International Journal of Scientific Research in Science, Engineering and Technology
Print ISSN: 2395-1990 | Online ISSN : 2394-4099 (www.ijsrset.com)
doi : https://doi.org/10.32628/IJSRSET2181122

Study and Analysis of Software Effort Estimation Methods using Watersluice Technique

Lakshmi Shanker Singh[1], Dr. Rupak Sharma[2]

[1]Research Scholar, Department of Computer Science, Monad University, N.H. 9, Delhi Hapur Road, Hapur, Uttar Pradesh, India

[2]Assistant Professor, S.R.M. Institute of Science, S.R.M. University, NCR Campus, Delhi Meerut Road, Modinagar Ghaziabad, Uttar Pradesh, India

Article Info
Volume 8 Issue 1
Page Number: 306-314

Publication Issue :
January-February-2021

Article History
Accepted : 01 Jan 2021
Published : 20 Jan 2021

ABSTRACT

There are many techniques we have to find out the cost estimation of the software project, when the overall labor is figured on a variety of factors. In This thesis has different research papers in it. Some used different techniques to get an estimate of the s/w cost but the estimated hours for the salary of a programmer isn't always very close to the actual s/w cost. To solve this issue, this author developed an artificial neural network to estimate the s/w cost.

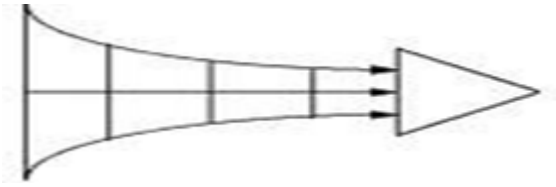
In this report, we have found that the Multilayer Feed-forward Neural Network technique is a good way to run a cost-analysis program. In our proposed work there will be 23 parameters that we control including 15 costs from the Ext. supplier, 2 biases, lines of code (Line of Code), actual cost and 5 scale factors. The calculation of our proposed work, MRE, and MMRE is the outcome we have planned for. The COCOMO data set has been used to train and test this framework or neural network. Neural networks trained on data from previous trials are compared with the conclusions that COCOMO model would have reached. The determination of our research is to increase the accuracy of the estimation of the COCOMO model by introducing it to the Multi layer neural network [6].

Keywords: COCOMO model, Watersluice Technique, Multi layer neural network, artificial neural network

I. INTRODUCTION

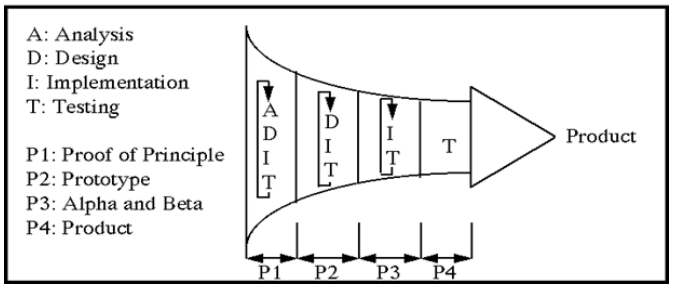
WaterSluice technique: WaterSluice technique is a composite technique which is a sequential and iterative Techniques. In this Fig.1 represent the many things which is covered in the WaterSluice representation model. WaterSluice approach having the sequential and iterative techniques. In this era there was a new technology where if we used all these techniques and methodology then we will solve many problems in a minimum time and execution

will be better rator than we used traditional methodologies.



In the s/w required estimation of development and effort is extremely important to arrange and managing of the s/w project management. The most significant thing in the s/w industry is the cost factor. For the preparation and control of project capital, effort estimation may be used. The critical problem in the s/w industry is the correct prediction of the cost of developing s/w. Likewise, for system developers and managers, system analysts and s/w developers, the precise determination of how much effort and time a program needed and function was also significant. In the last two decades, massive growth in the volume of data has been stored in databases. Also, and significant upsurge in the use of databases for commercial and technical purposes has been noticed.

