ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering &Technology (IJARCET)
Volume 2, Issue 9, September 2013

A SURVEY ON METRICS, MODELS & TOOLS OF SOFTWARE COST ESTIMATION

Soumyabrata Mukherjee, Bishnubrata Bhattacharya, Suvajit Mandal

Abstract- **Software cost estimation is the process of predicting the cost in terms of efforts required to develop a software product. A number of factors contribute to overall cost estimation process of software but factors such as software size and its complexity affects substantially to the effort software development accurately. Many models and metrics that have been proposed over the last 30 years in the literature in order to reduce software development cost. This paper summarizes the different models and metrics. Also provides a overview of software cost estimation tools which is essential for estimating. As the volume and complexity of software application are continuously increasing for that cost estimation becomes very effort intensive task.**

Index terms- **Effort estimation, size & complexity estimation, metrics, models & tools.**

I. INTRODUCTION

Software requirements are the starting point for every new project and are a key contributor to enhancement projects, as well. Software requirements are also very ambiguous, often filled with bad assumptions and severe errors, and are unusually difficult to pin down in a clear and comprehensive way. From a software cost estimating standpoint, the most tricky part of estimating requirements is the fact that requirements are usually unstable and grow steadily during the software development cycle in the coding and even the testing phases. Software cost estimation is an important activity of software project management and primarily used to estimate development effort of software. This involves the knowledge of a number of key factors related to the project for which the estimate is being constructed. Cost estimating is sometimes termed "parametric estimating" because accuracy demands understanding the relationships among scores of discrete parameters that can affect the outcomes of software projects, both individually and in concert. Now a days, software becomes the most expensive component of computer system project. The grow of the cost of software development is due to the human effort , most cost estimate tools focus on this aspect and gives the estimate.

Soumyabrata Mukherjee, M.Tech research scholar school of computer Engineering, KIIT University, Bhubaneswar, India.
Bishnubrata Bhattacharya, M.Tech research scholar school of computer Engineering, KIIT University, Bhubaneswar, India.
Suvajit Mandal, M.Tech research scholar school of computer Engineering, KIIT University, Bhubaneswar, India.

A. Need for software cost estimation

Cost estimating, risk analysis, project scheduling, quality management planning and change management planning are the five major for software project planning. Our attempt to measure how much money, effort, resource and time will take to make a specific software based system or product. Estimating is important if you make a car without knowing much you were spend the tasks you required to do it and the time limit for the work to be completed. For that we required or develop an estimate before we start making the software [1]. After survey from various papers and review report we got the following statistics [2]:-

1) Near about 15% of the projects get cancelled before their completion.
2) Successful project rate is about 34% i.e. it gets completed on time and on budget.
3) 20-25% projects don't meet Return on Investment.
4) Directly or indirectly approx 20% get failed at the initial phase.

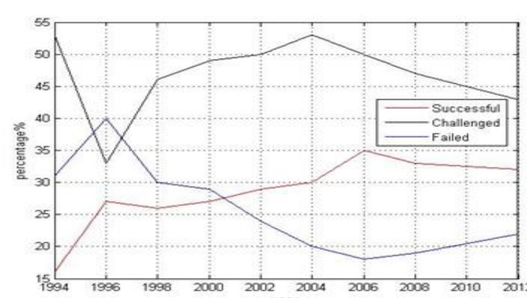

Fig:1- Graphical representation of failed and successful project

The numbers of challenged and failed projects are higher than successful ones due to improper cost estimation [3].

B. Software cost estimation steps

Software cost estimation process involves basic seven steps [4]:-

1) Demonstrating the specific goals.
2) Generating a sketch for required data and resources.
3) Gathering the software requirements.
4) Checking the feasibility of the software.
5) Using various cost estimation techniques to estimate the cost.
6) Balancing different estimates and restate the process.

7) Monitoring the estimated output and carry out the above steps.

C. Inputs and outputs to the estimation process

This is a set of techniques and procedures that is used to derive the software cost estimate by using set of inputs to the process and then it will generate the outputs based on the given inputs.

