*i*Journals: International Journal of Software & Hardware Research in Engineering (IJSHRE)
ISSN-2347-4890
Volume 9 Issue 5 May 2021

Software Development Using Productivity Metrics and Different Cost Estimation Technique

Asmaa' Hadi AL_Bayati
College of Computer Science and Mathematics, University of Mosul, Mosul, Iraq
asmashade77@uomosul.edu.iq

ABSTRACT

The academic community needs a confidentiality system that are ease to use by student and faculty members. This system depends on the generating process of a random identification numbers (ID No.), that are not repeatable and have non-repetitive numerical categories to facilitate their handling. In addition to being highly reliable, these numbers are placed on the exam papers instead of the student's information such as the name and exam material School stage and department. The system was designed using Arduino due microcontroller and its development environment. Arduino webserver monitoring system have many facilities that can be programmed using the C programming language to verify monitoring options.

One of the significant and challenging activities in software development is the effective software cost estimation. Since the software cost estimation in all software projects is considered to be extremely complicated, challenging and confusing works for all software companies and software cost estimation is the most basic phase for starting software projects, since it is providing an overview related to resources, efforts and schedule/time needed for software projects with regard to software company costs. Generally, software project is based on the software cost estimation since it is providing primary idea regarding tracks, risks and challenges included in software project development. Also, developers are requiring accurate and straightforward approach for effort estimation. Cost estimation prior to starting work is considered to be a prediction, while predictions are not constantly precise. The software effort estimation can be defined as one of the essential tasks in software engineering as well as for controlling the efficiency and quality, thus an adequate technique for estimation is vital. The presented work is providing thorough overview regarding the current models and approaches for software cost estimation, also providing comparative study on the basis of the usefulness and efficiency of such models. There is no single approach considered to be optimum for all conditions, and therefore adequate comparison regarding the results of many methods will probably generate realistic estimation. generally, this paper that is make a comparison between the types of cost estimation in a number of research and studies that have depends on different data and which later will provide the correct path for researchers in choosing the appropriate cost estimation technique and thus the efficiency of the software development.

Keywords: Software cost estimation, Estimation techniques, Cost models, artificial neural networks.

1. INTRODUCTION

Software cost estimation is of high importance in software engineering, majorly specifying the failure of success of project execution and contract negotiation. The major goal of software effort and cost estimation is scientifically estimating the needed workload as well as its equivalent costs in the life-cycle of software systems [1]. In addition, cost estimation might be specified as approximation or process of possible costs related to projects, programs, or products, evaluated based on provided information. Precise estimation of costs is significant to all project types, if projects are not estimated adequately; then there will be very high project costs, often being in range of (150 and 200) % over original costs. Thus, projects must be properly estimated [2]. The majority of software development

projects are highly dependent on software cost estimation. Regrettably, the software development cost estimation has inaccurate and difficult task. Despite the availability regarding various estimation approaches, guidelines and procedures, enhancement remain a major aim. Analyzing cost estimation error is one way to reduce software cost estimation errors [3]. Starting from the 1950s, software engineering cost models as well as estimation approached were utilized for some will power, including investment analysis, software budgeting, software control, software planning, software improvement, and risk analysis. Recently, the major costly element of computer system projects is software. Accurate and reliable software cost estimates were vital for clients/customers and developers [4].

To effectively manage a software project, there is high importance in estimating software efforts and costs efficiently. One of the major demanding, inspiring and challenging in each of the software industries is to expect accurate software effort and cost estimation. With regard to projects which are huge, with high budget, significant and having temporal constraints, and adequate software effort and cost estimation is of high importance and analyzing and predicting adequate software effort and cost estimation is more complicated[5].

The possibilities to have accurate software effort and cost estimation is going to be high, medium as well as low, since as the software complexity is increased, it will be more complicated to precise estimation regarding software
efforts and costs. Also, the relation between the software projects' complexity and the possibility to have adequate software effort and cost estimation might be utilized via management to assess the projects proposal as well as for efficient management regarding the process of software development [3]. Thus, accurate estimation of the costs related to software development has major economic impact: actually, about 60% related to large projects are exceeding their software cost estimates as well as the projects which have never been accomplished due to inaccurate estimation of the software development costs. Majorly, effort is considered as the major driving force for costs and due to such fact, software cost estimation is an issue of software effort estimation. A lot of researchers and scientists are attempting to develop more precise and novel approaches for software cost estimation. The majority of software cost estimation methods were on the basis of algorithmic models, machine learning and expert judgment methods [5,6]. Precise estimation of costs is significant because:

a) For measuring the effect of re-planning, support planning and variations.
b) For defining the resources required for projects and how effectively such resources are used.
c) For categorizing and arranging the development projects in terms of total business plan.
d) The projects might be easier to manage and control in the case when resources are effectively matched to real requirements.

