programmers are continuously performed well over solo programmer. Even though very limiit number of errors occurred during integration, among the errors more than 80% of the errors occurred are from solo programming. From this it is clear that the quality of programming is very good in pair programming. Actually the pair programmer must take 50% of time of solo programmer since pair programmer has two programmer. But the result shows that the pair programming takes only 35 % time of Solo programming instead of 50%. In addition from the table it is clear 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. Even though the lays and development software are changed, the pair programmer is always performanced far better than solo programmer. Since it is a practical method it gives more accurate results. A easy understandable working model is designed and presented in the Fig. 8 which is the eXtreme Software Development Process.

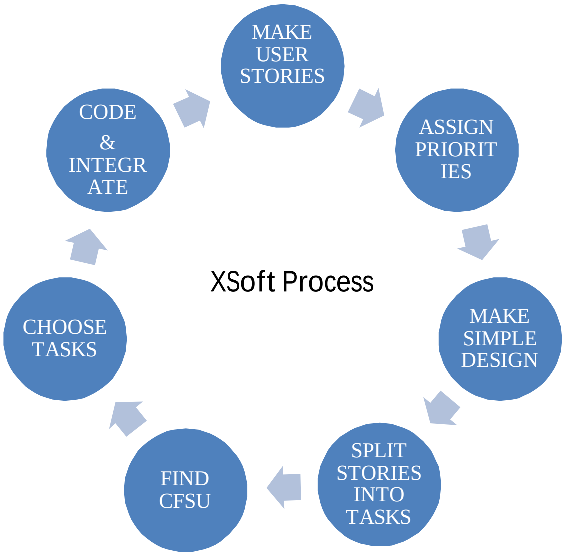

Fig. 8 eXtreme Software Development Process

An algoritham for eXtreme Software Development is presented below:

Algorithm for XSoft

Step : 1 Get User Stories from the Custormer.

Step : 2 Assign priorities to the Strories.

Step : 3 Draw a simple over all design (Metaphor).

Step : 4 Take one story at a time and repeat Step:5 to Step:7 until all the Stories are over.

Step : 5 Split Story into Tasks.

Step : 6 Find Cfsu for each task.

Step : 7 Choose tasks by the pair programmer on their choice.

Step : 8 Create a Schedule for the stories using the assignment of task to the programmers.

Step : 9 Code by coding standards, Integrate and test.

Step: 10 Deliver according to the priority Schedule.

5 IMPLEMENTATION – A CASE STUDY

A Case study is undertaken to explain the implementation process. The Fig. 9 shows that a block diagram of E-Mail Process. The various stages using XSOFT are detailed below :

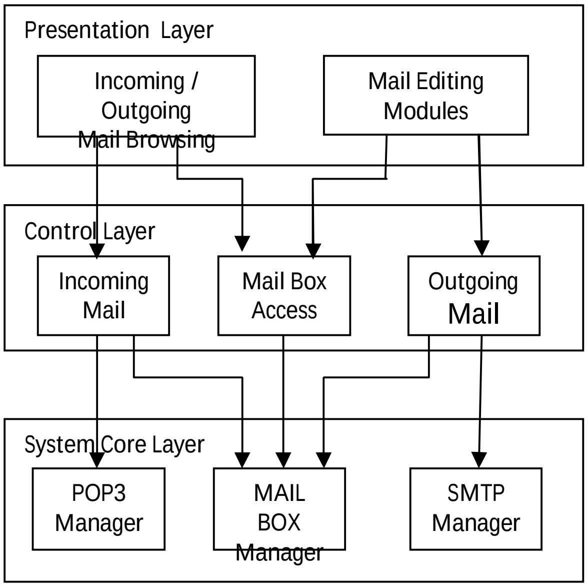

Fig.9 E-Mail Process Block Diagram

5.1 XSOFT MEASUREMENT

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 17
SMTPObject Measurement

Name	Data Movement	Type	Cfsu
Add Attachment	Read from the Source	Read	1
	Display the filename	Exit	1
Add Recipient	User entry	Entry	1
	Display error message	Exit	1
Cancel	Display blank	Exit	1
Send	Error message	Exit	1
Total			6