The WaterSluice Methodology



In the Watersluice model success fullness of the interactive model for data storage as well as the evolution and growing of info extraction and handling techniques are the major reasons behind the explosion in the volume of data which is being stored in electronic manner. In the recent times, advanced techniques are being developed to meet the demand requirements. Also, some efforts have been made for the development of software for data analysis. The Watersluice The companies on the other hand have realized the important of valuable data hidden within these massive amounts of data that was being considered as insignificant till now [1]. The masses of stored data comprise information of a number of factors of different organizations in the offing to be extracted and adopted for supporting the business decision-making procedure with higher competence. DMS employed the management of these databases in recent time that just permits the client for getting database information in explicit manner. In databases, the stored data represents merely a minor portion of the "mountain of information" existing in it.

Without a fairly accurate cost estimating capability, it will be difficult to Line of Codeate the developers and managers of the s/w to decide the time needed for a project and the individual. During the system design process, system analysts cannot conduct practical hardware-s/w trade-off analyses; the s/w project workers may therefore not tell managers and customers that their planned budget and schedule are impracticable. This may benefit the production of s/w as a result of hopeful over promising and ineluctable overruns and performance accommodations.

1.1.1 Advantage of WaterSluice :

WaterSluice takes the advantages of both methods and technological respective of iterative as well as linear. In that case When selective the clear attention, it is said that WaterSluice has many benefits. The most important and critical benefit is the simplicity of the WaterSluice. WaterSluice is very basic techniques and very easy to follow. It gives the oversight of the project to the manager and departments. Testing is performed on a modular basis, so there is a reliability involved. When the project is reliable, then your customers are pleased and satisfied with your work. If customers are happy, then it will count for the overall performance of your company and staff.

1.1.2 Disadvantages of WaterSluice :

The WaterSluice model is a very new conceptual and HoLine of Codeene epoch model and managers are very hesitant to pursue this approach. If the software project is quite easy then, it is wastage of time to follow the WaterSluice technique. But if the project is complex and fresh and needs to be changed over the span of time, then WaterSluice approach would be the very good approach and safest technological way. Thus, wastage of time is one of the drawbacks of this technique. It also makes the basic stuff complex, so fewer managers go for this strategy. It is time consuming. The project's overview or the final product's form is noticeable by the end of the project. If you plan to go back and repeat all the steps, in case of any problem, then you will lag behind the schedule.

1.2 SCOPE AND IMPORTANCE

The research contains two methods of estimation, namely the feature point method and the case point method of use for various public administration, finance, wholesale/retail, and academic market application ventures. For ten different applications, it requires the study of different attributes/matrix of machine estimation. The homogeneity of the chosen application enables one to draw more generalized conclusions for the business application sector projects. Software is said to be an intangible commodity. Software development is a kind of all new stream in world business and there's very little expertise in developing software products. Most tech products are tailor designed to suit client's requirements. The most critical is that the underlying technology shifts and progresses so often and quickly that knowledge with one product can not be extended to the other one. All such market and environmental constraints bring risk in software development hence it is important to manage software projects efficiently.

The picture above illustrates triple constraints for software projects. It is an integral part of software company to deliver quality product, keeping the cost within client's budget constrain and deliver the project as per planned. There are many variables, both internal and external, which can affect this triple constrain triangle. Any of three factor will severely impact the other two. Therefore, software project management is important to integrate user expectations along with budget and time constraints. A software project manager is a person who undertakes the task of implementing the software project. Software project manager is well aware of all the stages of SDLC that the software will go through. Project manager can never directly participate in manufacturing the finished product but he monitors and oversees the activities involved in production.

II. LITERATURE SURVEY

Kemerer et al. fifteen groups of projects were sampled from business and marketing applications, and compared using four systems: SLIM (Slim size), COCOMO (COCOMo), Estimacs (Estimacs), and Function Points (Functionality Point)." He reported an estimation error in terms of the Mean Magnitude of Relative Error (MMRE) ranging from 85 percent to 772 percent .

In order to fully understand how many people may be harmed by mobile devices, Barry Boehm decided to build on that model even further, using mobile devices. Just after COCOMO was introduced in 1993, a new version of COCOMO emerged called COCOMO Two Thousand years. (COCOMO 2.0) In 1994, Rajiv D Banker and Hsihui Chang and Chris F Kemmerer suggested that s/w development is similar to a factory production line, and that s/w projects must be evaluated and managed in a similar way to expensive manufacturing projects. This number, obtained from the original information, estimated that Robotic surgical solutions had to be able to size below 2 to 10. In 1999, the original models were accessed and attention was put on accuracy of

information. In the year of 1998, Chatzoglou created a latest model called MARCS.