2. THE PROCESS OF SOFTWARE COST ESTIMATION

The process of cost estimation is set of procedures and approaches used by organization for arriving at cost estimation, (Figure 1) is showing the process of estimation. There are 7 steps involved in the process of cost estimation [7]:

a) Establishing cost estimating objectives
b) Generating a project plan for the needed resources and data
c) Identifying the requirements of software
d) Working out all details regarding the software system as possible

e) Using many independent cost estimation approaches for capitalizing on their combined strengths

f) Comparing various estimates and iterating the process of estimation

g) Following the start of the project,

monitor its actual progress and costs, and feedback results to project management.

Fig 1 : Asoftware Cost Estimation Process [8,9]

3. MODELS OF COST ESTIMATION

In 1960, cost estimation was created. Typically, there are various approaches for software cost estimation that will be divided into 3 groups [3,7,10]:

a) Algorithmic techniques.

b) Non algorithmic techniques.

c) Machine Learning techniques.

With regard to algorithmic technique, the formula is used to calculate cost estimation. Also, the formula will be generated via combining associated cost factors in different models. Also, the non-algorithmic approaches are not using any formula for calculating the software cost estimation. The two groups were significant to perform precise estimation. In the case when requirements were known well, then the performance is going to be better [10]. With regard to machine learning techniques, training rules are used to estimate and repeat run cycles. Thus, this might be adequate method since it is increasing the accuracyof results [3]. (Figure 2) showing the techniques and methods for software cost estimation.

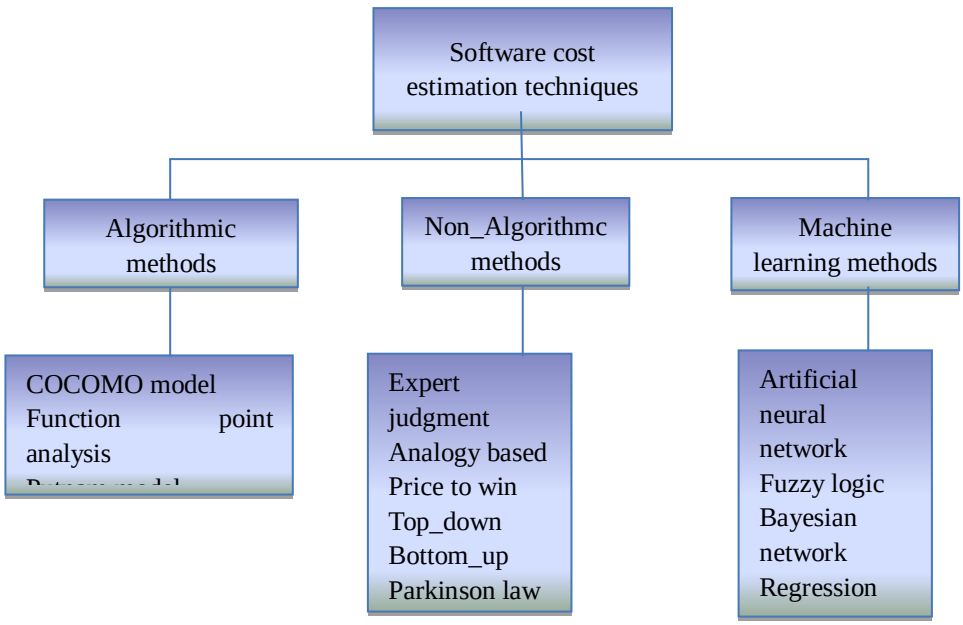

Fig 2: Cost Estimation Models [9]

4. PROS AND CONS CURRENT COST ESTIMATION TECHNIQUES

The next section is describing the Pros and cons of current cost estimation approaches, the description might be of high importance to choose adequate approach in specific project. Table 1 and 2 are showing a comparison regarding the mentioned approaches for estimation. Common current approaches of estimation were selected to do such comparison.