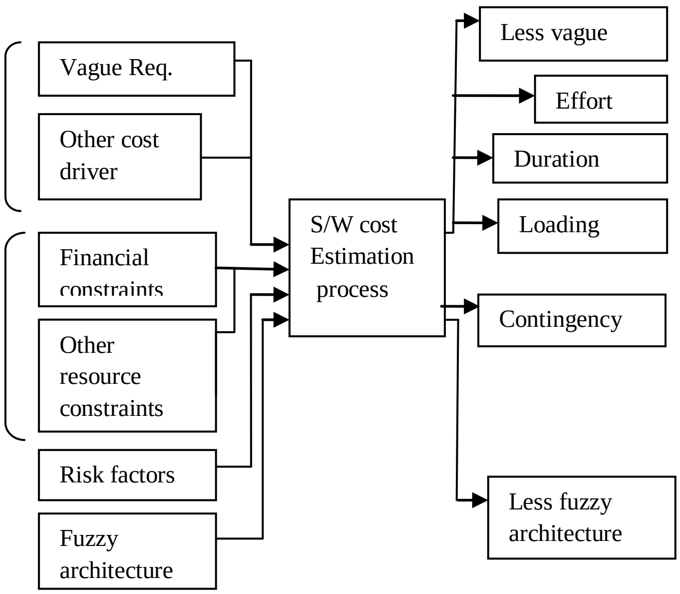

Fig:2- *Actual cost estimation process* [5]

The estimation of cost and effort depends on the actual prediction of the size. Generally in the software project the effort and cost estimation are difficult. Because software projects are mostly unique and there is no background or past experience about them, for that prediction is complicated. The main objectives of this paper to demonstrating the ability of the software cost estimation tools and clustering them based on their features that makes the readers to better understand.

II. BACKGROUND

Software project failure is an important matter in the last decade. Most of the project fails are related to the planning and estimation steps. Approximately within 30%-40% of the software project completed and rest are fails for going to over time and cost (Molokken and Jorgenson, 2003). The Standish groups CHAOS reports failure rate of 70% for the software project (Glass, 2006).Also it has been indicated 189% project fails for cost overran(Molokken and Jorgenson-Ostvold,2006). Galorath and Evans (2006) doing a brief search between 2100 internet site and found 5000 above reasons for the software project failures and most of the important reasons, to occur for inaccurate estimation, it is required regarding estimation the efficient technique and tools [6]. The scenario of the software development are not stable and the situation is continuously changing so several technique, tools should be presented for estimation that each technique appropriate for a special project [6].Many software estimation techniques and tools have been developed by researches like Boehm(COCOMO), Putnam & Mayer(SLIM), Jensen(SEER-SEM), Capers Jones(Checkpoint). In present scenario many estimation methodology exist. Many of them are a combination of the popular methodologies like expert judgment, algorithmic, top-down, bottom-up, estimate by analogy and price to win. Based on above methodology there are many tools developed.

III. TECHNIQUES OF SOFTWARE COST ESTIMATION

Calculation of software cost estimation can be done in various ways, but in this paper we will broadly subdivide into two major parts i.e. metrics and models.

A.Metrics

Metrics are nothing but standards & with these standards we will measure the whole software by calculating resource management, risk factors and other parameters which will give us the cost estimation. Now improper measuring i.e. improper cost estimation will lead the project to be failed, so these standards are highly important in our paper.

(1)Line of code

The codes written in lines for developing a program, excluding comment and blank lines are called Line of code. LOC [7] is language dependent i.e. for C-language it is 125 and for java it is 53.

For determining the LOC we use a technique called PERT [8]. It involves expert judgement of three possible code sizes: "S_l" the lowest possible size, "S_h" the highest possible size, "S_m" the most likely size. The estimate of the code-size S is computed as:

$$S = \frac{S_l + S_h + S_m}{6}$$

(2)Function points

Cost estimation can also be done based on function points [9] which are used in program & total no. of function points are equivalent to total no. of processing logic of that program. These

Function points can be subdivided into 5types

(a) User-input types
(b) User-output types
(c) Inquiry types
(d) Internal file types
(e) External file types

Each of these types has three different kind complexity levels i.e. 1=simple, 2=medium, 3=complex and weighting value varies from 3(for simple input) to 15(for complex internal file).

Now, the Unadjusted Function Point (UFC) is calculated by the following equation-

$$UFC = \sum_{i=1}^{5}\sum_{j=1}^{3} N_{ij}W_{ij}$$

Here N and W are respectively the number and weight of types of class i with complexity j. For example, if the raw function point counts of a project 3 simple inputs (W_{ij}=3), 2 complex input (W_{ij}=6) and 2 complex internal file (W_{ij}=15).

$$UFC = \big((3 \times 3) + (2 \times 6) + (2 \times 15)\big) = 51$$

B.Models

Models also play a vital role in case of software cost estimation. Here we will classify the models into two types based on the methods of their working principal i.e. algorithmic and non-algorithmic.