Mukhopadyay et al. used the project data set of Kemmerer and discovered that the COCOMO model was outperformed by 100 maintenance projects have been used by Jorgensen [24] to test several regression variations, artificial neural networks, and OSR combinations with regression. He found that it worked best in terms of accuracy for two multiple regression models and a hybrid model combining OSR with regression. In general, in order to justify investments in those models, he recommended the use of more sophisticated prediction models like OSR together with expert estimates.

It was a methodical process in which the illustrations had found in huge data sets. Assuming the data from the data set was the main objective of this data mining technique. This data was changed into a logical structure so as to use it further. The diabetes data that was collected used to learn design focusing on this part of medical conclusion. The smart therapeutic choice was produced that was expressively supportive network useful for the doctors. The assembling of Intelligent Diabetes Disease Prediction System was the fundamental goal of this examination. The database of diabetes patient had employed in this system to provide the diabetes malady analysis. The exploitation of KNN and Beyesian algorithms was suggested in this system that had carried out in the datasets of diabetes patients. Several diabetes features were extracted for analysis of these algorithms for predicting the disease of diabetes.

Santosh Rani, et.al (2018) studied the data related to the heath that was generated in huge amount at several stages of health system [46]. This data was not processed easily and the analysis of this data was hard to extract because of its huge size. However, the data was processed using various approaches planed on the basis of ML. The efficient data was obtained from this approach and this data was useful for the treatment of patients. This approach is also useful for predicting the disease's future. The contribution of past history of patients for various parameters had helped in the possibility of different health issues. In the continuous data, the data mining based on Association clustering and Time Series had utilized to build up the early warning system. When the existing parameters were analyzed, the disease had described using system based on prediction. The patient had saved from the disease at some extent with some level of care.

Rukhsar Syed, et.al (2018) suggested an algorithm that received benefits from the tree-based partitioning [47]. Furthermore, this algorithm utilized the adaptive approach of SVM for the categorization. The preprocessing was carried out in sampling SMORT in this algorithm for pruning the data. The Weka tool had employed for the experiment purpose on the diabetic dataset. The comparison of the suggested algorithm had performed with RF based on tree, RT approach and J48 approach. The experimental outcomes demonstrated that the suggested algorithm achieved the more effectiveness than conventional solution to process the diabetic data and the proficient categorization was found from that data, In 2007, different methods for estimating efforts were introduced. The average accuracy of the effort estimates based on expert judgment it concentrated on predicting the precision of models. The neuro-fuzzy system was therefore used to generate the model[33] as a soft-computing approach.

Some theoretical problems comparing estimation models have been identified during 2009. The selection of one or two datasets to prove the validity of the new technique was invalid[9]. Ayman Alahmar, et.al (2018) proposed the stacked ensemble methodology that was carried out with DL and considered as the meta-learning algorithm [51]. The

short as well as long LOS had foreseen for the patients of diabetes. The capability of stacked ensemble methodology was proved in this field by its outcomes. The superior performance for prediction had achieved in the results of algorithms of stacking multiple classification learning as compared to other basic learning algorithmic approaches. The sensible estimation on LOS for the diabetes patients had attained that proved useful to diminish the cost of healthcare and enhanced the contentment of patient who suffered from diabetes.

Geetha Guttikonda, et.al (2019) suggested the relevant data mining methods that were utilized for predicting the diabetes [52]. The extraction of preferred knowledge from the stored information of dataset and the analysis of the patterns of data was the major motive of data mining. The people were suffering from many issues related to health and some were not even conscious about the symptoms of these health problems. The Diabetes Mellitus was one among such health diseases. The young ones were also faced this disease. The prediction of these diseases was done employing the HUE for predictive analysis. The behavior of these diseases was relentless. The Pima Indian database was utilized to assemble the dataset. The persons who had diabetes were calculated using effectual technique that was achieved from the suggested framework together with SVM classification.