Table 1: Pros of Cost Estimation Techniques [3,10,11,12]

Technique	Type	Pros
COCOMO model	Algorithmic	Extremely mutual (extremely common), pure results, transparent, can see the way it works
Function point analysis	Algorithmic	Layout independent, language independent, results were enhanced compared to source lines of code
Putnam	Algorithmic	Fit for large-scale projects, easy and fast
Expert Judgment	Non-Algorithmic	Fairly inexpensive method for estimation. Might be adequate as the experts having direct experience regarding comparable systems, adapting to certain projects, rapid prediction, using experience on the past projects for assessing factors on new project
Analogy	Non-Algorithmic	Works according to actual experience, there is no requirement to have especial expert, accurate with the availability of project data
Top-down	Non-Algorithmic	Focusing on high-level activities or the system
Bottom-up	Non-Algorithmic	Estimating each software component result in thorough results of estimation
Price to win	Non-Algorithmic	Sometimes getting the contract
Parkinson's Law	Non-Algorithmic	No-overspend, correlating with certain experience
Delphi Approach	Non-Algorithmic	Free of social pressure, individual dominance, and personality influence, allow sharing of reasoning and information between participants, typically not biased, group discussions ensuring the issues of estimation aren't discarded
Neural Network	Machine Learning	Consistent with unlike databases, Power of reasoning
Fuzzy Logic	Machine Learning	No need for training, flexibility, no requirement for real formal training
Case-Based Reasoning	Machine Learning	No need for expert, processes comparable to human thinking, handle failed cases
Regression trees	Machine Learning	Resisting data outlier, might be modified to new data, grasped easily, adequate for non-numerical variables

Table 2: Cons of Cost Estimation Techniques [3,10,11,12]

Technique	Type	Cons
COCOMO	Algorithmic	A lot of data is needed, it isn't adequate for all projects, success largely depending on model's tuning to requirements of organizations, utilizing historical data that isn't constantly available
Function point analysis	Algorithmic	It is hard to do mechanization, output's quality isn't considered
Putnam	Algorithmic	Absence of certain details needed for estimating some software parts
Expert Judgment	Non-Algorithmic	Extremely imprecise when there are not experts, thus its success is depending on experts, typically is done incomplete
Analogy	Non-Algorithmic	Much information regarding past projects is needed, in a few conditions there is no comparable project, requires systematically maintained cost data-base
Top-down	Non-Algorithmic	Might not be covering low-level components and activities
Bottom-up	Non-Algorithmic	A lot of time is required, resources for implementing it. Needs understanding of components
Price to win	Non-Algorithmic	Typically producing large overruns, the possibility that the customer is getting the system they want is narrow
Parkinson's Law	Non-Algorithmic	Systems are typically unfinished, reinforces poor practice
Delphi Approach	Non-Algorithmic	Consume a lot of time, requiring skills in written communications, requiring suitable time as well as participant commitment (requiring 30-45 days for completing the whole process), panel must be extremely experienced
Neural Network	Machine Learning	There are not designing guidelines, performance is depending on large training data that aren't often available
Fuzzy Logic	Machine Learning	Complicated to use, difficult to maintain the degree of meaningfulness
Case-Based Reasoning	Machine Learning	Difficult to combine cases, prediction is considered to be limited to the cases which were identified
Regression trees	Machine Learning	High sensitivity to the algorithm utilized for constructing a tree depth, no ability for eliminating a which is value outside the range provided in training data

5. SOFTWARE PROJECT DEVELOPMENT EXISTING PAPERS

The software industry has to have efficiency. As a result of the fast technological changes, implementing the complicated software systems at lower costs and the need for maintaining software of higher quality are two of the main issues for software companies. A very tough work is the estimation of

the cost, in software engineering area. It is estimating the entire cost which is needed for software development [13].

A variety of the models for effort estimations were advanced throughout the past 40 years. The most widely utilized approaches for the prediction of software development attempts are the function Point Analysis and Constructive Cost Model (COCOMO) [14]. The function point analysis can be defined as an approach of the quantification of the complexity and the size of the software system based on the functions that a system is delivering to a user [15]. The function is not dependent upon tools or programming languages which have been utilized for the development of the software project [16]. The COCOMO has been advanced by Boehm [17]. It has been based upon the linear-least-squares regression. With the use of the LOC (i.e. Line Of Code) as a measure unit for the actual software size includes numerous issues [18]. Those approaches have not been successful in dealing with implicit nonlinearity and interactions between project characteristics and efforts [19, 20].

Recently, several alternative approaches of the modeling were suggested, which included Artificial Neural Networks ANNs, fuzzy system, analogy-based reasoning and ensemble methods. Ensemble has been utilized for combining the results of the individual approaches [21, 22]. In the analogy-based cost estimations, the measures of similarity between two projects have an essential impact [23]. Such model type performs the calculation of the distance between software project which is estimated and every historical software project and after that, performs the retrieval of the most similar project for the generation of effort estimation [24].

In 2003, Stamelos I. etal. proposed a method for estimating the costs of the software development. The approach has been based upon characterizing the software which will be developed based on the attributes of the environment and the project and comparisons with a few of the similar completed project (or projects) which have been recovered from historical data-base. A case study has been presented as well, which has been focused upon calibrating and applying the approach on 59 information systems, which implement the functions of the supply chains in the industry. A variety of the strategies have been discovered, the best of which predicted the effort rather efficiently, with a 24% average estimation error based on actual effort [4]. In the same year Shepperd and Lefley, implemented the genetic programming for improving cost estimation of the software on the public data-sets with a considerable success [25].

