Strategies Lead to Efficient Cost Estimation for Software Product

B. Sheeba

Abstract— **Software development is a much complex and multifaceted work. Estimation for software product is very much complicated task to accomplish. Software development is not only writing codes but also keep the cost and time. Cost and time will keep on changing in phase by phase. In software development we have five phases' analysis, design, implementation, testing and maintenance. The estimation is classified into three preliminary cost estimation, improved cost estimation and final cost estimation. Each estimate is enhancement of previous estimation and added additional information, error correction and work activities. The comparison of cost for development is much lesser than the maintenance. The maintenance phase will acquire to a great extent of time and a cost. When the changes keep on going in the maintenance phase, the cost will persist in increasing. So we cannot predict the precise cost for software development and maintenance, unless the programmer follow the software cost estimation strategies.**

Index Terms— **Estimation, Cost Factors, Cost Estimation models, Strategies for cost estimation.**

I. INTRODUCTION

The main objective of software companies not only is developing software, but also with effective cost estimation, which will lead them to a high target in the modern trendy world. How to prepare effective cost estimation for software product? This will arise in the analysis phase. The beginners will tediously suffer to prepare cost estimation for a project. Because for effective cost estimation, we need experienced person's judgments.

If the project is small cost will be reduced according to its size, if the project is big and very big the cost will get increase. Cost estimation for the bigger size projects will lead us to a headache. If we have good experts in our hands we can solve this problem, with their good experience. There are some of the models available to find the cost of a software product like COCOMO model, Expert Judgment, Delphi cost estimation, work breakdown structure etc.

While estimating the cost we need to consider the cost factors to empower the estimation. Considering the cost factors while estimating, it may lead to accurate estimation.

II. SIGNIFICANCE OF COST ESTIMATION

The foundation of the project is project manager and team members should estimate the work effort, resource and peripheral devices need, finally time needed from start to end. The milestones for engineering task will indentify the responsible person for each task which will be decided by the team manager. The manager is accountable for allocating the farm duties according to the personal skill.

How does the project manager estimate the cost of a project? This is incredibly wearisome job. Project manager can track any of the cost models, converse with software engineers or collect data from the foregoing project. For best estimation the project should undergo fragmentation, each and every fragment estimated separately. Final evaluation will be abstract of all the fragments.

Estimation resource falls under three categories they are development environment (software and hardware tools), people and software components. For development environment there exits software tools, hardware and network resources. While considering people falls under three category number of person needed, person skill and location. The reusable software components classified into four as below OTS (off the shelf) components, new components, full-experience components and partial experience components.

Off the shelf (OTS) components – has been developed for past project, these components are purchased from them and ready to use for current project.

Full experience components – the specifications, coding, design, algorithm, structure designing are similar to the past project. So the members of current software team have full experience, the risk is low.

Partial-experience components – developed previous project partially, similar to current project but it need enormous modification, so the team members have less experience.

New components – the current project's components must be built by the software team members.

III. REVIEW OF LITERATURE

According to Pressman, a real need for software has been established; stakeholders and on-board; software engineers are ready to start; and the project is about to begin. But how do you proceed? Software project planning encompasses five major activities—estimation, scheduling, risk analysis, quality management planning, and change management planning [1].

Estimation begins with a description of the scope of the product. The problem is then decomposed into a set of smaller problems, and each of these is estimated using historical data and experience as guides. Problem complexity and risk are considered before a final estimate made [1].

From the view of Pressman that's hard because you won't really know until the project has been completed. However, if you have experience and follow a systematic approach, generate estimates using solid historical data, create

Manuscript received May 23, 2016

B. Sheeba, Assistant Professor, Department of Computer Science and Applications, Nazareth College of Arts and Science, Chennai, India

estimation data points using at least two different methods, establish a realistic schedule, and continually adapt it as the project moves forward, you can feel confident that you've given it your best shot [1].

Richard Fairly states that there are many factors influence the cost of a software product. Primary among the cost factors are the individual abilities of project personnel and their familiarity with the application area; the complexity of the product; size of the product the availability, familiarity, and reliability; the level of technology utilized; and the availability, familiarity, and stability of the system used to develop the product [3].

There is a lot of software cost estimation methods or techniques in the software industry. Here are a few techniques [8]:

- Algorithmic (Parametric) model
- Expert Judgment (Expertise Based)
- Top - Down
- Bottom - Up
- Estimation by Analogy
- Price to Win Estimation

Examples of the parametric models are COCOMO (COnstructive COst MOdel), COCOMO II [10]. COCOMO stands for Constructive Cost Model; it is a software cost estimation model that was first published in 1981 by Barry Bohem. It is an algorithmic approach to estimating the cost of a software project. By using COCOMO you can calculate the amount of effort and the time schedule for projects. From these calculations you can then find out how much staffing is required to complete a project on time. COCOMO's main metric used for calculating these values is lines of code (denoted KLOC for COCOMO II, or KDSI for COCOMO) [10].

