Journal of Harbin Engineering University
ISSN: 1006-7043

Vol 44 No. 7
July 2023

Development of Software Projects: A Review of Various Cost Estimation Techniques

AJAY JAISWAL[*]
Prestige Institute of Engineering Management & Research,
Indore (MP) India
Email id: ajay.jaiswal55555@gmail.com

JAGDISH RAIKWAL[2]
Institute of Engineering and Technology,
Devi Ahilya University, Indore (MP), India
Email id: jraikwal@ietdavv.edu.in

CHANDRASHEKHAR CHAUHAN[3]
Institute of Engineering and Technology,
Devi Ahilya University, Indore (MP), India
Email id: cchauhan@ietdavv.edu.in

Abstract-In the standardized software development method, costs are predicted more accurately before the software is actually developed. Recently, software development cost estimation has received significant attention from analysts and has become a venture for the industry. The project applied the method for software development is currently growing increasingly varied. The most motivating factor for this range inquiry has been the inaccurate estimation that was revealed during software development. The goal of this investigation was to present the current state of the art in measuring the effort and to suggest a modern strategy. There is not a single approach per definition that can be regarded as the guiding approach in this essay. It will be suggested that a combination of the techniques be used to provide an accurate set to take measures. The most widely used accuracy measurements are the Magnitude of Relative Error (MRE), Mean Magnitude of Relative Error (MMRE) and Prediction Accuracy (PRED). The datasets that are utilized the most common include ISBSG, COCOMO, Albrecht, and Kemerer.

Keywords-Software Project Development, Estimation Methodology, Inaccurate Estimation and Cost Estimation.

I- Introduction

Some of the major issues for software companies because of the rapid technological change are the introduction of software applications at a reduced cost and the need to retain better quality software. If they are delivered on time and within the budget projected, the software will operate as planned. The main objective of the software is to produce high-quality software on time and on a budget. Many significant components of software have been developed throughout history in response to the growing in demand for software in both industry and societies. It is necessary to arrange and manage things in an acceptable approach as a result. Different techniques are being used in order to deal with concerns like improper performance, a delay in the program's deadline, and higher programme generation costs, etc. [1][2]

The work required to create a software programme is simply estimated when evaluating the software costs. Here [5][6], the accuracy of the relevant factor could be important. The software system scale is indeed a fuzzy range because there are many unpredictable factors in the evaluation of software system costs, including artificial language, development methodology, organization, etc. The procedure is further advanced because some necessary factors are known even at the start of the process. The acronyms used in this research are listed in table 1.

Table 1. List of Acronyms

Acronyms	**Explanation**
UCP	Use Case Point
ISBSG	International Software Benchmarking Standards Group
MRE	Magnitude of Relative Error
ML	Machine Learning
RF	Random Forest
COCOMO	Constructive Cost Estimation Model
FP	Function Point
IFPUG	International Function Point Users Group
KLOC	Kilo Lines of Code
SDECE	Software Development Effort and Cost Estimation
GA	Genetic Algorithms
UUCP	Unadjusted Use Case Points
ANN	Artificial Neural Network
FFNN	Feed-Forward Neural Network
LOC	Lines of Code
SA	Standardized Accuracy measure
LR	Literate Review

II – Literature Survey

An accurate prediction results right away in a successful software project that stays within the budget; conversely, any deviation in the total effort and cost estimate may lead to a project failure in terms of delivery time, budget, or

possible choices. The estimation of the cost of software development is regarded as a fundamental activity for managing the project's cost, time, and quality throughout its life cycle. Nowadays, the software development project activity stage is much more varied and extensive. The accuracy of several cost estimation techniques has also been brought up in recent research along with the technological discovery. Therefore, it appears that we agree with the results of the recently successful review. Table 2 shows a number of previous researches that take a quick look at cost projections in software development initiatives.