In 2005, and 2008, Jingzhou et al. have implemented the ANN to the estimation of the cost. The NN is capable of the generalization from the trained dataset. Through a training dataset, the NN learning approach constructs mappings which fit data, and fits the previously unseen data in reasonable ways [26,27].

In 2008, V. kumar et al. have utilized the wavelet Neural Networks NNs for prediction of software cost estimations. Unfortunately, those models' accuracies have not been good enough, which is why there has always been a likelihood for higher accuracy methods of software cost estimations [28].

In 2010, I. Attarzadeh and S. H. Ow have provided an approach for the estimation of the software cost which has a performance superior to the rest of the approaches on a specific group of test cases. Which has been carried out by presenting an innovative model to handle the uncertainty and the imprecisions through the use of NNs. Therefore, the present study was providing an approach for the software cost estimations, performing better compared to other approaches on effort estimation accuracy. This study showed through the application of the NNs on algorithmic and non-algorithmic models of the software effort estimations, the precise estimation has been practicable, and suggested NN model has shown more accurate estimates of the software efforts in view of mean magnitude of relative error, PRED (0.25) criteria of evaluation in comparison with conventional COCOMO [29].

In 2010, S. Kang et al. have produced a model of estimation for the agile based projects, according to the function points, which are commonly utilized for the estimation of the efforts and costs which have been required for developing the product. This method has been commonly utilized in the conventional method. In agile, the most commonly accepted approach of the estimation has been based upon the story points. In that study, they included the method of the function point besides the story points for the achievement of the maximum accuracy degree. The status of the project has been tracked dynamically by using the Kalman filtering approach. Validation has been carried out using a case study through the comparison of the results to conventional approaches [30].

In 2011, B. Boehm, Valerdi R. has carried out a research, which has been based upon the literature analyses, conducting surveys and interviews with the main resource estimators and users, and has concluded that the research on the resource estimation directly impacts the practices of the software engineering [31].

In 2012, E. Kocaguneli identified an assumption concerning the analogy-based estimation of efforts. Those assumptions have been tested the through obtaining binary tree of the clusters and comparing the super-trees' variance against the sub-trees. It has been discovered that the estimation of the super-trees variance have been smaller compared to the sub-trees, is implying the fact that if the cluster variance has been lower compared to the estimation has minimum error occurrences. This study concluded the fact that dynamically selecting the nearest neighboring project through the use of the project data with lower variance considerably enhances the estimation by the analogy [32].

In 2012, E. Kocaguneli et al. Have suggested an approach combining several approaches of estimation to one. The optimal single independent approaches of the estimation have been selected, the chosen approach was implemented afterwards to data-sets and then validated with the use of 7 critical measures of the error. This study has confirmed that collections of several solo approaches have been more precise and consistent for the estimations, in comparison with independent approaches [33].

In 2012, K. Lind and R. Heldal method is the estimation of the embedded software components' size. For the embedded system applications, the estimation of the precise size of the code early saves massive amounts of effort and costs for the component development. The Functional size measurement approach was implemented in a variety of the automotive types of industry. This research has proven that there is a strong correlation between the code size and the functional size that is necessary to obtain more precise results of the estimation [34].

In 2013, Sweta Kumari and Shashank Pushkar are developed COCOMO by providing the utilization of the Support Vector Regression (SVR) for estimating the effort of the software projects. Simulations have been performed with the use of COCOMO data-set. They utilized the weka (i.e. data mining) tools for the simulations due to the fact that it includes different-different approaches of machine learning which may be helpful for the easy classification of data. results have been compared against Multiple Objective Particle Swarm Optimization (MOPSO) as well as Intermediate COCOMO models. The model precision is measured based on its error rate. It has been observed from results that the SVR provides more sufficient results. On the testing of the model efficiency in terms of Mean Absolute Relative Error (MARE) and Predicting the results have been discovered useful [35].

In 2013, A. Ren and C. Yun have conducted researches on the most commonly utilized models of the size estimation such as the analogy based Program Evaluation and Review Technique (PERT), Function Point, Delphi as well as other approaches. Not all of the methodologies have been appropriate for all project types. According to the benefits and the drawbacks of every model, this method has suggested on the way of choosing the most suitable approach for certain requirements of the project. None-the-less, this method might not be applicable for agile based projects because requirements in the agile are instable and not well defined in advance as well [36].

