

COMPARATIVE STUDY OF SOFTWARE ESTIMATION TECHNIQUES

Mohammed Aljohani and Rizwan Qureshi

Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, KSA

ABSTRACT

Many information technology firms among other organizations have been working on how to perform estimation of the resources such as fund and other resources during software development processes. Software development life cycles require lot of activities and skills to avoid risks and the best software estimation technique is supposed to be employed. Therefore, in this research, a comparative study was conducted, that consider the accuracy, usage, and suitability of existing methods. It will be suitable for the project managers and project consultants during the whole software project development process. In this project technique such as linear regression; both algorithmic and non-algorithmic are applied. Model, composite and regression techniques are used to derive COCOMO, COCOMO II, SLIM and linear multiple respectively. Moreover, expertise-based and linear-based rules are applied in non-algorithm methods. However, the technique needs some advancement to reduce the errors that are experienced during the software development process. Therefore, this paper in relation to software estimation techniques has proposed a model that can be helpful to the information technology firms, researchers and other firms that use information technology in the processes such as budgeting and decision-making processes.

Keywords

Software estimation techniques, project management, PERT, Risk management, Risk estimation

1. INTRODUCTION

Software engineering techniques, processes, and tools are used to manage the process of software development lifecycle. In addition failure of any method may cause negative influences such as effects on SDLC phases and project progress as well as the outcome of a software project. In order to complete the project properly and deliver it, to the customer as scheduled, the project manager must estimate the resources, effort and time needed as well as the cost of the software product. There are several estimation methods those are applicable to the development of object-oriented system [1]. In fact, these techniques can be used for software estimation, but basically, these techniques are divided into two categories i.e., algorithmic and non-algorithmic methods. Software project manager must estimate the resources, effort and time need as well as the cost, that in order to complete the project properly. On the other hand, the difficulty is finding the best and optimal techniques to be used before start software project in term of accuracy and usage as well as its suitability to the project.

Software estimation challenges include but are not limited to accuracy, budget, and schedule. Most of the existing techniques are used, to estimate time, effort and cost. In addition, these techniques are easy to understand and applied to the simple project, but it is failure led to not accurate and wrong estimate. Estimation techniques can be enhanced by using hybrid and combined two and more techniques in order to improve the accuracy.

In this paper, a comparative study among existing and traditional estimation techniques is addressed. Figure 1 shows the software estimation techniques, those used in this study. The purpose of this contribution is to facilitate and help project managers and researchers to choose the best techniques for estimating software projects. Nevertheless, the accuracy of existing estimation techniques is a big challenge. The method accuracy allows project managers to choose the best techniques to be used.

The paper is organized as, first an overview of estimation techniques, literature review, comparison criteria, results and proposed model.

2. RELATED WORK

The classification of software cost estimation models and techniques are reported and categories by Suri and Ranjan [1]. The most existing methods for software cost and effort estimation are estimation by Analogy, Expert Opinion, Putnam's Software Life-cycle Model (SLIM) and COCOMO. The comparative analysis of software estimation techniques and the aspects of each method was discussed as analogy estimation needs historical data which not available in all organization, secondly Delphi estimation method which is expert opinion for project specification is taken, thirdly SLIM Putnam software effort estimation mode finally basic COCOMO which is good at quick and early cost estimate for software with limitation of accuracy due to lack of accounting factors in project attribute (cost driven) which intermediate COCOMO account the detailed and influence of project phases. The main limitation in Suri and Ranjan [1] Paper that further work on learning method should be applied based on software module proprieties and criticalities. Also, the result of case study needs to be proved.

The generic model [2] used Principal Component Analysis (PCA) with Artificial Neural Network (ANN) base of (COCOMO II) model to improve the correctness of cost estimation. The relation to the cost of resources that are essential during software development, So the combination of these models aid to estimate software cost and effort by calculate the size and create a hybrid tool. The PCA and ANN are the two models that are more accurate; hence, they can be the solution to the cost estimation. Therefore, the result of PCA and ANN based on COCOMO are more accurate and increase the correctness of estimation without worsening the variability. Future work on Kernel Principal Component Analysis (KPCA) model displays a significantly different classification bias, as a result, confirms the accuracy is improved by the addition of KPCA model [2,22].