Table 2. Review of the literature for several articles

Study Paper	Research	Research Connectivity	Research Gap
[4]	The IFPUG presented Function Point as an appropriate option used in government and industry, in contrast to the Use Case Points, based on surveys that were done.	Compared to UCP we chose FP as the major factor.	IFPUG only matches Function Point to the original Use Case Point (not yet integrated) method.
[5]	Software cost estimate overview, Three most effective software cost estimation approaches.	COCOMO II is commonly used, compare with Model COCOMO.	The comparison of the estimation only Theoretical, not practically define Accuracy of model.
[6]	Measure the performance of an Academic Information System using Function Points Analysis (FPA).	Relative Complexity Adjustment Variables, evaluated by Basic Function Points, then estimate the Function Points.	Lacking a clear how Utilize the Use Case Diagram and Class Diagram to estimate points and weighting factors.
[8]	An international organization's systematic assessment of the literature in order to develop the FPA process and recommend adoption of FPA by a government body.	Based on making the FPA process more effective, suggested over the past 13 years.	Need to update FPA to accommodate possible enhancements. Investigation of the relation with the paradigm of design.
[10]	Comparative analysis of the methods for estimating the size of software growth.	The case study of research involving the licence applications of four distinct companies allows for a comparison of these two methods.	For commercial enterprise software, developers are not yet in a position to find out which approach is excellent independently.
[9]	Survey on some estimate of the software effort, try to get better estimate of the effort.	Summed up definition for improving the accuracy of the different methods.	Different cost estimation methods focus on only algorithm.

[11]	Use Case Point activity based count as an Integration Model for Software Cost Estimation.	For the company's limited resources and the difficulties with enterprise scale software, further testing is still necessary.	Advance testing is still necessary due to the company's limited resources and the difficulties with enterprise scale software.
[13]	The value and cost for fictitious software were proceeded to estimate using the proposed simulation using three methods: Waterfall, Scrum, and Kanban.	Evaluate the cost and value that confirm to the requirements of individual software.	By analysing the function, performing detailed cost estimation such as the function point method is conceivable.
[14]	Cost estimation of computation using FP with Non-algorithmic approach	An effective method for estimating function point-based analysis is produced by analysis of the Direct Method and Indirect Method initiative.	Only FP represent measured line of code with use of Direct and Indirect fuzzy logic methods.
[17]	A Comprehensive Analysis of the Literature Update on Effort Estimate in Agile Software Development.	The agile approach suggests that the application of team and project aspects is more common than the incorporation of more technical variables.	Finally, we found that advances have been achieved, despite the fact that precision is still an issue.
[18]	Software Development Cost and Effort Estimation Methods Over Five Decades: An Automated Text Mining Approach Study	Finding emerging research trends in technologies, methodologies, frameworks, tools, and techniques in a particular area or issue in any field of science or social science requires the proposal of a general automated text-mining framework.	It was discovered that the results were identical, with the exception of a little variation in the ranking of the most used SDECE approaches (Software Development Effort and Cost Estimation).

III - Estimation Approach

The following table lists the advantages and disadvantages of adopting various software development cost estimation techniques:

Table 3: Advantage and Disadvantage of Existing Cost Estimation Methods

Method	Advantages	Disadvantages
Analogy	The benchmark needs to be accurate, Quickly but clearly applied.	The benchmark needs to be accurate. It calls for important, complete data.
Expert Judgment	The experts may consider the effects of the project brought on by new systems, software, and languages included in the future project, as well as the exceptional traits and interactions of the team, among other things.	Expert is still somewhat biased, positive, and cynical, though the community consensus requires them to be that. The expert judgment method also supports methods such as algorithmic method for estimating the opposite expense.

COCOMO	COCOMO is straightforward and genuine. Projects that use historical data are more accurate and reliable. The drivers are particularly useful in determining the effects of the various project cost-influencing components.	The COCOMO model disregarded all of the reports and parameters. It ignores factors like customer participation, knowledge, and competencies. Issues with safety and security have an oversimplified impact.
Function Point Analysis	It is simple to use in early project planning stages. Comparing different projects can be used even if they use different technologies (database, language, etc).	Many cost estimation models such as COCOMO use LOC and thus FPC has to be Modified to LOC.[7]
Use Case Point	UCP (size estimate) should be irrespective of the capacity, ability and expertise of the project implementing team.	Depending on well-written, purpose-oriented use cases. Technical and environmental influences have a very important effect on UCP.
Machine learning method	Training is not necessary. Power of reasoning, unlike databases.	Hard to use, it is difficult to maintain the degree of meaningfulness. There is no layout guideline; the success is based on large data from the training.[22]
Regression Techniques	The value of regression analysis resides in the fact that it offers an effective mathematical tool allowing an organization to analyse the relationship between two or more interest variables.	Just linear regression aims at the longitudinal correlations between dependent variables and independents. That is, it assumes they have a straight-line relation. Sometimes this is incorrect.
Genetic Algorithm (GA)	GA model sometimes exploits the improvement downside by victimization an evolutionary process.	Less flexibility, several hyper parameters which has preference of functions, reproduction rates, the share of ideology and cross over, managing out of certain conditions, making a technique and setting the specified tree sizes and depths inside the model[16].
Agile Process	The corporate people and the engineers must engage on a daily basis while using the Agile methodology. Even in the later stages of development, needs changes are tolerated.	When using the Agile technique, it can be challenging to predict the intended outcome because the requirements are not always clear.
Data Mining Methods	In any case, enormous amounts of data are evaluated using mathematical and statistical techniques before being arranged into a data structure or database for quick access and additional analysis.	Data and privacy issues are one of the main drawbacks of data mining. In the past, businesses would only exchange customer information with other businesses when necessary to deliver a service.[18][38]
Linear Regression and Least Median Squares	Least median squares (LMS) is a strong regression technique that includes outlier detection because it reduces the median rather than the mean (Rousseeuw & Leroy, 1987).	When the data are not normally distributed, test statistics may not be accurate (but with many data points that problem gets mitigated)[19][20]