Use case point estimation is a famous model which is utilized to estimate in the agile software development. In 2013, Parvez has advanced a new layer in available use case model of point estimation, where they proposed 2 contributing factors, which are: the efficiency and the risks for the estimation of efforts which are needed for the test. The available use case point approach has considered only the project characteristics, however, this study has been focused upon the team characteristics, as well as the project. The significant aspects which should be taken under consideration in new layer have been Test team resources, testing Weight, Cycle length, risk factors, and Efficiency factor. Introducing a new layer in available use case point results in improving the estimation performance and effectiveness [37].

In 2013, A. Bou Nassif etal. Have performed the comparison of the DTF (i.e. the Decision Tree Forest) model with the model of the Decision Tree (DT) and multiple Linear Regression Model (MLR) for the estimation of the effort of development which is required for the project. The abovementioned models have been assessed and results have been compared with the criteria of the evaluation, like the MMRE and PRED(x). The results have proven that the model of the DTF has outperformed the rest 2 models (i.e. the MLR and the DT) based on the precision of the effort estimation. None-the-less, as a result of such

heavy-weight method, the model might not be appropriate for the agile based projects [38].

In 2013, Waghmode, R.M., Patil, L.V. and Joshi, S.D. have proposed the approach of the reduction, referred to as feed-forward NN with the PCA. The major goal of the authors was using this for the purpose of measuring the software cost estimation model's precision. The suggested method has been based upon algorithmic as well as non-algorithmic approaches. Which is why, they have utilized a combination of the algorithmic approach (i.e. the COCOMO) and the non-algorithmic (i.e. the NN) for the estimation of the costs of the software project [39].

In 2015, E. Khatibi have suggested an innovative approach for addressing the task in an Analogy based estimation system, which has been more commonly utilized in the past years, due to its simplicity and estimation capabilities that have been utilized to estimate efforts which are needed for product development. The previous approaches, in comparison with the 2 associated projects with no consideration of their internal attributes, leading to the biased and inaccurate estimations. This study has been focused on the development of hybrid model for addressing the abovementioned issue. The associated projects have been categorized as set of the clusters through the consideration of the internal attributes of the project like the platform of development, the type of the organization and the expertise level. After that, the process of the attribute weighing has to be carried out and the effort of the development has been analyzed for every

one of the clusters of projects. The results that have been accomplished have been validated through the comparison to existing model, which provides very good result concerning the metrics of performance and accuracy [40].

The analogy-based estimation of the effort is a very prominent technique which is utilized for handling the noisy data-sets. Same amount of the analogies might not be appropriate for all project type for making precise estimations. In 2015, M. Azzeh and A. B. Nassif have suggested an innovative approach, which has been based upon the bisecting k-medoids clustering algorithm for coming up with a group of the analogies for the individual projects. The properties of the data-set have been gathered through the application of the abovementioned algorithm which is, in turn, capable of automatically finding the group of the analogies for every one of the projects. Results have been compared with the conventional approaches of the analogy-based estimation and the suggested approach has delivered better results and has shown superior efficiency [41].

In 2015, S. Garg et al. have suggested a model of cost estimation, which that suited the projects of Agile software development. Sakshi has characterized the characteristics which are maximally correlated and proposed the PCA (i.e. the Principle Component Analysis) for the reduction of the amount of the considerable characteristics. The suggested method has been appropriate, even with the absence of the expert opinions and the statistical data. The results from their method has proven to be having a higher accuracy and precision of the cost estimation in the agile software development project [42,43].

The Story Point Approach (SPA) has been the most commonly utilized method in the agile software effort estimations. In 2015, A. Panda et al. have improved the estimation precision in the agile projects according to the NNs. This method has considered various NN types such as the General Regression NNs (GRNNs), polynomial NNs and probabilistic NNs for improving the effort estimation accuracy. This approach has considered the best fit for the estimation of efforts, none-the-less it did not solve other estimation aspects, such as the duration, risk or cost [44].

In 2016, K. Moharreri et al. have suggested an automatic approach of the estimation, which is referred to as the "Auto Estimation" so as to estimate the efforts for the agile based projects. Such method has been complementing to the commonly utilized manual planning poker approach. The optimal algorithm of learning has been automatically chosen through following those steps: collection of the data, through the use of the story cards, extraction of features with the use of the textual analyses,construction of the model with the obtained characteristics and carries out the analyses through performance measurement. It has provided promising results as well concerning to the precision, in comparison to the common approach of the planning poker [45,46].

In 2016, P. Rijwani and S. Jain proposed using an NNbased approach, which has been advanced technologically, with the use of the Multi Layered Feed Forward NN that has been given training with the Back Propagation training approach. The COCOMO dataset has been produced for the purpose of testing and training the network. MMRE and

Mean Squared Error (MSE) have been utilized as indices of performance measurements. The outputs of the experiments have suggested that the proposed model may give more sufficient results and precisely predict the effort which is required for the software development [47].