Algorithmic and Non-algorithmic methods, Function point size, COCOMO, and ANN were combined by [3] to help in getting better the correctness of software approximation techniques. The study combined the three techniques as a hybrid model and resulted that the accuracy has been improved compared with COCOMOII and ANN. The benefits of the proposed model are the time turnaround and accuracy. Therefore, the correctness of software estimation increased and obviously, the accuracy improved. The main limitation of the proposed model [3] is different classification presented but accuracy still moderates and less than if we are adding KPCA based model which will improve the accuracy better.

An Adaptive Learning Approach to Software Cost Estimation article by Reddy and Raju [4,15] proposes the use of back propagation neural networks as an approach to perform cost estimation for software. An artificial neural network is almost similar to the biological neural networks. Backpropagation neural network accommodates and improves the COCOMO techniques. The model consists of COCOMO dataset and the dataset for COCOMO NASA 2 that are used to train and test based on the simulation and prototypes performed. The blend, integration of back propagation neural networks more accurate than traditional COCOMO. Hence, the actual and the approximated cost can be so near to the real cost [4].

Saroha and Sahu [5] reviewed tools and methods of software effort estimation by using use case point model. The review discussed the estimation of the development processes of software by consideration of effort used in estimation as take more and consumes a lot of time. The review shows that the application of the UCP can be more effective and efficient as compared to other models such as COCOMO and FPA. Despite the difficulty is to use UCP values through companies because of no use case standard while the evaluation through Saroha and Sahu paper [5] applied to the small company.

Enhancing COCOMO estimation by using neural network proposed by [7]. The proposed solution seeks to improve efficiency by enhancing the prediction accuracy in the COCOMO model as a software effort estimation system applying the artificial neural networks. A multi-layer feed forward neural network used to sustain the model plus its related parameters. The proposed network is doused with a back propagation learning algorithm training. A COCOMO dataset is applied in training and resetting the network, as observed, it was noted that the suggested neural network model enhances the correctness of estimation of the COCOMO model. The main setback of the neural networks though is the over-reliance on the training set. Notwithstanding the minor shortfalls, neural networks remain the best technique so far. The neural network model showed relatively better results compared to the COCOMO model. Despite that the limitation of [4,7] there is proving of another estimation model, that is a generic model to improve the accuracy of cost estimation as a hybrid approach [3,7].

Failure in errors estimation at the early stages of software products can be costly regarding expenses and time discussed by Gupta et al. [8]. The proposed rule based test case reduction technique used decision table and it showed errors might arise at later stages of the development of software. Moreover, the errors in the initial stage of SDLC with less accuracy and efficiently attended too are likely to cost software development companies quite a consider penny and time loss. Forms are considered as a backbone of this study. Moreover, the proposed approach [8] does not demand whoever is performing the test to have prior knowledge of coding or logic programming. As a measure of its efficiency, the model's redundancy reduction stands at approximately 33% hence, significantly cutting down on expenses on cost as well as time.

The use of the regression analysis to forecast the cost was proposed by due to the application of the COCOMO in software effort estimation [9]. The regression analysis is the statistical approaches used to measure the correlation between variables. The proposal aims at using the regression analysis to perform the relative magnitude error that is essential in computing ultimate error percentage between the real and anticipated efforts. In the prototype, the COCOMO set of data for the estimation of software effort is utilized to assess the performance of the regression tools M5 and linear regression. The prototype uses the features of COCOMO dataset by [9]. The proposal demonstrated that the errors such as MMRE and MdmRE of the M5 algorithm is a less compared linear regression in forecasting by 80.20 and 45.30 percentage respectively and their anticipation is to reduce the forecasting errors, but more investigation still needed to reduce error forecasting.

Lilja et al. [10] discussed uses of Delphi method and aims to explore the usability of Delphi technique to assemble complex qualitative data from advisors and simple groups of qualified experts that will use as key materials for scientific research concerning software engineering. It is the data collected from the named sources that will form the primary source information and. Therefore, the data should be as valid and reliable as possible. The paper prefaces itself with the method and a discussion of its theoretical and philosophical background. Secondly, two variant research studies and introduced, both representing the traditional Delphi method and a modified Delphi research. The last part hosts a discussion on the reliability and validity of the data

collected for purposes of scientific deliberations given Delphi method. It goes further proceeds to enumerate the factors to be taken into account when one uses the Delphi method to assemble data. Much as Delphi method may be preferred in certain circumstances as the most suitable method of scientific data collection and analysis, it presents quite another of cons. Moreover, this method is mainly limited to research in particular disciplines such as health, social and medical disciplines. The validity and credibility of data collected for purposes of the study are, particularly of interest. The data has to be exhaustively screened to limit absurd conclusions as a result of using invalid and unreliable data.