TABLE 18
Aggregate Result (Measurement)

Layers	Functional Process	Total Cfsu
TC	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 XSOFT 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

5.3 XSOFT PRACTICES

The following practices were followed :

a) ONSITE CUSTOMER :

The Client once visited at the project launch and described the complete stories of the project. In this project the Client was not physically present, but always available on call. Each story was noted on a separate cards and the task estimation exercise was done as lots were developed for multiple iterations. A User Case Diagram was drawn for the complete project. There are 3 User Cases in this project.

b) PAIR PROGRAMMER :

Team A – Mr. Sankar & Mr.Raghul
Team B – Mr. Hari Prasanth & Mr. Muruganadham

c) CODING STANDARDS :

C++ coding standards of version 3.0 was used for present project and the site address is www.soft32.com These standards represent a good set of off-the-shelf construction guidelines for C++ development. Variables should be made up of full words but not shortened. Eg. numberOfPoints, not NumPts or np.

5.4 XSOFT PLANNING

In XSoft Plannig the project is made into user stories and loaded into the XSoft tool. The user stories are prioritised by the customer, a simple and clear UML design Fig. 10 is made for the project. A iteration plan is made for each stories and tasks are allotted to the pair programmers, their starting and ending dates of the stories are listed in the table 22.

a) MAKING USER STORIES

Story-1
The user can receive mail from the server and store the detail in the local machine for further reference. The user can work simultaneously and should have the provision to abort the receiving mail.
Story-2
The user can send mail from the server and store the detail in the local machine for futher reference. The user can work simultaneously and should have the provision to abourt the sending mail.
Story-3
In multitasking environment, access local mail boxes can be done simultaneously, especially when C1/C2 and C4 occur t the same time. Where C1-Retrieve messages from local mail boxes. C2-Delete messages received from remote servers to local mailboxes. C4-Append sending messages to local mailboxes after they have been successfully sent to remote servers.

b) MAKING PRIORITIES

The customer decides the priorities as per his requirements. For the present case study the priorities are has listed below:

Story-1
Story-2
Story-3

c) METAPHOR (SYSTEM DESIGN)

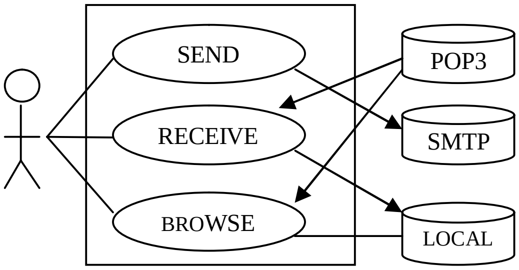

Fig. 10 System Design

d) ITERARATON PLAN

ITERATION PLAN -1

STORY-1 : Receive Mail using POP3
To be stored in Local Mail Box.
Work in asynchronous mode.
It should have Abort provision.

TABLE 19
MAKING TASKS

1.	Connect (Connects to POP3 Server)	4 Cfsu
2.	Cancel (Cancel downloading of messages)	1 Cfsu
3.	Disconnect (Disconnect from connected POP3)	3 Cfsu
4.	Get downloadable count (No of messages)	3 Cfsu
5.	Get downloadable size	3 Cfsu
6.	Download message	3 Cfsu

ITERATION PLAN -2
STORY-2 : Send mail by either composing or
attaching file using SMTP.
Store the send mail into local box.
It should work in asynchronous mode
with abort option.

TABLE 20
MAKING TASKS

1.	Compose Mail	7 Cfsu
2.	Add Recipient	2 Cfsu
3.	Add Attachment	3 Cfsu
4.	Cancel	1 Cfsu
5.	Send (includes connect & disconnect)	7 Cfsu
6.	Forward	2 Cfsu
7.	Reply	2Cfsu
8.	Reply All	2 Cfsu

ITERATION PLAN -3
STORY-3 : To access(create, delete, modify,

update) the incoming & outgoing mail.
Stored in mail box.
Simultaneous access in multitasking environment.