IV Evaluation Criteria

Different criteria are used to evaluate the performance that was attained using different strategies. These standards are used to assess how well various strategies perform in terms of generated error value and forecast accuracy. To compare the statistical significance of various procedures, one criterion is also applied. A few other factors are also taken into consideration when assessing the effect size, or the size of the therapeutic effect. While the effect size test is independent of sample size, the statistical significance test is reliant on sample size. The complete list of the intended path is as follows:

The performance of model generating continuous output can be assesses in many ways including PRED(42), MMRE, correlation etc., PRED(42) is a measure calculated from the relative error, or RE, which is the relative size of the difference between the actual and estimated value. One way to think about these measurements is to say that training data includes records with variables 1, 2, 3, etc., and performance measures and extra new variables N+1, N+2...

MRE (Magnitude of Relative Error): First, determine the Magnitude of Relative Error for each data point, which measures the degree of estimation error in a single estimate. This stage serves as a model for the following one and is utilised to determine PRED (n). A number of 25% or less indicates satisfactory outcomes [42].

$$MRE = |predicted - actual| / actual \quad (1)$$

MRE (mean magnitude of the relative error): The mean magnitude of the relative error, or MMRE, is the average percentage of the absolute values of the relative errors across the entire data set.

$$MRE = (100/N)* \sum /|predicted_i – actual_i| / actual_i \quad (2)$$

where, N = total number of estimates

PRED (n) Prediction Accuracy: Additionally, a model should be accurate to within 25%, 75% of the time [42]. Divide the total number of data points in a data set with an MRE of 0.25 or less (represented by k) by the total number of data points in the data set to obtain the accuracy rate PRED(n) (represented by n). PRED (n) = k/n, where n = 0.25, is the resulting equation [42]. PRED(n) commonly shows the average percentage of estimates that were within n percent of the actual values. If there are N datasets, then PRED (42) = 50%, for instance, indicates that half of the estimates are within 30% of the actual.

$$PRED(x) =(100/N)*\Sigma_{i.....N} \ 1 \text{ if } MREi <=n/100, 0 \text{ otherwise} \quad (3)$$

The estimations' accuracy directly relates to PRED(x) and, in the opposite direction, to MMRE.

V Summary of Observations

Over the past 30 years, a number of methods for effort estimating have been put out in the field of software engineering. It is feasible to demonstrate that a number of researchers have used function point and COCOMO as input in the construction of prediction models for the effort estimating process. It has been reported that estimates of the effort needed to create object-oriented software are most typically made using the data sets.

It has been demonstrated that traditional techniques, such as the Constructive Cost Estimation Model (COCOMO) and Function Point Analysis (FPA), are insufficient for determining the expense and effort of all forms of software development, because programming language makes use of the ideas of function point (FP) and line of code (LOC), whereas object-oriented design combines data and process, procedural design separates the two. Even so, as Unified Modelling Language (UML) diagrams attract viewers as a method of portraying object-oriented software, the use case point approach derived from Class Diagram and use case point process based on the requirement analysis phases as well as from Use Case Diagram is the solutions, which will have wider acceptance for object-oriented software effort estimation purposes. Given the growing reliance on it, web-based software must be produced rapidly and efficiently.

Implied Accuracy of Basic Research

The contribution of the identified papers is described, along with the estimation method. To examine accurate performance analysis, we have taken accuracy values under the mentioned system among many datasets and approaches.