Klaus Nielsen [11] proposed software estimation using a Combination of Techniques. The project managers should consider doing a thorough research before choosing the cost estimation model. Software cost estimation is supposed to be characterized by common sense and help from the friends with software cost estimation skills. The cost estimation challenges are required to be recognized to ensure that all the activities are performed swiftly. Considering the past and the future estimation is very crucial. In brief, the estimation is a proper application of knowledge, skills, basic techniques, and tools in the management of the project to meet all the requirements that required for project maturity, the mix methods and best project practice [11].

Balaji et al. [12] propose a cost of software approximation applying function point with the non-algorithmic method. They choose the approach to help the industries from the deficits problems. The approach involves the estimation of the final output of software. The activities that are done include effort estimation, scheduling the preliminaries of the project, and lastly overall estimation of the project cost. There is a sizing problem in the proposed the cost of software approximation using function point with the non-algorithmic approach by [12].

Software effort estimation with robust global techniques of linear regression was proposed by Lavazza and Morasca [13]. The use of linear quintile of squares of which is the generalization least median of squares (LMS). The purpose of the LMS is to minimize the mean square residual. The use of the LQS n effort estimation that is applied during software development on the four projects of the promise dataset that helps in obtaining the significant univariate approaches. LQS is likely to provide the valid option to the LMS, and the conventional least square regression in building estimation while balancing the need of eliminating outliers and maintaining sufficient data points to build the significant model statistically and to use lenient assumption basically for regression techniques as well as analyze more data will get benefits of drawing reliable result.

An impact of linear regression models for improving the software quality with estimated cost proposed by Marandi and Khan [14] as they considered the costs that most of the organizations are spending in evaluating the errors and bugs. They proposed the use of quantitative and qualitative management via the application of defect removal and effectiveness and statistical procedure of controlling using the historical data in the cost analysis. With good plan the quality of the project will improve continuously; hence it results in the cost-effectiveness of the project or software development. The key purpose of the proposal is to reduce the software products cost estimation and to improve the techniques for effectiveness and efficiency. Also, the proposed model is shorter in development schedules.

Karna and Gotovac [15] modeled expert effort estimation by developing collected data in a real environment using effort estimation methodology. The article shows the challenges that are faced by the experts when performing estimation. The model targets only the certain environments where the expert methods are being in developing the software products. The model can be able to produce the top predictors that will help In performing valid estimation. The predictors are used for identifying each variable input that is essential during prediction process. Despite the efforts that are exerted by the experts, they still face challenges that are related to the nature of producing the estimates, their triggers of estimation errors.

Pandey [16] analyzed the software cost estimation techniques. The article discussed the relation of estimation to the cost of developing software. However, the cost and expert efforts can be reached by employing models such as; Fuzzy logic and logarithms. Also, it's obvious that the two models are more accurate than other models. On the other hand, predicting the estimates for the software cost and the software effort is the most challenging activity in all software industries.

Table 1. Summary of related work and limitations

Paper title	The limitation
comparative analysis of software effort estimation techniques[1]	Learning method should be applied based on software module proprieties and criticalities. Also, the result of case study needs to be proved.
Generic model of software cost estimation: a hybrid approach [2]	Displays a significantly different classification bias, as a result, confirms the accuracy is improved by the addition of KPCA model.
a collective study of PCA and neural network [3]	Accuracy still moderates and less than if we adding KPCA based model which will improve the accuracy better
enhancement of prediction accuracy in COCOMO[4]	Another estimation model Generic Model to improve the accuracy of cost estimation as a hybrid approach.
Tools & Methods for Software Effort Estimation Using Use Case Points Model [5]	It is difficult to use UCP values through companies because of no use case standard during the evaluation.
COCOMO estimates using neural networks [7]	Another estimation model Generic Model to improve the accuracy of cost estimation as a hybrid approach.
Rule-based test case reduction technique using decision table[8]	The significantly is cutting down on expenses the cost as well as time.
Regression Techniques in software effort estimation using COCOMO dataset [9]	more investigation still needed to reduce error forecasting
Using Delphi Method [10]	Similar informed view much caution should be taken when using Delphi method for research.
Software estimation using a combination of techniques [11]	Learning method should be applied based on software module proprieties and criticalities. Also, the result of case study needs to be proved.
Software Cost Estimation using Function Point with Non-Algorithmic Approach [12]	There is a sizing problem in the proposed the cost of software approximation using function point with non-algorithmic approach
Software effort estimation with a generalized robust Linear Regression Technique [13]	Analyze more data will get benefits of drawing reliable result.
Impact of linear regression models for improving the software quality with estimated cost [14]	The model limits to shorter development schedules.
Modeling Expert Effort Estimation of Software Projects[15]	They still face challenges that are related to the nature of producing the estimates, their triggers of estimation errors.
Analysis of the techniques for software cost estimation [16]	The software cost and the software effort is the most challenging activity in all software industries.
Activity estimation using regression [17]	consumed computation time higher than MLR which make it some weakness
An expert estimator tool to estimate Cost and Risk with early stage of function points[18]	Expert estimation methods, still face challenges that are related to the nature of producing the estimates, their triggers of estimation errors.

