ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 5, May 2017

Usage and Analysis of Estimation Techniques in Software Project Management

Vishakha Sharma[1], Gaurav Garg[2]

M. Tech Student, Department of Computer Science Engineering , AITM, Palwal

MDU University, India[1]

Asst. Professor, Department of Computer Science Engineering , AITM, Palwal

MDU University, India[2]

ABSTRACT:. Software Cost Estimation is a process of predicting the efforts and cost in terms of money,schedule and staff for any software system. The purpose of our application which we have proposed to develop is to effectively gather software requirements from the customer and prioritize them. After the proposed software requirements are gathered from the customer, they are evaluated on the basis of quality attributes and sub-attributes like Project Type, Scope etc. in order to calculate the desirability values on the basis of which, they are prioritized. After prioritization, we take the user's input in relation to their expected prioritization and try to compare the prioritization of requirements from the developer and customer's perspective. The overall objective of the application is to minimize the disagreement counts that arise from the difference in prioritization of requirements from the developer's and customer's perspectives.

KEYWORD: Effort prediction, function point, estimation accuracy, estimation techniques, KLOC,these are all unit of software size, Software Efforts estimation, Person-month, Person-Hours these are units of e fforts.

I. INTRODUCTION

The procedure of calculating the plan, hard work, effort, size of the software solution, and overall cost associated with developing the software program application is referred to as software cost estimation. A cost estimation completed at the start of project can help decide which functions can be involved inside learning resource difficulties from the project. The risk of project is increases when the most important functions are involved at the end of the project. Thus, cost estimation may have a large effect on the life cycle and timetable for just a project. Just as we typically need to determine the weight, volume, and dynamic flight characteristics of a developmental aircraft as part of the planning process, you need to determine how much software to build. One of the main reasons software programs fail is our inability to accurately estimate software size. Because we almost always estimate size too low, we do not sufficiently fund or allow enough time for development. Poor size estimates are usually at the heart of cost and schedule overruns. We have chosen to develop this application/tool considering the objective of understanding the process involved in the estimation & usage of significant factors like Effort, Time & People involved in a certain software project. The organization is not clearly sure how to give these estimates as precise as possible because this classification or domain of the project is very new for them. We assume that they have not worked on such level of projects before due to which there are certain hidden factors or challenges about which the organization is unclear & uncertain at the moment. But they need to confirm the details within a stipulated time to the client failing which the company might loose the project also. Which is among the most complicated work within managing as well as preserving software program, during the improvement method cost as well as time period evaluation performs an essential part inside software cost evaluation method. Cost evaluation to get a usual software project will start through first scoping as well as planning cycle of the project.

- **ALGORITHMIC METHOD**
  Algorithmic models also called parametric models. These techniques start using a formula to calculate the cost & effort estimation. In this method costs are analysed using mathematical formulae inputs with metrics to produce an estimated output. This technique uses the mathematical equations to accomplish the application estimation.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 5, May 2017

- **NON-ALGORITHMIC METHOD**

In this method estimation process is done according to the analysis of previous datasets. Non-algorithmic methods usually do not use any formula to calculate the software cost estimation. This method makes comparison between previous dataset and existence dataset. After considering the categories of software cost estimation, we've proposed a novel concept of "Kernel Principle Component Analysis" (KPCA) which improves reliability and accuracy without applying the exhaustive procedures.

II. RELATED WORK

One of the main reasons software programs fail is our inability to accurately estimate software size. Because we almost always estimate size too low, we do not adequately fund or allow enough time for development. Poor size estimates are usually at the heart of cost and schedule overruns.Before developing this application, we considered below research papers:

According to Hareton Leung, Zhang Fan [15] today, almost no model can estimate the cost of software with a high degree of accuracy. This state of the practice is created because:

(1) There are a large number of interrelated factors that influence the software development process of a given development team and a large number of project attributes, such as number of user screens, volatility of system requirements and the use of reusable software components.

(2) The development environment is evolving continuously.

(3) The lack of measurement that truly reflects the complexity of a software system.