(1)Algorithmic

Algorithmic method depends on the mathematical equations which are based on theory or historical data. SLOC, number of function points etc are used as input. Any algorithmic model can be written as:
Effort= $f(x_1,x_2\ldots,x_n)$ where $\{x_1,x_2\ldots,x_n\}$denote the cost factors.

(a)Linear model

Boehm said that "there are too many nonlinear interactions in software development for a linear model to work well" [10].The effort is calculated by the following equation:

$$Effort = a_0 \prod_{i=1}^{n} a_i^{x_i}$$

Where the coefficients $a_1,\ldots, a_n$ are chosen to best fit the completed project data.

(b)COCOMO

The Constructive Cost Model i.e. COCOMO is an algorithmic method, developed by Boehm [11] for effort calculation which takes KLOC as input. COCOMO classifies the product into three categories i.e. organic, semidetached and embedded which uses different constant values.

$$Effort = a \times (size)^b$$

Where a & b are constants having different values such as 2.4, 1.05 for organic, 3.0, 1.12 for semi-detached and 3.6, 1.20 for embedded system.

(c)Delphi

Delphi is a review technique to determine software cost estimation. A group of 3-7 experts on some specific domain gives feedback based upon some specific scenario. The "Wideband Delphi" developed by Barry Boehm and John A. Farquhar and considered as better compared to Delphi. [12]

(d)SEER SEM

The System Evolution and Estimation of Resources Software Estimation Model (SEER SEM) is a parametric approach to estimate and was developed by Galorath Inc. It uses parameters as size, personnel, complexity environment and constraints for estimating project cost, risk and schedule. [11]

(e)SLIM

Software life-cycle model was developed by Putnam and uses Rayleigh curve to estimate project effort, schedule and defect rate. SLIM uses lines of code for estimating project size and then modifies this through the use of the Rayleigh curve model to produce its effort estimates.

$$Effort = \left\{\left(\frac{size}{productivity} \times time^{\frac{4}{3}}\right)^3 \times A\right\}$$

Where size is the product size expressed in Effective Source Lines of Code i.e. ESLOC, "A" is the special skills factor a function of the project size, Productivity is the Process Productivity. Time is the total schedule of the project in years and Effort is the total effort applied to the project in person years.

(2)Non- Algorithmic

Non-Algorithmic method requires one or more completed projects and derives estimation through reasoning by analogy using the actual costs of previous projects. Estimation by analogy can be done either at the total project level or at sub-system level. The total project level has the advantage that all cost components of the system will be considered while subsystem level has the advantage of providing more detailed assessment of the similarities and differences between the new project and completed one.

(a)Parkinson

Using Parkinson's principle "work expands to fill the available volume " [13], the cost is determined (not estimated) by the available resources rather than based on an objective assessment. Suppose if the software is to be delivered in 10 months and 4 persons are available, then effort is estimated to be 40 person-months. Although it sometimes gives good estimation but it is very unrealistic estimates. Also, this method does not promote good software engineering practice.

(b)Price-to-win

The software cost is estimated to win the project. The estimation is based on customer's budget instead of software functionalities. For example if 100 person-month is required to complete the project but customer can provide 80 person-month, then the estimator modifies the estimation to 80 person-month which is again not a good software engineering practice as it badly delay of delivery or force the development team to work overtime.

(c)Bottom up

In this approach, each component of the software is separately estimated i.e. system is decomposed into sub-systems, then results aggregated to produce an estimate for overall system.

(d)Top down

Top down model is also called Macro Model. An overall cost estimate for the system is derived from global properties. The total cost can be then split up among various components. This approach is more suitable for cost estimation at the early stage.