Table.4 Accuracy of predictions from primary research on standalone approaches

Study	Author(s)Ref.	Estimation technique/contribution	Dataset	MMRE	PRED
[46]	Nassif et al.	Based on the use case points method, the tree boost model is suggested to forecast software effort.	Dataset-1: 58 industrial projects Dataset-2:26educationalprojects	0.29	64
[47]	Satapathy et al.	The prediction values were enhanced using random forest (RF), and UCP was used to evaluate the software effort.	149softwareprojects	0.33	68.25
[48]	Malhotra and Jain	On a dataset from a software project, linear regression, ANN, DT, SVM, and bagging were assessed and compared.	499softwareprojects	0.17	52
[49]	Sharmaand Singh	Utilized support vector machines, multilayer perceptron, and random forest.	4softwareprojects	0.30	72.09
[50]	Pospieszny et al.	Support vector machines, neural networks, and general linear models were combined into an ensemble and averaged.	11variablesofsoftwareprojects	0.13	76.91
[51]	Nassif et al.	According on the UCP model, a regression model was suggested. The estimation was enhanced by using the Sugeno fuzzy inference system (FIS) technique on this model.	Comparison with Karner & Schneider models	0.28	54
[52]	Dan	Particle swarm optimization was used to improve an artificial neural network (ANN) model that was COCOMO-integrated (PSO).	COCOMO-I,NASA	0.40	55.10
[53]	Burgess and Lefley	Critically analysed the utility of genetic programming (GP) for estimating software effort	Desharnais	0.37	28

[54]	Pandey et al	Presented a workable method for deciding which of four well-liked estimate techniques—MLR, MLP-NN, GA, and nave—best fits the effort estimation of apps.	SAMOA	0.9	94
[55]	Singal et al.	Investigation on the effectiveness of the differential evolution algorithm in enhancing the COCOMO and COCOMO II parametric values.	COCOMO81, NASA93	0.22	53.16
[56]	Kumar and Datta	To accurately predict values, an adaptive neuro-fuzzy inference model with bagging and boosting was taken into consideration.	11 industrial projects	0.12	69.37

The various single estimation methodologies were combined to produce unique estimates with comparatively high accuracy. It should be observed that the smaller value for the MMRE assessment measure and the greater value for PRED (25) are both preferable.

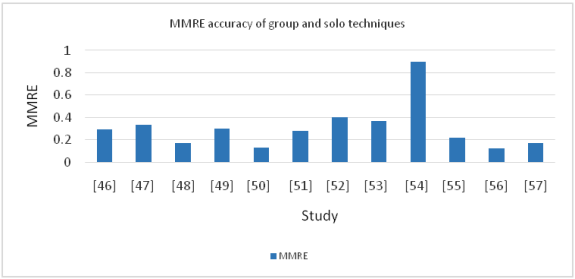

Figure 1. Accuracy value MMRE of different cost estimation techniques

In order to develop web-based software successfully, that is, without any cost or resource (human or otherwise) overruns, the estimations produced prior to the start of development must be exact. As a result, by extending research into the use of the ISBSG dataset for the web effort estimation process and improving prediction accuracy with statistical and machine learning techniques, it will give industries more flexibility as well as better prediction accuracy to estimate the effort of developing web applications. Another interesting area for research is estimating the work required for agile software. Numerous scholars have given various methods for the agile software development process.

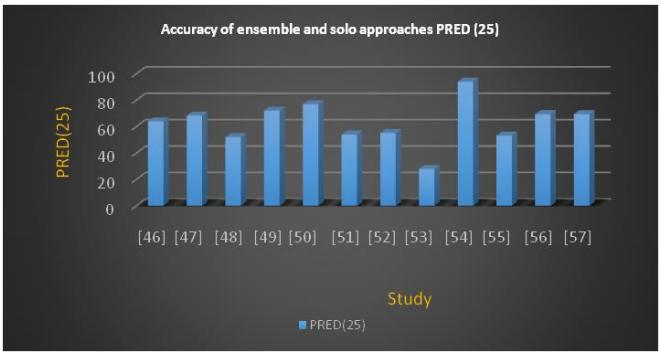

Figure 2. Accuracy levels of several cost prediction models

To provide a rigorous methodology to evaluating the work of software produced using agile methodologies, there are not enough studies currently accessible. The story point technique is one of the most popular methods for estimating the amount of work needed to develop software using the agile methodology. The agile software effort estimating process will be more accurate because of employing various statistical and machine learning algorithms on the dataset of milestones under consideration. Summaries of studies that used a variety of methods for precise cost estimation.

VI Proposed Model, Testing and Results

An Literate Review is a method for locating, assessing, and interpreting all source papers pertinent to a certain research question; an

activities focused area, or a specific phenomenon of interest [3][40]. Secondary investigations rely on observational or experimental studies that have a direct bearing on the topic that needs to be answered [41]. The Literate Review process begins with the formulation of the research questions that must be addressed based on the collection and analysis of data [41]. The methods recommended by Kitchenham[40] were used to conduct this Literate Review. This section explains of the proposed model and results that are obtained after executed it. In the future, the work may be enhanced in the following aspects.