TABLE 21
MAKING TASKS

1.	Open Mail Box	2 Cfsu
2.	Close	1 Cfsu
3.	Get headers	2 Cfsu
4.	Get message	2 Cfsu
5.	Get count	2 Cfsu
6.	Append	2 Cfsu
7.	Delete	2Cfsu

e) ASSIGNING & SCHEDULING

TABLE 22
Task Assignment And Schedule

Story No	Team	Task Assigned	Cfsu	Man days	No.of Days Required	Start Date	End Date
1	A	1,2,3,4	4+1+3=3=11	20	11	2.10.13	17.10.13
	B	5,6,7	3+3+3=9	18	12	2.10.13	18.10.13
TOTAL			17	34	12	2.10.13	18.10.13

Story No	Team	Task Assigned	Cfsu	Man days	No.of Days Required	Start Date	End Date
2	A	1,2,4,6	7+2+1+2=12	24	16	19.10.13	8.11.13
	B	3,5,7,8	2+7+2+2=13	26	17	19.10.13	8.11.13
TOTAL			25	50	17	19.10.13	8.11.13

Story No	Team	Task Assigned	Cfsu	Man days	No.of Days Required	Start Date	End Date
3	A	1,2,3,6	2+1+2+2=7	14	9	9.11.13	20.11.13
	B	4,5,7	2+2+2=6	12	8	9.11.13	20.11.13
TOTAL			13	28	17	9.11.13	20.11.13

6 CONCLUSION AND FUTURE WORK

In the present study an attempt has been made to address the risk involved in software development with the help of the proposed methodology which is developed with the features of both XP and 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 TC 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 TC 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 TC 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 for developing software.
A new methodology called XSoft was developed by integrating the features of both XP and COSMIC-FFP concepts to lay more for customer satisfaction. Also a tool called “XSoft Tool” was developed for determing the software size, estimation, scheduling, maintaining the details of XSoft project such as stories, tasks, programmers details, refactoring, scheduling etc. The case study is really useful to understand and develop the software using XSoft.

From our experience it was concluded that XSoft is appropriate and most noteworthy points are as below :

- More accurate measurement.
- Clear and best estimation.
- Making Stories and Pair programming gives more code, better code, better understanding of the system and also improves skills of the programmers.
- Customer satisfaction release and schedule provides early benefit and feedback to the programmer.
- Using the previous information about the skill of the developers (i.e.. the man days to complete one Cfsu in various Layers / Software) the company can make cost estimation and time estimation accurately for their future projects.

ACKNOWLEDGEMENT

The authors would like to thank Dr.S.Kanmani, Professor, Department of Information and Technology, Dr.N.Sreenath, Professor, Department of Computer Science and Engineering, Pondicherry Engineering College for their constant guidance. In recent days several researchers have focused to incorporate extreme programming in the teaching environment. The authors planned to concentrate to design an innovative teaching methodology to integrate XSoft in pedagogy. and develop a tool to measure & maintain the students programming skill.

REFERENCES

[1] ISO/IEC. 2003 19761: Software Engineering COSMIC FFP: A Functional Size Measurement Method, 2003.

[2] Jean-Marc Desharnais, Computer Engineening 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] Extreme Programming on PWAP (ficticious) a wireless application, Paul Rodrigues, E.Karunakaran, Naturesoft Esolution Provider, Chennai, India. Agile International Conference, Itally, 2001, pp. 73.

[5] 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.

[6] Extreme Programming Explained : Embrace Change by Kent Beck, Marti Fowler, Addison-Wesley, 2000.

[7] Planning Extreme Programming : Kent Beck et al; Addison-Wesley, 2000.

[8] Extreme Programming Installed : Ron Jeffries, et al; Addison-Wesley, 2001.

[9] Joseph C.Giarratano, Gary D.Riley, Expert Systems Principles and Programming. Beijing: China Machine Press, 2006.

[10] Z. Li, L.X.Lin, Y.B. Cao, Method and Practice of Function Point Analysis. Beijing: Tsinghua University Press, 2005.

[11] 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.

[12] 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.

[13] 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.

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

[15] 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.

[16] www.extremeprogrammig.org

[17] www.cosmicoa.com