To produce a better estimate, we must improve our understanding of these project attributes and their causal relationships, model the impact of evolving environment and develop effective ways of measuring software complexity. At the initial stage of a project, there is high uncertainty about these project attributes. The estimate produced at this stage is inevitably inaccurate, as the accuracy depends highly on the amount of reliable information available to the estimator. As we learn more about the project during analysis and later design stages, the uncertainties are reduced and more accurate estimates can be made. Most models produce exact results without regard to this uncertainty. They need to be enhanced to produce a range of estimates and their probabilities. To improve the algorithmic models, there is a great need for the industry to collect project data on a wider scale. With new types of applications, new development paradigms and new development tools, cost estimators are facing great challenges in applying known estimation models in the new millennium. Historical data may prove to be irrelevant for the future projects. The search for reliable, accurate and low cost estimation methods must continue. Several areas are in need of immediate attention. For example, we need models for development based on formal methods, or iterative software process.

Whereas Lionel C. Briand, Khaled El Emam, Frank Bomarius, Fraunhofer[16] presented a method for Cost estimation, Benchmarking, and Risk Assessment: COBRA. This method has shown to be convenient and low cost when an organization needs to develop local cost and risk models and is not able to collect or retrieve a large set of project data. For example, the case study presented above took approximately 2 man-months of interviewing and analysis effort.

They have also shown how project manager expertise can be collected, refined, modelled, validated, and used for cost overhead estimation and cost risk benchmarking and assessment. They have also illustrated how the uncertainty associated with expert opinion can be modelled, integrated in the cost overhead model, and used through Monte Carlo simulation. Their resulting cost estimation and risk assessment models are operational and their construction is repeatable through a well-defined process. Their case study has shown good initial results on actual projects, thus demonstrating the feasibility of such an approach.

According to Bernhard Peischl, MihaiNica [17] this paper contributed a knowledge-based recommendation application to the domain of selecting appropriate effort estimation methods for software project management. The results of our survey on the pilot system confirmed our assumption that tailoring a recommendation application for project managers and engineers - in contrast to the typical tailoring for sales representatives - requires a very precise formalization of the underlying problem.

However, the answers to our questionnaire highlighted that such a recommendation application might be used in practice. This holds particularly for a complex knowledge base as a recommendation system might have genuine value

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 5, May 2017

for entering even more detailed facts about the project under consideration. In addition several technical issues and the need for further clarification of the capabilities of diverse methods (e.g. requesting the application domain directly vs. conveying that the application domain is an input to the method being proposed), traceability must be further improved if the system is to be widely adopted. In the near future they plan to extend the knowledge base to cover more project details. Besides of this, a user receiving a certain recommendation might be interested in obtaining the necessary information to extend into an accurate method.

III. SOFTWARE ESTIMATION TECHNIQUE

Software cost estimation can be approached and inspected in many different ways, there is collaborating between the concept technique, method and even model in the research papers mentioned in the literature, so it's necessary to make a distinction for all of these definitions. Many Organizations are used this methods, techniques and models in estimation process, Briand and Basilia and Thomas [11] classify the software cost estimation into six categories like model based, expert, learning oriented, dynamic based, regression based and composite Bayesian. Evans and Lanham and Marsh [12] propose 10 methods to estimate costs and consider each one a category according to the authors.
we recommend to make a classification of SCE where the techniques classified into qualitative and quantitative techniques, the qualitative techniques are further subdivided into intuitive and analogical techniques and mainly depend on similarity and comparison , and the quantitative ones into parametric and analytical techniques where its depend on the detailed design and can be used in advance phase of the implementation, we recommend select the techniques according to the availability of data and the environment, or it may be classified statistical or numerical techniques and the second is artificial intelligence techniques.

The current movements in utilizing the current available techniques is the ability to combine the techniques "amalgamated technique "together to get an optimum predicted values in the estimation process, Aggarawal et.al [13] use linear regression techniques and neural network to predict the LOC size from the function points, X.et.al [14] use neuro-fuzzy approach to build Constructive Cost Model (COCOMO).

IV. IMPLEMENTATION

Keeping the above scenario into consideration, we have planned to build an application that helps the organization to attain the below objectives:

1. Calculate the involved estimated effort in person-months
2. Calculate the required estimated time to develop the project
3. Calculate the estimated number of people required

We have planned to achieve the above objectives through a hybrid implementation of various software estimation & measurement techniques like Constructive Cost Modelling etc.

In addition, if we consider the above real time scenario, the organization would also want to utilize the above factors in some meaningful & fruitful way. Therefore, we have planned to utilize the estimation factors to attain the below secondary objectives:

Project Scheduling: Herein, we would evaluate how can the calculated estimated time or scheduled be used to create the milestones signifying various software development life cycle phases within the project.