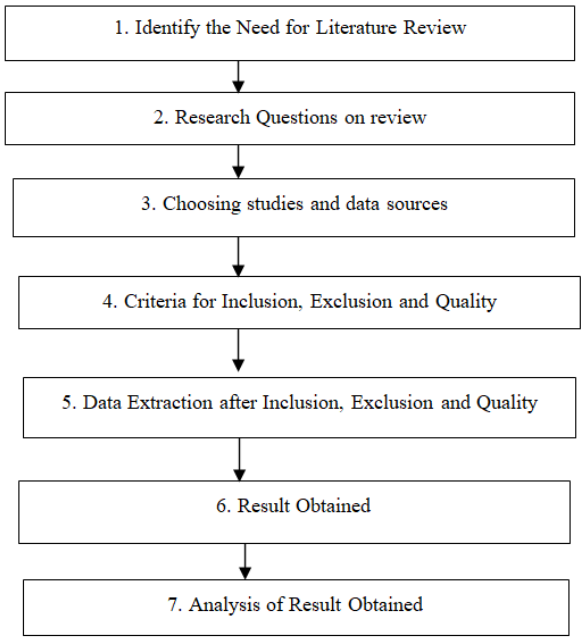

Figure 3. Workflow of Proposed Cost Estimation Model for different Techniques.

VII – Conclusion

In this paper, we compared many methods for estimating the cost of software projects. The findings of this study are listed below confirming these results. Schedule and effort are at least somewhat intertwined. At the early phases of the software development life cycle, cost estimation should be applied as a project feasibility study; budgeting and project planning both require cost estimation. Software project costs are frequently estimated using a variety of techniques. However, it is also important to consider how crucial it is to regularly assess software project expenses and to make a comparison between expected and actual costs at each critical milestone [21]. In this paper, we compared many methods for estimating the cost of software projects. The findings of this study are listed below confirming these results. Schedule and effort are at least somewhat intertwined. At the early phases of the software development life cycle, cost estimation should be applied as a project feasibility study; budgeting and project planning both require cost estimation. It is found that the majority of the company uses model-based methods, such as COCOMO, Use-case-based estimating, and FPA [12]. According to the study, it is evident that the best size estimating methodology for FPA is considered when determining the project's scope. However, because of time constraints and communication challenges, FPA needs to provide specialized instruction to the estimators [8]. A method for measuring the size and complexity of software projects in relation to the functions they provide for the user is called function point analysis. Use cases are used to provide a reliable estimate of the size of the associated future system because it is anticipated that the software project development life cycle will be available very soon. Some businesses use their own tools or methodologies for estimating and deploying them. Iterative models continue to dominate studies on regression-based estimate methods. The most popular methodology used to create cost-estimating models for software projects is regression analysis. The need for precise effort estimates for projects is one of the most crucial and dynamic concerns in the software industry. The accuracy of the prediction models could be increased through automatic collection and other combinations of strategies.

References

[1] Albrecht, "Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation" IEEE Transactions on Software Engineering, Vol. 9, no. 6, 1983.

[2] G. Karner, "Resource Estimation for Objectory Projects," Objective Systems SF AB, Kista, PP 1-9, 1993.

[3] M. Jorgensen et. Al, “A Systematic Review of Software Development Cost Estimation