The main objectives of COCOMO II were set out when it was first published [9]. They are:

- To develop a software cost and schedule estimation model tuned to the life cycle practices of the 1990's and 2000's [9].
- To develop software cost database and tool support capabilities for continuous model improvement ([9].
- To provide a quantitative analytic framework, and set of tools and techniques for evaluating the effects of software technology improvements on software life cycle costs and schedules [9].

As Samuel Lee, Lance Titchkosky, Seth Bowen, software development is notorious for going over time and budget. This problem is due to the fact that software development is a complex process because of the number of factors involved, including the human factor, and the complexity of the product that is developed. Furthermore, the industry is highly competitive [8].

Although expert-based estimation is one of the most common methods of estimation because of its lightweight nature, the method suffers from being highly dependent upon competent estimators. Any model should be calibrated to the development environment because all development environments are different. In the end, an accurate estimate cannot be guaranteed and so using more than one method of estimation is recommended for verification of an estimate [8].

IV. CLASSIFICATION OF SOFTWARE COST FACTORS

Cost estimation is on the whole extremely significant. In the case of over estimate, the bidding software company may lose the project, because the opposite party has estimated more precisely and quoted low. In the case of under estimate, the company will make lose and also delay the project; this may cause penalties for the software organization.

[A] Development environment's availability and stability – to start the project hardware and software resources required. If the resources not accessible on time or not functioning as it should be then the programmers should wait for it or will do the development in shifts. So they cannot able to communicate with one another, they cannot conduct meeting. As stability if the compiler or operating system is trial version, the development team will devastate their time in chasing bugs instead of developing the project.

[B] Team Experience and Programmer skill - team experience and programmer skill plays significant role in this. Experienced team members will have knowledge in many projects so in the term of cost estimation they can envisage the cost correctly or approximately. Lack in programmer skill will lead to more elapsed time and it will lead to increased in cost.

[C] Team constancy – due to programmer's resignations or shifting to other projects will lead to instability in the team, will drag the existing project. The new team members will take a large amount of time to understand the project so increased in time, increased in cost. So team constancy plays important role in cost estimation.

[D] Maturity of organization – higher maturity of organization will widen the project with high productive.

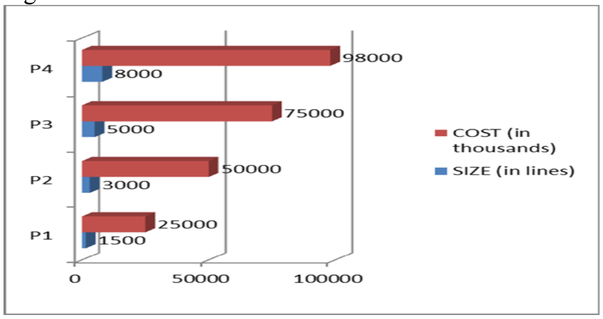
[E] Complexity of the product – there are three kinds of software products: application programs (data processing, scientific programmers), utility programs (inventory systems, linkage editors, compilers), system level programs (real-time systems, operating systems, data base management systems). The level of complexity is 1-3-9. The utility programs three time complexity than application programs. The system level programs three times more complexity than utility programmers.

[F] Project Size - according to the project size the work time will be elapsed. If the project is tiny then elapsed work time will be a lesser amount of time. If the project size is bigger elapsed work time will be more. Larger projects will be more expensive than the smaller projects. In larger project more number of lines will be there.

Table 1:

PROJECT	SIZE (in lines)	COST (in thousands)
P1	1500	25000
P2	3000	50000
P3	5000	75000
P4	8000	98000

Figure 1:



According to figure 1 P1, P2 represents Project Numbers, when the size of the project increases cost also increases.

[G] Height of Technology - the resources (hardware and software) used to develop the software will reflect the quality of the software. For example tradition of high level languages is better than assembly languages. Height of technology will increase the reliability and stability in the project. The high level technology tools will increase the productivity.

V. SOFTWARE COST ESTIMATION MODELS

To estimate the cost of a project there are two ways available top-down approach or bottom-up approach. Mutually have advantage and disadvantage. Top-down approach first starts from the resource requirements and staffing level estimation etc. Bottom-up approach first starts analyze from module, and functional points. There are numerous cost estimation models available. To find precise cost estimation we necessitate calculating using two or three models and compare the result and take the average from the results. The cost estimation models are

Expert Judgment model
Algorithmic cost model (COCOMO,COCOMO II)
Delphi cost model
Work breakdown structure

[A] EXPERT JUDGMENT MODEL – one of the top-down approach is expert judgment model widely used one. Expert judgment relies on the historical data of previous developed project. Current project somewhat similar to previous project so we can reuse the coding, tools, documents etc. The experts will estimate the cost of the current project from the similar previous developed projects. And for margin for safety the cost and time will be increased little bit.