III. USE CASE POINT ANALYSIS

We had the opportunity to generate estimates early in the project lifecycle for our use. However, like COCOMO and Functionality PointA, a large number of relevant historical metrics are largely dependent on the accuracy of estimates generated using the Use-Case Points estimating technique. The usage case diagram and use case descriptions are required for the Use-Case Points calculation process. This description contains the steps involved in the execution of a use case. These steps are referred to as 'transaction' of the use case. Bakshi Rohit Prasad, et.al (2014) suggested a systematic framework for constructing a model to predict possibility of T2DM in a person [57]. The diabetes risk level had predicted with the best set of indicators. These indicators were recognized using GBRE algorithm. The accuracy of numerous classifiers was evaluated after trained them. The voting policy scheme was utilized for selecting the best classifier. The prediction precision was enhanced by integrated the mining process and expert knowledge of medical domain. This recommended approach was carried out in predictive models of other diseases. The patterns that were applicable and the inter dependencies of disease attributes had recognised using this work. The analysis of function points was developed in 1975. The function point measures the software-provided functionality. It can be determined from the specification of requirements, specification of design and code of the program. It is independent of the software implementation technology and language that is used.

Rational Software Corporation developed a method of estimating the software project based on Use Case Points, in the same way that Functionality PointA assigns function points and includes statistically weighted modifiers. We like the ability to generate estimates early in the project lifecycle for our use as a way to respond to our customers' needs. Similar to COCOMO, however,

The accuracy of estimates generated using the Use-Case Points estimating technique and Functionality PointA is largely dependent on a large number of historical metrics that are relevant. The usage case diagram and use case descriptions are required for the Use-Case Points calculation process. This description contains the steps involved in the execution of a use case. These steps are referred to as 'transaction' of the use case.

1. Hypothesis

S/w estimation is an important element in s/w development, helping managers tender for programs, and effectively used all resources in software project management. There are so many approaches for estimating machine efforts. Among them, feature point analysis and case point analysis are two significant techniques. From the comparison of these techniques, we can describe the approximation of the techniques to the true.

Our analysis aims to address the following questions:

How many projects in s/w development deviate from the original plan in terms of expense, schedule and functionality?

Which methods are used to estimate machine effort, and if they vary in accuracy systematically?

How important is the accurate measurement of effort perceived to be, and how important is the degree of accuracy considered an issue in s/w industry?

What are the key reasons of deviating from their initial strategy for s/w projects?

OBJECTIVE

Comparison of method of feature point with the s/w effort estimation method to make the decision process simple by conducting case study on ten applications.

The storage of the huge amount of data is possible for the extraction of valuable knowledge. Over the time, researchers have developed numerous algorithms to explore useful patterns available in the data. These algorithms have the ability to classify the data in either automatic or semi-automatic manner. The set of rules can be obtained using these patterns. It is imperative for the explored patterns to be meaningful to serve different purposes. The meaningful patterns may be fruitful in many ways. Some major benefits include decisions making, bazar study,economic evolution, and so on. There is the need of enormous quantity of data for getting such worthwhile patterns [2]. he knowledge base utilized for guiding the search or computing the likeliness of the result patterns. The data obtained from user experiences are included in it which can be proved helpful in software development procedure.

Following are the major motivational ideas practised in the real world:

The following steps are required:

- Studying the assessment of the program effort
- Study the method of feature point and use case point method of estimation of effort
- Select 10 requests for our job.
- Calculate each application's attributes in both approaches,
- Comparison of these attributes between the two methods of all applications The comparisons require the following steps,
- Do before the application finishes
- Take one request
- Evaluate the Ext. input, Ext. output, Ext. inquiries, as well as the application's general device features
- Calculate the point count of the function
- Estimate the effort from the feature point count, Line of Code, Product Delivery Rate (PDR), efficiency and time taken for growth.
- Draw a case diagram for use
- Determine the difficulty and use cases of actors
- Identify the variables of technological and environmental complexity of the application
- Calculate the point count of the use case
- Estimate the effort, the Line of Code, the rate of product delivery (PDR), productivity and time taken from the use case point count for growth
- Compare attribute values for both methods

with statistical methods.