ANNs and the Analogy-Based estimations can be considered as the most widely utilized approaches for the estimation of efforts which are needed for the software project developments. In 2017, K. Bardsiri et al. have suggested a hybrid approach, which is an integration of the fuzzy clustering, analogy-based and ANN approaches, which are utilized for the improvement of the effort estimation accuracy. The associated projects have been clustered for the purpose of reducing the irrelevant and inconsistent projects, and that in turn will improve the precision of the estimation. The suggested model was validated according to the metrics of efficiency, like the Mean Magnitude of Relative Error (MMRE) and Percentage of the Prediction (PRED) (0.25) and the prospective results which have proven that the proposed approach has been more efficient compare to the other approaches [48].

6. CONCLUSIONS

Nowadays, nearly none of the models are capable of estimating the software cost with a high accuracy level. This practice state has been produced due to the following reasons:

a) There are numerous interrelated aspects, influencing the process of the software development of a certain team of development and numerous attributes of the project, like the amount of the user screens, system requirement volatility and utilizing the reusable components of the software.
b) The environment of the development is constantly evolving.
c) The lack of the measurement which is an actual reflection of the software system complexity.

For the purpose of producing a more sufficient estimation, it is necessary to enhance the comprehension of those project attributes as well as their causal correlations, model the impact of the evolving environments, and developing the efficient ways to measure the complexity of the software.

Earlier approaches of the estimation for predicting software development efforts were confronted with irrelevant and inconsistent projects it have decreased the estimates' precision. The vague and uncertain software projects' nature has resulted in increasing the complexity of this issue.

In the present paper the software cost estimation approaches have been presented, besides several of the software cost estimation methods, which include algorithmic approaches, estimating through the analogy, top-down approach, expert judgment approach, as well as the bottom-up approach. In addition to that, an insight has been given about the methods of the software cost estimations for researchers that have been newly introduced to this field. There isn't a single approach which is necessarily worse or better compared to the others, actually, their strength and weakness points have been usually complimentary to one another. In the absolute sense, no single model performed especially well at the estimation of the software development efforts.

the purpose of picking an approach for a certain project, a suitable project analysis is quite necessary. An incorrect method of estimation has the ability of considerably delaying a project, and an appropriate approach has the ability of making a project breeze through the deadline. A thorough global factor analysis has to be carried out, on the contrary, it turns quite hard giving accurate deadlines. In addition to that, there is a necessity in emphasizing the smaller details, due to the fact that they are capable of causing delays in the cases of being added up together. The observations have shown that the best thing is using several different approaches of estimation or cost models for project manager, and after that, results have been compared, prior to the determination of reasons for the large variations and documentation of any assumptions which have been made while making estimates.

The observations showed that it has been best using several different approaches of estimation or cost models, and after that, comparing results prior to the determination of reasons for any large variations. No approaches are necessarily more or less sufficient compare to the others. It has been actually discovered that their benefits and drawbacks are usually complementing one another. Which is why, the major conclusion which has been drawn is that there isn't a single approach which has been the ultimate for all situations, and results several different methods must be considered carefully for discovering what is the most possibly to be producing the realistic estimations.

7. ACKNOWLEDGEMENT

The author would like to express their utmost gratitude to University of Mosul/ College of Computer Sciences and Mathematics for the facilities that have been provided by them, which were helpful in improving this work's quality.

8. REFRENCES

[1] Pa Pa Win, War War Myint, Hlaing Phyu Phyu Mon, Seint Wint Thu, "Review on Algorithmic and Non-Algorithmic Software Cost Estimation Techniques", International Journal of Trend in Scientific Research and Development (IJTSRD), e-ISSN: 2456 – 6470, Volume 3 Issue 5, August (2019).

[2] Narendra Sharma and Ratnesh Litoriya, "Incorporating Data Mining Techniques on Software Cost Estimation: Validation and Improvement", International Journal of Emerging Technology and Advanced Engineering, 2012, vol.-2.

[3] Dawid Zima, "MODERN METHODS OF SOFTWARE DEVELOPMENT", TASK QUARTERLY vol. 19, No 4, 2015, pp. 481–493.

[4] Jones, C., "Software Quality in 2002: A Survey of the State of the Art", Software Productivity Research, Marlborough, Massachusetts, (2005).

[5] Boraso, M., Montangero, C., and Sedehi, H., "Software cost estimation: An experimental study of model performances", Technical Report TR-96-22, del Dipartimento Di Informatica, Universita Di Pisa, Pisa, Italy, (1996).