Studies," IEEE Transactions on Software Engineering, Vol.33, PP33-53, January 2007.
[4] M. Aguiar, "Function Points or Use Case Points?" IFPUG Metric Views Summer, PP 14-15, 2009.
[5] Xiaotie Qin, Miao Fang, China, "Summarization of Software Cost Estimation", Advance in Control Engineering and Information Science, Vol. 15, PP 3027-3031,2011.
[6] Dian Pratiwi, Indonesia, "Implementation of FPA in Measuring the Volume Estimation of Software System in Object Oriented and Structural Model of Academic System", International Journal of Computer Applications (0975 – 8887) Vol. 70, No.10, 2013.
[7] Dr. N. Balaji, N. Shivkumar, India, "Software cost estimation using FP with Non algorithmic Approach", Global Journal of Computer Science and Technology Software and Data Engineering, Vol. 13 Issue 8 Version 1.0, 2013.
[8] Marcos de Freitas Junior, Marcelo Fantinato, "Improvements of FPA Method: Systematic Literature Review", IEEE Transaction on Engineering Management, Vol. 62 No. 4, 2015.
[9] Usharani K., Vignaraj A. V., Tamilnadu-India, "A Survey on Software Effort Estimation," dalam International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), 2016.
[10] Sholiq, Arifin Puji Widodo, Indonesia, "A Model to Determine Cost Estimation for Software Development Projects of Small and Medium Scales Using Use case points", Journalof Theoretical and Applied Information Technology, Vol.85. No.1, PP 87-9, 2016.
[11] Renny Sari Dewi, A. P. Subriadi dan Sholiq, "Use Case Points - Activity Based Costing as an Integration Model for Software Cost Estimation," Second International Conferences on Science in Information Technology (ICSITech), Vol. 5, PP 318-325, 2016.
[12]. Shashank Mouli Satapathy et. Al, "Early stage software effort estimation using random forest technique based on use case points", IET Software, The Institution of Engineering and Technology, Vol. 10, Issue. 1, PP 10–17, 2016.
[13] Atstushi Simoda, Taro Yabuki, "Cost and Value Analysis of Software Development Method Focused on Individual Functions", 8th IEEE International Conference on IntelligentComputing and Information Systems (ICICIS), 2017.
[14] Sholiq, R.S. Dewi, Apol P. Sbriadi, "Comparative Study of Software Development Size Estimation Method: UCPabc vs Function Points", 4th Information Systems International Conference (ISICO-Indonesia), 2017.
[15] Renny Sari Dewi et. Al, "A Modification Complexity Factor in Function Point Method for Software Cost Estimation towards Public Service Application", 4th Information System International Conference, Indonesia, Vol. 124, PP 470-477, 2017.
[16] Murillo-Morera et Al, "A genetic algorithm based framework for software effort prediction", Journal of Software Engineering Research and Development, DOI 10.1186/s40411-017-0037, Vol 5, PP 1-33, 2017.
[17]. Marta Fernandez-Diego[1], Erwin R. Mendez[2], Silvia Abrahao[3], "An Update on Effort Estimation in Agile Software Development: A Systematic Literature Review", Digital Object Identifier 10.1109/ACCESS.2020.3021664, Vol. 8 current version, PP 166768- 166800, 2020.
[18]. Anil Jadhav[1] Mandeep Kaur[2], Farzana Akter[3], Review Article "Evolution of Software Development Effort and Cost Estimation Techniques: Five Decades Study Using Automated Text Mining Approach", Hindawi, Mathematical Problems in Engineering, https://doi.org/10.1155/2022/5782587, Vol. 2022, Article ID 5782587, PP 1-17,2022.
[19]. J. J. Dolado, University of the Basque Country, Spain, Chapter-I "A Two Stage Zone Regression Method for Global Characterization of a Project Database", Advances in Machine Learning Applications in Software Engineering, DOI: 10.4018/978-1-59140-941-1.ch001, PP 1-13, 2007.
[20]. Ali Bou Nassifa,, Danny Hob, Luiz Fernando Capretza, "Towards an early software estimation using log-linear

regression and a multilayer perceptron model", The Journal of Systems and Software, Vol. 86, PP 144– 160, Sci Verse Science Direct, 2012.

[21]. Bhawana Srivastave, Manoj Wadwa, "Relative Analysis of Softawre Cost and Effort Estimation Techniques", International Journal of Computer Science and Engineering (IJCSE) ISSN 2278-9960, Vol. 2, Issue 3, PP 53-68, July 2013.

[22]. Alaa Sheta, David Rine, Sofian Kassaymeh, "Software Effort and Function Points Estimation Models Based Radial Basic Function and Feed-forward Artificial Neural Networks", International Journal of Next-Generation Computing, Vol. 2, No. 3, 2001.

[23] T. J. Gandomani, K. T. Wei, and A. K. Binhamid, ``A Case Study Research on Software Cost Estimation Using Experts' Estimates, Wideband Delphi, and Planning Poker Technique,'' International Journal of Software Engineering and its Applications, vol. 8, no. 11, PP 173-182, 2014.

[24] E. Dantas, A. Costa, M. Vinicius, M. Perkusich H. Almeida, and A. Perkusich, ``An Effort Estimation Support Tool for Agile Software Development: An Empirical Evaluation,'' in 31st International Conference on Software Engineering and Knowledge Engineering (SEKE), doi: 10.18293/SEKE2019-141, PP 82-87, 2019.