C. Comparative discussions of Models

Models	Advantages	Limitations
Parkinson	Helps to distribute project development time among different activities in a balanced manner without any additional overspend [10].	It may not promote good software engineering practice as it toggles between good estimation, overestimation as well as unrealistic estimation [14].
Price to Win	Often rewarded with the contract.	Time and money run out before the job is done.
Bottom up	More stable & detailed.	Sometimes overlook system level cost & it is time consuming.
Top down	Requires minimal project details & it is faster very easy to implement.	No detailed basis & overlook low level components.
COCOMO	It is simple to use and is widely acceptable by practitioners [15]. It has a better accuracy of prediction.	Again, it requires the knowledge of a number of cost drivers while ignoring the personal experience.
Delphi	It a simple process assumptions are documented, discussed and agreed on, and finally resulted with a more accurate estimation compared to individual estimation.	It depends on requires management cooperation, agreement among team members, and it depends on expertise of team members [15] [16]. This technique takes long time and considerable effort to complete.
SEER-SEM	Gives better accuracy in effort estimation. SEER-SEM with neuro-fuzzy model of effort estimation produces unique characteristics and an improved performance.	There are over 50 input parameters related to the various factors of a project, which increases complexity of SEER-SEM, especially for managing uncertainty from these outputs.
SLIM	The advantage of this model is that it supports most of the popular size estimating methods.	The estimates are sensitive to the technology factors [15].

D. Tools

Instructors should seek software tools for studying and computing software metrics. In addition, it is highly desirable that some computerized software metrics model be available for student experimentation. It may be possible to acquire software tools, spreadsheets and utilities that supports cost estimation. Tools Used for Estimating-

There are number of tools used for estimation but following tools are generally used in market:-

i)ACEIT (Automated Cost Estimating Integrated Tools)

ACEIT is a family of applications that support program managers and cost/financial analysts during all phases of a program's life-cycle. ACEIT applications are the premier tool for analyzing, developing, sharing, and reporting cost estimates, providing a framework to automate key analysis tasks and simplify/standardize the estimating process [17].

ii)Agile COCOMO II

Agile COCOMO II is a web-based software cost estimation tool that enables you to adjust your estimates by analogy through identifying the factors that will be changing and by how much [18].

iii)Center for Software Engineering (CSE) Tools Section

CSE access to the COCOMO Suite, USC COCOMO 81, USC COCOMO II, CodeCount and WinWin [19].

iv)COOLSoft

COOLSoft utilizes a hybrid approach of intermediate and detailed versions of the Constructive Cost Model (COCOMO). This allows for the reuse of existing code, development of new codes. It purchase and integration of third party code and hardware integration [20].

v)COSMOS

This state-of-the-art software project estimation and analysis tool gives project managers insight into the size, effort, and schedule of their software development project. This tool is unique in that it combines the well-known Function Point and COCOMO models as well as a Rayleigh model of staff buildup proposed by Lawrence Putnam. These three models can be used independently or work together [21].

vi)Costar and SystemStar

Costar is an automated implementation of COCOMO II developed by SoftStar Systems. SystemStar, an automated implementation of COSYSMO [22].

vii)PMPal

PMPal is a fully collaborative, full featured, integrated tool for software project management and software metrics programs [23].

viii)r2ESTIMATOR

r2ESTIMATOR is a Microsoft Windows application that estimates (with an associated probability of success) the cost, schedule, effort, staffing, and product reliability that can be expected from a given software development project [24].

ix)Taasc Estimator

Taasc Estimator is a comprehensive toolkit of advanced software components; each dealing with a specific aspect of planning, organizing and controlling the construction and development of software-intensive systems [25].

x)SEER

SEER is a powerful decision-support tools for estimating software development and maintenance cost, labor, staffing, schedule, reliability and risk as a function of size, technology and any project management constraints. It is a model base software cost estimation technique. SEER for software projects from commercial IT business applications to real time embedded aerospace system. It assist for making vital discussions about the development and maintenance and also ensure that the project plans are realistic, objective and developable [26].

Commercially available cost estimation tools try to offer the user greater utility by packaging the parametric model with a user interface, database of completed projects, some way of estimating the size of the project, and/or context-sensitive help. Whatever features any tool may have, most parametric models are likely to have the COCOMO equations at the core.

IV. CONCLUSION

Most of the effort estimation metrics takes the input as the software size, which can be measured with function point, LOC, object point, class point. But the complexity part of software has not properly addressed by most effort estimation metrics, whose presence may affect the overall estimation result. Generally in the software project, the effort and cost estimations are difficult. Because software project are mostly unique and there is no background or past experience about them, for that prediction is complicated.

References

[1] Roger S. Pressman, *Software Engineering A Practitioner's Approach*, 7th ed.: McGraw-Hill.

[2] Ph.D. Joseph Gulla, "Seven Reasons Why Information Technology Projects Fail," 2011.