[6] IA Saleh, AH AL_Bayati, K Hadi Thanoon ," Measure the Software Quality based on GrasshopperOptimization Algorithm", International Journal of Computing and Digital Systems, 2020.

[7] Hareton, L. and Zhang F., "Software Cost Estimation. In: Handbook of Software Engineering and Knowledge Engineering", Volume 2: Emerging Technologies , World Scientific Publishing Co Pte Ltd., Singapore, 307-324, 2002.

[8] Shivani Sharma, Aman Kaushik and Abhishek Tomar, "Software Cost Estimation using Hybrid Algorithm", International Journal of Engineering Trends and Technology (IJETT), ISSN: 2231-5381, Volume 37 Number 2- July 2016.

[9] Sai Mohan Reddy Chirra and Hassan Reza, "A Survey on Software Cost Estimation Techniques", Journal of Software Engineering and Applications, 12, 226-248, DOI: 10.4236/jsea.2019.126014 Jun. 30, 2019.

[10] Tannu and Yogesh Kumar, "Comparative Analysis of Different Software Cost Estimation Methods", International Journal of Computer Science and Mobile Computing, ISSN 2320–088X, Vol.3 Issue.6, June- 2014, pg. 547-557.

[11] Muhammad Indra Zul Aqlani and Nuning Septiana, "Comparative Analyisis of Software Cost Estimation Project using Algorithmic Method", Engineering Software Requirements Vol. 1, No. 1, 2018.

[12] Mahmud Alkoffash, Mohammed J. Bawaneh and Adnan Alrabea, "Which Software Cost Estimation Model to Choose in a Particular Project", Journal of Computer Science 4 (7): 606-612, 2008, ISSN 1549-3636.

[13] Shivangi Shekhar and Umesh Kumar, " Review of Various Software Cost Estimation

Techniques", International Journal of Computer Applications 141(11):31-34, DOI:10.5120/ijca2016909867, May 2016.

[14] B. Kitchenham, L.M. Pickard, S. Linkman and P.W. Jones, "Modelling software bidding risks," IEEE Transactions on Software Engineering, 2003, pp. 542–554.

[15] J.E. Matson, B.E Barrett and J.M. Mellichamp, "Software development cost estimation using function points," IEEE Transactions on Software Engineering, 1994, pp. 275–287.

[16] A.J. Albrecht and J.E. Gaffney, "Software function, source lines of code, and development effort prediction: a software science validation," IEEE Transactions on Software Engineering, 1983, pp. 639–647.

[17] G.W. Flake, S. Lawrence, Efficient SVM regression training with SMO, Mach. Learn. 46 (1–3) (2002) 271–290.

[18] A.Idri, T.M. Khosgoftaar and A. Abran, "Can neural networks be easily interpreted in software cost estimation," World Congress on Computational Intelligence, Honolulu, Hawaii, USA, 2002, pp. 12–17.

[19] A.R. Gray,"A simulation-based comparison of empirical modelling techniques for software metric models of development effort," In: Proceedings of ICONIP, Sixth International Conference on Neural Information Processing, Perth, WA, Australia, 1999, pp. 526–531.

[20] IA Saleh, OI Alsaif, MA Yahya ," Optimal distributed decision in wireless sensor network using gray wolf optimization",IAES International Journal of Artificial Intelligence, 2020.[21] K.K. Aggarwal, Y. Singh, P.Chandra and M.Puri, "An expert committee model to estimate line of code," ACM New York, NY, USA, 2005, pp. 1-4.

[22] K. Vinay Kumar, V. Ravi and Mahil Carr, "Software Cost Estimation using Soft Computing Approaches," Handbook on Machine Learning Applications and Trends: Algorithms, Methods and Techniques, Eds. E. Soria, J.D. Martin, R. Magdalena, M.Martinez, A.J. Serrano, IGI Global, USA, 2009.

[23] Y.F. Li, M. Xie and T.N. Goh, "A study of project selection and feature weighting for analogy-based software cost estimation," Journal of Systems and Software, 2009, pp. 241–252.

[24] Chiu N. H. and Huang S. J., "The adjusted analogy-based software effort estimation based on similarity distances", Journal of Systems and Software, Vol. 25, pp. 628–640, (2007).

[25] M. Lefley and M. J. Shepperd, "Using Genetic Programming to Improve Software Effort Estimation Based on General Data Sets", LNCS, Genetic and Evolutionary Computation-GECCO 2003, ISBN: 978-3- 540-40603-7, page-208.

[26] Jingzhou, L. and Guenther, R., "Analysis of attribute weighting heuristics for analogy-based software effort estimation method AQUA+", in Proceedings of Empirical Software Engineering Journal (2008), Vol. 13, No. 1, pp. 63–96, Feb (2008).