People Planning: Herein, we will divide the calculated, estimated number of people required in various SDLC phases after proper analysis.

- Phase-wise Distribution of Effort
- Phase-wise Distribution of time in months

Apart from the above functions, if the organization wishes to view or crosscheck the data of the analysis that it performed for previous projects, it can do so through a detailed report.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

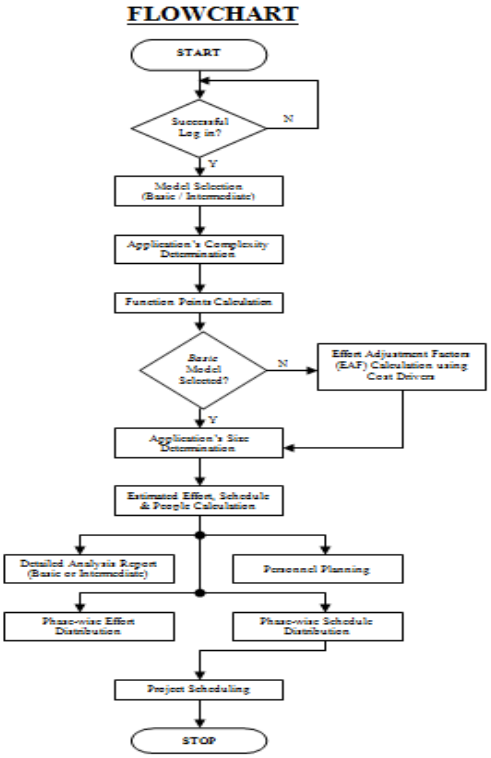
International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 5, May 2017

FLOWCHART



V. RESULT

In this project, we created different test cases as below in order to perform module-wise testing. The test cases were formulated keeping the overall objectives of the applications into consideration. In other words, we tested a certain module to ensure that it should perform its own function in addition to some other related functionality with other modules, if necessary. We recorded our testing results by giving different inputs to the modules and observing the actual result as against the expected one. Wherever, the test result failed, we incorporated the essential modifications to correct it.

S.NO.	MODULE	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	TEST RESULT
1)	Login	Incorrect Password	Error should be displayed to the user	Error is displaying	Ok
2)	Login	Correct Password	Estimation Model selection file should be displayed	Selection of estimation model is displayed	Ok
3)	Estimation Model Selection	Basic Model	Selection of cost drivers should be disabled	Selection of cost drivers is disabled	Ok
4)	Estimation Model Selection	Intermediate Model	Selection of cost drivers should be enabled	Selection of cost drivers is enabled	Ok
5)	Calculation	Size & Complexity	Effort, Time & People should be calculated	Effort, Time & People are getting calculated	Ok
6)	Cost Estimation	Non-numeric/ Blank Salary	User should be prompted to input	Message is displayed	Ok

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 5, May 2017

			numeric salary		
7)	Calculation	Basic/ Intermediate	After calculation of effort, time & people, the results should be correctly inputted in the database	Results are correctly getting stored in the database for the models individually	Ok
8)	Calculation	Basic/ Intermediate	After Effort, Time & People estimation, the phase-wise distribution of effort & time, project scheduling and people planning should be done automatically	Phase-wise distribution of effort & time, project scheduling and people planning is getting calculated	Ok

VI. CONCLUSION

This application is specifically targeted & meant for commercial software organizations that need to estimate, analyze& utilize different vital metrics involved within a software project like Complexity, Size, Effort, Schedule, Manpower, Cost etc. during the project discussion & planning phase. The importance of estimating the software metrics lies within the technique & methodology adopted for same. It is highly essential that the estimation tool should be user-friendly, self-explanatory and easy to use. The results that is generates should be extremely informative and helpful for a software organization not only in terms of project planning & management but also in giving sufficient convenience & confidence to the organization in extending near-precise commitments to their client. In other words, the tool should leverage the organization's capabilities and strengthens it in terms of maturity and increased goodwill & credibility.
All in all, this tool helps any commercial software organization to quantitatively estimate, analyse & utilize the fundamental software metrics that are a key to the success of any software project.