[3] Christopher Brooks, "The Role of Project Management (PM) in Academic Information Technology (IT)," Center for Hybrid and Embedded Software, 2012.

[4] Centres for System and Software Engineering: http://sunset.usc.edu/classes/cs510_2012/EPs_2012/EP25_Chapter_21.pdf

[5] Dr.T.Boult,VAST Lab: http://www.vast.uccs.edu/~tboult/CS330/NOTES/Project%20Estimation.ppt

[6] Vahid Khatibi, Dayang N. A. Jawawi, "Software Cost Estimation Methods: A Review," *Journal of Emerging Trends in Computing and Information Sciences*, vol. 2, 2010-11.

[7] N.E.Fenton and S. L. Pfleeger, "Software Metrics: A Rigorous and Practical Approach," *PWS Publishing Company*, 1997.

[8] J. D. Aron, "Estimating Resource for Large Programming Systems," *NATO Science Committee*, October 1969.

[9] Mel Rosso-LIport, Carnegie Mellon University, School of Computer Science: http://www.cs.cmu.edu/~aldrich/courses/413/slides/4-estimation.pdf

[10] University of Pittsburg, Department of Computer Science ftp://cs.pitt.edu/chang/handbook/42b.pdf

[11] Kabita Hazra , and Rajib Mall Samaresh Mishra, "A Survey of Metrics for Software Development Effort Estimation," *International Journal of Research and Reviews in Computer Science*, vol. 2, October 2011.

[12] B. W. Boehm, "Software engineering economics," 1981.

[13] G.N. Parkinson, "Parkinson's Law and Other Studies in Administration," Boston, 1957.

[14] Zhang Fan Hareton Leung, "Software Cost Estimation," Department of Computing The Hong Kong Polytechnic University, 2002.

[15] Sunita Chulani Dr. Barry Boehm, "Software Development Cost Estimation Approaches," *Computer Science department at USC*, 1998.

[16] Amal K. Ali, "Using the Delphi technique to search for empirical measures," *The Qualitative Report*, vol. 10, pp. 718-744 , December 2005.

[17] Automated Cost Estimating Integrated Tools (ACEIT): http://www.aceit.com/

[18] USC Viterbi School of Engineering, Centre for Systems and Software Engineering: http://sunset.usc.edu/cse/pub/research/AgileCOCOMO/AgileCOCOMOII/Main.html

[19] USC Viterbi School of Engineering, Centre for Systems and Software Engineering: http://sunset.usc.edu/cse/pub/tools/

[20] Wright Williams & Kelly:http://www.wwk.com/coolsoft.html

[21] Department of Computing East Tennessee State University: http://csciwww.etsu.edu/cosmos/

[22] Costar Software Estimation Tool: http://www.SoftstarSystems.com/

[23] Consultants for Software Quality, Process Improvement, Estimation, Project Management and Trainers in Software Engineering: http://www.chemuturi.com/pmpal/index.html

[24] r2Estimating: Products: Next Generation Software Cost Estimating and Software Project Management: http://www.r2estimating.com/products.htm

[25] Project Estimation Software Circa Estimating Risk Management Gantt Charts Objectmetrix Resource Loading Bug Tracking Defect Logging: http://www.tassc-

solutions.com/software_pages/software.htm

[26] SEER – Cost Estimating Software - Project Planning, Project Tracking & Estimation Software for Complex Programs – Galorath: www.galorath.com

Soumyabrata Mukherjee received his B.Tech degree in the year 2012 from BIET, Suri affiliated to West Bengal University Of Technology(WB) and currently pursuing M.Tech in Computer Science Engineering with a Specialization in Software Engineering from KIIT University, Bhubaneswar. His research area includes Software Cost Estimation, Software Testing.

Bishnubrata Bhattacharya received his B.Tech degree in the year 2012 from BIET, Suri affiliated to West Bengal University Of Technology(WB) and currently pursuing M.Tech in Computer Science Engineering with a Specialization in Software Engineering from KIIT University, Bhubaneswar. His research area includes Automation Testing, S/W Cost Estimation..

Suvajit Mandal received his B.Tech degree in the year 2012 from BIET, Suri affiliated to West Bengal University Of Technology(WB) and currently pursuing M.Tech in Computer Science Engineering with a Specialization in Software Engineering from KIIT University, Bhubaneswar. His research area includes Software Cost Estimation, S/W Fault Localization Testing.