[27] IA Saleh, OI Alsaif, KH Thanoon," Deep Coverage Strategy for Private Wireless Network Power Using Hybrid (Salp Optimization – Genetic) Algorithms, Technology Reports of Kansai University, January, 2020.

[28] K. Vinaykumar, V. Ravi, M. Carr and N. Rajkiran, "Software cost estimation using wavelet neural networks," Journal of Systems and Software, 2008, pp. 1853-1867.

[29] Iman Attarzadeh and Siew Hock Ow, "Proposing a New Software Cost Estimation Model Based on Artificial Neural Networks", 2nd International Conference on Computer Engineering and Technology, Volume 3, (2010).

[30] Sungjoo Kang, Okjoo Choi, Jongmoon Baik, "Model based dynamic cost estimation and tracking method for Agile Software Development," in Computer and Information Science (ICIS), IEEE / ACIS 9th International conference on, 2010: 743-748.

[31]Boehm, B., Valerdi, R, "Impact of software resource estimation research on practice: a preliminary report on achievements, synergies, and challenges," in Software Engineering (ICSE), 33rd Internation Conference, 2011.

[32]Kocaguneli, E, "Exploiting the essential assumptions of Analogy-Based Effort Estimation", IEEE Transactions on Software Engineering, 2012; 38 (2): 425-438.

[33]Kocaguneli, E, Menzies, T, Keung, J W, "On the value of ensemble effort estimation," in Software Engineering, IEEE Transactions on, 2012; 38 (6): 1403-1416.

[34]Lind, K, Heldal, R, "A Practical Approach to Size Estimation of Embedded Software Components," in Software Engineering, IEEE Transactions on, 2012; 38 (5): 993-1007.

[35]Sweta Kumari and Shashank Pushkar, "Comparison and Analysis of Different Software Cost Estimation Methods", International Journal of Advanced Computer Science and Applications (IJACSA), Vol. 4, No.1, 2013.

[36]Aihua Ren, Chen Yun, "Research of Software Size Estimation Method," Cloud and Service Computing (CSC), International Conference on, Beijing, 2013.

[37]Parvez, A W M M, "Efficiency factor and risk factor based use case point test effort estimation model compatible with agile software development," Information Technology and Electrical Engineering (ICITEE), International Conference on, Yogyakarta, 2013: 113-118.

[38]Nassif, A B, Azzeh, M, Capretz, L F, Ho, D, "A comparision between decision trees and decision tree forest models for software development effort estimation," in Communications and Information Technology (ICCIT), Third International Conference on 2013: 220-224.

[39]Waghmode, R.M., Patil, L.V. and Joshi, S.D. (2013) "A Collective Study of PCA and Neural Network based on COCOMO for Software Cost Estimation". International Journal of Computer Applications, 74.

[40] IA Saleh, WA Alawsi, OI Alsaif, K Alsaif ," A Prediction of Grain Yield Based on Hybrid Intelligent Algorithm ",Journal of Physics, 2020 .

[41] Azzeh, M, Nassif, A B, "Analogy-based effort estimation: a new method to discover set of analogies from dataset characteristics," in IET Software, 2015; 9 (2): 39-50.

[42] Garg, S, Gupta, D, "PCA based cost estimation model for agile software development projects," Industrial Engineering and

Operations Management (IEOM), International Conference on, Dubai, 2015: 1-7.

[43] Z. Oleiwi, K. Thanoon, K. Alsaif," High Frequency Coefficient Effect on Image Based on Contourlet Transformation", 2019 International Conference on Computing and Information Science and Technology and Their Applications (ICCISTA),2019.

[44] Aditi Panda, Shashank Mouli Satapathy, Santanu kumar Rath, "Emprical validation of Neural Network Models for Agile Software Effort Estimation based on Story Points," Procedia Computer Science, 2015; 57: 772-781.

[45] Kayhan Moharreri, Alhad Vinayak Sapre, Jayashree Ramanathan, Rajiv Ramnath, "Cost-Effective Supervised Learning Models for Software Effort Estimation in Agile Environments", IEEE 40th Annual Computer Software and Applications Conference, 2016.

[46] k.h.thanoon, "Morphological Properties for feature Extractionof Geometrical shapes, International Journal of Engineering and Innovative, 2017.

[47] Poonam Rijwani and Sonal Jain, "Enhanced Software Effort Estimation using Multi Layered Feed Forward Artificial Neural Network Technique", Procedia Computer Science, Volume 89, 2016, Pages 307-312.

[48] Amid Khatibi Bardsiri and Seyyed Mohsen Hashemi, "Machine learning methods with feature selection approach to estimate software services development effort", International Journal of Services Sciences, DOI: 10.1504/IJSSCI.2017.088034, January 2017.