[B] ALGORITHMIC COST MODEL (COCOMO, COCOMO II) – Constructive Cost Model is COCOMO model introduced by Barry Bohem in 1981. This model used to calculate work and time. COCOMO is trendy observed model used in collaboration of numerous tools. This model has three types of models

Basic model – in the early development stage it will give rough estimation

Intermediate model – this model will be used for more comprehensive requirements

Advanced model - in the final stage of the project, this model can be applied to refine the estimation

COCOMO II was introduced in 1997. This is an updated model of COCOMO. In COCOMO II the variable names and type of cost drivers has been changed as per modern trend. Code metric for a line is LOC – Line of Code. The decision control statements (if\if-else) statements will be counted as one line. Automated tools are available to count the lines in the program. There are three models available in COCOMO II

Application Composition Model – for rapid applications this model will applicable. Code metric for a line is objects points.

Early design model – this model developed in the early of analysis phase where requirements are made. Size estimate is in function points.

Post Architecture model – after the final architecture has been designed this architectural model will be made. Size estimates functions points or LOC is used.

[C] DELPHI COST ESTIMATION - Standard Delphi Technique – In this model the coordinator will provide project definition and a cost estimation form to a number of the cost estimators. The estimators will sit around they are not allowed to thrash out; they will quote the estimation in the form. The coordinator will examine the estimations. Any estimation differs sharply then this process will be iterated. There is another Delphi cost estimation offered that is the estimators are allowed to discuss about the project and their estimations.

[D] WORK BREAKDOWN STRUCTURE – this model belongs to bottom-up approach. The whole project will be divided into modules; either it can be process or product. The advantage of this model is every requirement will be met. Every module will be estimated by the experts and summation of all the estimation will give the cost estimation of the project.

VI. STRATEGIES FOR EFFICIENT COST JUDGMENT

- Create clear requirement
- Analyze nuke and corner of the module
- employ experience individual for requirement analyzing
- update and maintain current cost rate for the historical data
- Do the requirements for customer not for your sake
- Have report for everything
- Have conclusion from experts for everything
- Do calculate more models and compare
- Trial demo and original data
- Do test systematically to avoid major error
- Have team communication
- Do have more meeting
- Do have communication with customers
- Reuse the tools
- Have stabilized team
- Reduce the alteration
- employ knowledge people for maintenance
- Etc

CONCLUSION

Cost estimation is a complicated job in software engineering. Estimation begins at analyzing phase. A good expert analyzer will escort the company into better-quality cost estimation. The cost of resources increased day by day. So updating historical data will lead to precise cost estimation. Work breakdown structure model is one of the best models where we can meet all the requirements of the project. Reusing the tools for the current project will reduce the cost of the project. Team communication is very important cost factor. If the programmer did not communicate with the client then he may lead the project into wrong direction. After some time later programmer will come to know the project will not satisfy skate holders, and then redo the job. So elapsed time, work effort everything is waste of cost. And finally stabilized team members will lead to good cost effective project, and the quality of the project also will be good.

REFERENCE

[1] Rogar S.Pressman – “Software Engineering” A Practitioner’s Approach 6th edition 2005- McGraw-Hill International Edition [674,

[2] Swapna Kishore, Rajesh Naik - “Software requirements and Estimation” 2006 Tata McGraw Hill Edition

[3] Richard Fairly – “Software engineering concepts” – 2010 The McGraw-Hill Edition [65,

[4] Dean Leffingwell , Don Widrig – “Managing Software Requirements” A use case approach second edition 2007 – Pearson Education

[5] Robert Lagerstr¨om, Liv Marcks von W¨urtemberg, Hannes Holm andOscar Luczak -” Identifying Factors Affecting Software Development Cost” Industrial Information and Control Systems, The Royal Institute of Technology, Stockholm, Sweden 2007

[6] Joe Stangarone- “ 7 tips for cost-effective web application development” – 2015 Article

[7] PAUL BARNES - HEAD OF PROJECTS @ TOPTAL – “Software Costs Estimation In Agile Project Management”

[8] Samuel Lee, Lance Titchkosky,Seth Bowen – “Software Cost Estimation” Department of Computer Science University of Calgary 2002

[9] Boehm, B., Clark, B., Horowitz, E., Madachy, R., Shelby, R., and Westland C. (1995). Cost Models for Future Software Life Cycle Processes: COCOMO 2.0, Annals of Software Engineering. http://sunset.usc.edu/research/COCOMOII/Docs/stc.pdf

[10] Boehm, B., (2001) COCOMO Website. http://sunset.usc.edu/research/COCOMOII/cocomo_main.html