- Draw the graphical attribute plot
- Select next application for the next application
- For all the applications, compare the values

College affiliation system

Values of various attributes of College Affiliation system can be estimated as follows:

Line of Code (In JAVA)

Line of Code of College affiliation system can be calculated by using Eqn (5.1). Line of Code = Size * 46

1) In Use-Case Points:

Line of Code = Use-Case Points count * 46

= 61.05 * 46
= 2808

Product Delivery Rate (PDR)

PDR of College affiliation system can be calculated by using Eqn (5.2). PDR = Effort in hours / Count

2) In Functionality Point:

PDR = 1131.88/53.53
= 21.14

3) In Use-Case Points:

PDR = 1221/61.05
= 20.00

Productivity

Productivity of College affiliation system can be calculated by using Eqn (5.3). Productivity = Size / effort in hours

In Functionality Point:

Productivity = 53.53 / 1131.88
= 0.05

In Use-Case Points:

Productivity = 61.05 /1221
= 0.05

Effort from COCOMO

Effort through COCOMO model can be calculated from Eqn (5.4). Effort = 3.2 * KLine of Code $^{1.05}$

4) In Functionality Point:

Effort = 3.2 * 2.462 $^{1.05}$
= 8.24 person-months

Time

Time of development of College affiliation system can be calculated by using Eqn (5.5). Time = 2.5 * Effort $^{0.38}$

5) In Functionality Point:

Time = 2.5 * 8.24 $^{0.38}$
= 5.57 months

6) In Use-Case Points:

Time = 2.5 * 9.46 $^{0.38}$
= 5.87 months

These values of different attributes of the project College affiliation system estimated in function point method as well as use case point method can be summarized in the table 1

TABLE 1. ATTRIBUTE VALUES UNDER Functionality Point & Use-Case Points: COLLEGE AFFILIATION SYSTEM

Attributes	Functionality Point	Use-Case Points
Count	53.53	61.05
Effort(PM)	6.29	6.78
Effort(Person Hours)	1131.88	1221.00

Line of Code(In JAVA)	2462	2808
PDR(Hours effort by count)	21.14	20.00
Productivity	0.05	0.05
Time (months)	5.57	5.87

IV. CONCLUSION AND FUTURE SCOPE

The many aspects that influence the developing of s/w, including many other factors. There are many factors that can pose a risk or pose no risk. These can be human, technical, or political in nature. This reading should in no way be understood as a call to give up the use of estimating because even inaccurate estimates produce a better quality of life. Our researchers have examined several estimation techniques and have provided two approaches that can be used to measure the quantity during estimation process. If the estimation is done precisely, it will result in a decrease in error. In conclusion, we find that.

1. Assessment of effort in Use-Case Points is more as compared to Functionality Point
2. Assessment of Line of Code in Use-Case Points is more as compared to Functionality Point
3. Assessment of PDR (Product Delivery Rate) in Use-Case Points is less as compared to Functionality Point
4. Assessment of Productivity in Use-Case Points is more as compared to Functionality Point
5. Assessment of Time of Development in Use-Case Points is more as compared to Functionality Point

When we take into account the characteristic systems and situational levels, we have determined that a point on a use case is worth X function points and the affect that this has on the design is that we should use use cases to reduce effort try to code to the user's level of understanding.

Working on this project, we looked at several different factors – how different these factors are from each other, and how different these factors are from the ones we initially used ourselves. We found that the use of a more precise method of point computation has been more effective than the original point analysis method.

V. REFERENCES

[1]. Albrecht, A.J. Measuring application development productivity. In: SHARE/GUIDE: Proceedings of the IBM Applications Development Symposium, (October 1979) 83-92.

[2]. Boehm, B. Software Engineering Economics. Englewood CI@, NJ Prentice-Hall (198 1).

[3]. Putnam, L.H. A general empirical solution to the macro software sizing and estimation problem. IEEE Transactions on Sofiware Engineering, ~01.4, no. 4 (‘July 1978) 345-381.

[4]. Kemerer, C.F. An empirical validation of software cost estimation models. Communications of the ACM vol. 30, no. 5 (May 1987) 416-429.

[5]. Albrecht & Gafney. Software function, source lines of code, and development effort prediction: a software science validation, IEEE Transactions on Software Engineering 9, 6 (1983) 639-648.