[25] M. Usman, E. Mendes, and J. Börstler, ``Effort estimation in agile software development: A survey on the state of the practice,'' in Proc. 19th Int. Conf. Eval. Assessment Softw. Eng., doi:10.1145/2745802.2745813, PP 82–91, 2015.

[26] M. Usman, J. Börstler, and K. Petersen, "An effort estimation taxonomy for agile software development,'' Int. J. Softw. Eng. Knowl. Eng., doi: 10.1142/S0218194017500243 Vol. 27, version 04, PP 641-674, 2017.

[27] P. Rola and D. Kuchta, ``Application of Fuzzy Sets to the Expert Estimation of Scrum-Based Projects,'' Symmetry, doi: 10.3390/sym11081032, Vol. 11, no. 8, PP 1-23, 2019.

[28] S. Porru, A. Murgia, S. Demeyer, M. Marchesi, and R. Tonelli, ``Estimating Story Points from Issue Reports,'' in 12th International Conference on Predictive Models and Data Analytics in Software Engineering (PROMISE), Ciudad Real, Spain, doi:10.1145/2972958.2972959, PP 2:1-2:10, 2016.

[29] H. M. Premalatha and C. V. Srikrishna, ``Effort Estimation in Agile Software Development using Evolutionary Cost-Sensitive Deep Belief Network,'' International Journal of Intelligent Engineering and Systems, doi: 10.22266/ijies2019.0430.25, Vol. 12, no. 2, PP 261-269, 2019.

[30] A. Panda, S. M. Satapathy, and S. K. Rath, ``Empirical Validation of Neural Network Models for Agile Software Effort Estimation based on Story Points,'' Procedia Computer Science, doi:10.1016/j.procs.2015.07.474, Vol. 57, PP 772-781, 2015.

[31] N. V. Prykhodko and S. B. Prykhodko, ``A Multiple Non-Linear Regression Model to Estimate the Agile Testing Efforts for Small Web Projects,'' Radio Electronics Computer Science, Control, doi:10.15588/1607-3274-2019-2-17, no. 2, PP 158-166, 2019.

[32] E. Ungan, N. Çizmeli, and O. Demirörs, ``Comparison of Functional Size Based Estimation and Story Points, Based on Effort Estimation Effectiveness in SCRUM Projects,'' in 40th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), doi: 10.1109/SEAA.2014.83, PP 77-80, Aug. 2014.

[33] S. Dhir, D. Kumar, and V. b. Singh, "An Estimation Technique in Agile Archetype using Story Points and Function Point Analysis,'' International Journal of Process Management and Benchmarking, doi: 10.1504/IJPMB.2017.086933, Vol. 7, no. 4, PP 518-539, 2017.

[34] E. G. Wanderley, A. Vasconcelos, and B. T. Avila,"Using Function Points in Agile Projects: A Comparative Analysis Between Existing Approaches,'' in Brazilian Workshop on Agile Methods (WBMA), doi:10.1007/978-3-319-73673-0_4, PP 47-59, 2017.

[35] E. Dantas, M. Perkusich, E. Dilorenzo, D. F. S. Santos, H. Almeida, and A. Perkusich "Effort estimation in agile software

development: An updated review", Int. J. Softw. Eng. Knowl. Eng., doi: 10.1142/S0218194018400302, Vol. 28, nos. 11-12, PP 1811-1831, 2018.

[36] A. Kaushik, D. K. Tayal, and K. Yadav, "A Comparative Analysis on Effort Estimation for Agile and Non-agile Software Projects Using DBN-ALO," Arabian Journal for Science and Engineering, doi: 10.1007/s13369-019-04250-6, PP 1-14, 2019.

[37] M. Pozenel and T. Hovelja, "A Comparison of the Planning Poker and Team Estimation Game: A Case Study in Software Development Capstone Project Course", International Journal of Engineering Education, Vol. 35, no. 1(A), PP 195-208, 2019.

[38] N. A. Bhaskaran, V. Jayaraj, "A Hybrid Effort Estimation Technique for Agile Software Development (HEETAD),'' IJEAT, doi: 10.35940/ijeat.A9480.109119, Vol. 9, no. 1, PP 1078-1087, 2019.

[39] P. V. de Campos Souza, A. J. Guimaraes, V. S. Araujo, T. S. Rezende, and V. J. S. Araujo, "Incremental Regularized Data Density-Based Clustering Neural Networks to Aid in the Construction of Effort Forecasting Systems in Software Development,'' Applied Intelligence, doi: 10.1007/s10489-019-01449-w , Vol. 49, no. 9, PP 3221-3234, 2019.