REFERENCES

[1] V.S. Alagar, L.Qiaoyum, and O.S. Ormandjieva, "Assessment of Maintainability in Object-Oriented Software," Proc. 39th International Conference and Exhibition on Technology of Object-Oriented Languages andSystems (TOOLS39), 29 July -03 Aug. 2010, vol. 19, pp. 194 – 204, Santa Barbara, Callifornia, USA, 2010.
[2] M. Dagpinar and J. Jahnke, "Predicting Maintainability with Object-Oriented Metrics – an Empirical Comparison," Proc10th Working Conference on Reverse Engineering (WCRE'03),13 - 17 Nov. 2011, pp. 155- 164, 2011.
[3] S.W.A. Rizvi and R.A. Khan, "A Critical Review on Software Maintainability Models," Proc. of the National Conference on Cutting Edge Computer and Electronics Technologies, 14 - 16 Feb. 2009, pp. 144 – 148,Pantnagar, India, 2009.
[4] P. Antonellis, D. Antoniou, Y. Kanellopoulos, C. Makris, E. Theodoridis, C. Tjortjis, and N. Tsirakis, "A Data Mining Methodology for Evaluating Maintainability According to ISO/IEC-9126 Software Engineering Product Quality .Standard," Proc. 11th IEEE Conference on Software Maintenance and Reengineering (CSMR2007), 21 – 23 Mar.2007, Amsterdam, Netherlands, 2007.
[5] P.W. Oman and J.R. Hagemeister, "Construction and Testing of Polynomials Predicting Software Maintainability," Journal of Systems and Software, vol. 24, no. 3, pp. 251- 266, 2013.[6] K.D. Welker and P.W. Oman, "Software Maintainability Metrics Models in Practice," Journal of Defense Software Engineering, vol. 8, no. 11, pp. 19 - 23, 1995.
[7] J.H. Hayes, S.C. Patel, and L. Zhao, "A Metrics-Based Software Maintenance Effort Model," Proc. 8th European Conference on Software Maintenance and Reengineering (CSMR'04), 24 – 26 Mar. 2004, pp. 254 – 258,IEEE Computer Society, 2004.
[8] M. Polo, M. Piattini, and F. Ruiz, "Using Code Metrics to Predict Maintenance of Legacy Programs: a Case Study," Proc. of International Conference on Software Maintenance, ICSM 2001, pp. 202-208, IEEE Computer Society, Florence Italy, 2001.
[9] J.H. Hayes and L Zhao, "Maintainability Prediction: a Regression Analysis of Measures of Evolving Systems," Proc. 21st IEEE International Conference on Software Maintenance, 26 - 29 Sept. 2005, pp. 601 -604, 2005.
[10] S. Muthanna, K. Kontogiannis, K. Ponnambalam, and B. Stacey, "A Maintainability Model for Industrial Software Systems Using Design Level Metrics," Proc. 7th Working Conference on Reverse Engineering(WCRE'00), 23 - 25 Nov., 2000, pp. 248 – 256, Brisbane, Australia, 2000.
[11] C. Briand, R. Basili, M. Thomas, "pattern recognition approach for software engineering data analysis," IEEE Transactions on, vol. 18, pp. 931-942, 1992
[12] D.K. Evans, Dr J.D. Lanham1, Dr R. Marsh," Cost estimation method selection: matching user requirements and knowledge availability to methods", Systems Engineering and Estimation for Decision Support (SEEDS) Group, University of West of England, Bristol, UK.
[13]K. K. Aggarwal, Y. Singh, P. Chandra, M. Puri. "An Expert Committee Model to Estimate Lines of Code"ACM ,vol 30,pp 1-4 ,2005

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 5, May 2017

[14] X. Huanga, D. Hob, J. Rena and Luiz F. Capretza," Improving the COCOMO model using a neuro-fuzzy approach," Journal of Systems and Software, vol.7, pp. Pages 29-40, 2005.
[15]Software Cost Estimation by Hareton Leung, Zhang Fan, Department of Computing, The Hong Kong Polytechnic University
[16]COBRA: A Hybrid Method for Software Cost Estimation, Benchmarking, and Risk Assessment International Software Engineering Research Network Technical Report ISERN-97-24 (Revision 2) by Lionel C. Briand, Khaled El Emam, Frank Bomarius, Fraunhofer Institute for Experimental Software Engineering (IESE) Germany
[17]: Recommending effort estimation methods for software project management Bernhard Peischl, MihaiNica Institute for Software Technology TechnischeUniversit¨at Graz 8010 Graz, Austria