[6]. Shepperd, M., Schofield, C. Estimating software project effort using analogies. IEEE Transactions on Software Engineering, vol. 23, no. 12 (November 1997) 736-743.

[7]. Mukhopadhyay, T., Vicinanza, S.S., Prietula, M.J. Examining the feasibility of a case-based reasoning model for software effort estimation. MIS Quarterly (June 1992) 155- 171.

[8]. Srinivasan, K., Fisher, D. Machine learning approaches to estimating software development effort. IEEE Transactions on Software Engineering, VOI. 21, no. 2 (February 1995) 126-137.

[9]. Briand, L.C., Basili, V.R., Thomas, W.M. A pattern recognition approach for software engineering data analysis. IEEE Transactions on Software Engineering, vol. 18, no. II (1992) 931-942

[10]. Jorgensen, M. Experience with the accuracy of Software Maintenance Task Effort Prediction Models. IEEE Transactions on Software Engineering, vol. 2 1, no.8 (August, 1995) 674-68 1.

[11]. B.A. Kitchenham and K. Kansala, "Inter-Item Correlations among Function Points," Proc. First Int'l Symp. Software Metrics, Baltimore, Md.: IEEE CS Press, 1993.

[12]. N. Karunanithi, D. Whitley, and Y.K. Malaiya, "Using Neural Networks in Reliability Prediction," IEEE Software, vol. 9, no. 4, pp. 53-59, 1992.

[13]. G.E. Wittig and G.R. Finnie, "Using Artificial Neural Networks and Function Points to Estimate 4GL Software Development effort," Australian J. Information Systems, vol. 1, no. 2, pp. 87-94, 1994.

[14]. F.Brooks, The Mythical Man-Month; Essays on Software Engineering, 1975. Addison-Wesley, Reading, Massachusetts.

[15]. Robert C. Tausworthe, 1981. Deep Space Network Estimation Model, Jet Propulsion Report.

[16]. Charles Symons 1991. Software Sizing and Estimation Mark II function Points (Function Point Analysis), Wiley 1991.

[17]. Allan J. Albrecht May 1984. AD/M Productivity Measurement and Estimation Validation, IBM Corporate Information Systems. IBM Corp.

[18]. Barry W. Boehm, Bradford dark, Ellis Horowitz, Chris Westland, Ray Madachy and Richard Selby. Cost Models for Future Software Lifecycle Processes: COCOMO 2.0 Annals of Software Engineering. Volume 1, pp, 57-94, 1995.

[19]. Banker, R. D., H. Chang, et al. (1994). "Evidence on economies of scale in software development." Information and Software Technology 36(5): 275-282.

[20]. Cockcroft, S. (1996). "Estimating CASE development size from outline specifications." Information and Software Technology 38(6): 391-399.

[21]. Chatzoglou, P. D. and L. A. Macaulay (1998). "A rule- based approach to developing software developmentprediction models." Automated Software Engineering 5(2): 211-243.

[22]. Dolado, J. J. (2000). "A validation of the component- based method for software size estimation." IEEE Transactions on Software Engineering 26(10): 1006- 1021

[23]. Ali Idri, Alain Abran, Taghi M. Khosgoftaar. 2001. Fuzzy Analogy- A New Approach for Software Cost Estimation. International Workshop on Software Measurement (IWSM'01).

[24]. Magne Jorgensen, A Review of Studies on Expert Estimation of Software Development Effort, March 2002.

[25]. Yunsik Ahn, Jungseok Suh, Seungryeol Kim and Hyunsoo Kim. July 2002. Journal of Software Maintenance and Evolution: Research and Practice.

[26]. Magne Jorgensen. May 2007 Forecasting of Software Development Work Effort: Evidence on Expert Judgment and Formal Model.

Cite this article as :

Lakshmi Shanker Singh, Dr. Rupak Sharma, "Study and Analysis of Software Effort Estimation Methods using Watersluice Technique", International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET), Online ISSN : 2394-4099, Print ISSN : 2395-1990, Volume 8 Issue 1, pp. 306-314, January-February 2021. Available at doi : https://doi.org/10.32628/IJSRSET2181122
Journal URL : https://ijsrset.com/IJSRSET2181122