[40] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering—Joint technical report," Keele University, Keele, U.K. and University of Durham, Durham, U.K., Tech. Rep. EBSE-2007-01, 2007.

[41] J. Biolchini, P. G. Mian, A. C. C. Natali, and G. H. Travassos, "Systematic review in software engineering," Federal University of Rio de Janeiro, Rio de Janeiro, Brazil, Tech. Rep. RT-ES 679/05, 2005.

[42]. S.D. Conte, H.E. Dunsmore, V.Y.Shen, "Software Engineering Metrics and Models", Benjamin-Cummings Publishing Co., Inc.,ISBN: 978-0-8053-2162-3, PP 1- 396, 1986.

[43]. Rodríguez Montequín, V.; Villanueva Balsera, J.; Alba González, C.; Martínez Huerta, "Software project cost estimation using AI techniques", Proceedings of the 5th WSEAS/IASME Int. Conf. on System Theory and Scientific Computation, Malta, PP 289-293, 2005.

[44]. Ch. P. Rao, P. S. Kumar, S. R. Sree, and J. Devi,''An Agile Effort Estimation Based on Story Points UsingMachineLearningTechniques,''inSecondInternationalConference on Computational Intelligence and Informatics,doi:10.1007/978-981-10-8228-3_20,PP209-219, 2018.

[45]. E. Karunakaran and N. Sreenath, ''A Method to EffortEstimation for XP Projects in Clients Perspective,'' Inter-national Journal of Applied Engineering Research, vol. 10,no.7,PP18529–18550,2015.

[46] Nassif, A. B., Capretz, L. F. and Azzeh, M., "A treeboost model for software effort estimation based on use case points", 11th International Conference on Machine Learning and Applications, 2012.

[47] Satapathy, S. M., Acharya, B. P., and Rath, S. K. Early stage software effort estimation using random forest technique based on use case points. IET Software, doi:10.1049/iet-sen.2014.0122, 10(1), PP 10–17, 2016.

[48] Malhotra, R., Jain, A. Software effort prediction using statistical and machine learning methods. International Journal of Advanced Computer Science and Applications, PP 145–152, 2011.

[49] Sharma, P. and Singh, J. Machine learning based effort estimation using standardization. Paper presented at the 2018 International Conference on Computing, Power and Communication Technologies (GUCON), PP 28–29, 2018.

[50] Pospieszny, P., Czarnacka-Chrobot, B., and Kobylinski, A., "An effective approach for software project effort and duration estimation with machine learning algorithms", Journal of Systems and Software, doi:10.1016/j.jss.2017.11.066, Vol. 137, PP184–196, 2018.

[51] Nassif, A. B., Capretz, L. F., and Ho, D., "Estimating software effort based on use case point model using Sugeno fuzzy inference system", IEEE 23rd International Conference on Tools with Artificial Intelligence, 2011.

[52] Dan, Z., "Improving the accuracy in software effort estimation: using artificial neural network model based on particle swarm optimization", IEEE International Conference on Service Operations and Logistics and Informatics,
[53] Burgess, C. J. and Lefley, "Can genetic programming improve software effort estimation? A comparative evaluation", Information and Software Technology, 43(14), pp 863–873, 2001.
[54] Pandey, M. Litoriya, R., and Pandey, "Validation of existing software effort estimation techniques in context with mobile software applications", Wireless Personal Communications, doi:10.1007/s11277-019-06805-0, Vol. 10(4), PP 1659–1677, 2020.
[55] Singal, P., Kumari, A. C., and Sharma, P., "Estimation of software development effort: a differential evolution approach", Procedia Computer Science, doi:10.1016/j.procs.2020.03.343, Vol. 167, PP 2643–2652, 2020.
[56] Kumar A., and Datta U., "An effort estimation model for software development using ensemble learning. International Journal of Computer Applications, 115(21), 2015.
[57] Pospieszny, P., Czarnacka-Chrobot, B., and Kobylinski, A., "An effective approach for software project effort and duration estimation with machine learning algorithms", Journal of Systems and Software, doi:10.1016/j.jss.2017.11.066, Vol. 137, PP 184–196, 2018.
[58] Alhazmi O. and Khan, M., "Software effort prediction using ensemble learning methods. Journal of Software Engineering and Applications", doi:10.4236/jsea.2020.137010, Vol. 13, PP 143–160, 2020.
[59] Satapathy, S. M., and Rath, S. K., "Empirical assessment of machine learning models for agile software development effort estimation using story points", Innovations in Systems and Software Engineering, doi:10.1007/s11334-017-0288-z, Vol. 13(2), PP191–200, 2017.