Activity estimation using regression technique is proposed [17] to estimate human activities such as sitting standing and walking. The challenges in the quantification of the human activities are the problem that is discussed in the article. The MVR is the technique that can be used to estimate activities. The MVR evaluated by comparing it was with MLR regarding accuracy and time and resulted that MVR estimates better regarding accuracy. Table 1 displays summarization for all of these related works with limitations.

Jaiswal and sharma [18] proposed a tool to estimate cost and risk of software project. The tool is an estimator tool. The purpose of the proposed solution is estimate the cost, and risk of software projects. The paper combined function point and risk management process to calculate the cost and risk. The cost estimation based of function point as an input to the proposed tool, also adjust factors and reuse. The risk estimation was based on risk assessment processes. The proposed solution is simple and seems easy to be used for software project. While, the cost estimation that are exerted by the experts, still face challenges that are related to the nature of producing the estimates, their triggers of estimation errors. Also, further study is needed to improve the accuracy of estimation techniques [18].

3. SOFTWARE ESTIMATION CHALLENGES

Software estimation challenges include but are not limited to the accuracy, budget, and schedule. Most of the existing techniques used to estimate cost, effort, and duration. Also, these techniques are easy to understand and applied to software projects, but it is failure led to not accurate and wrong estimate. Estimation techniques can be enhanced by using hybrid and combined two and more techniques to improve the accuracy. Therefore, the accuracy of cost estimation techniques is a major challenge.

4. REVIEW OF SOFTWARE ESTIMATION TECHNIQUES

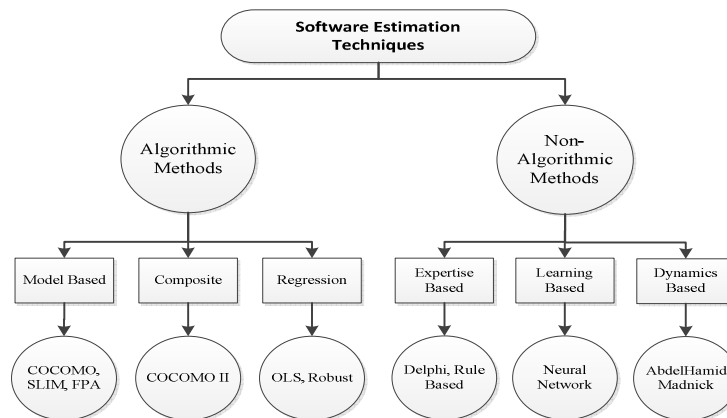


Figure 1. Software Estimation Techniques

There are many software estimation techniques, and it can be categorized into six categories [1] as appear in figure 1, in addition, it can be categorized into algorithmic and non- algorithmic method [2-3]. A conventional Algorithmic approach provides the mathematics and trial equations or formula to compute software cost and effort [3-4]. The formulas of this method are founded on inputs and past data such as scale factors, cost factors, and so on. In addition, some of the trendy algorithmic prototypes are COCOMO, FPA, COCOMOII, etc. Moreover, the non-algorithmic method applied by using previous data that are similar in requirements [3-4]. The estimation in this method performs examining the sequence of data and considering the person behavior

includes novel approaches that rely on soft calculation, such as equivalence, judgments of the expert, Delphi, etc.

The traditional and most estimation techniques those take up in this paper are as shown in Figure 1. Software estimations techniques, Algorithmic and Non- Algorithmic are applicable to the development of object-oriented systems. We can classify them by the following points [5].

4.1. MODEL-BASED ESTIMATION TECHNIQUES

Model-based software estimation techniques [1] are algorithms with mathematical models calibrating from the definite amount of known project data. Example of used methods is FPA, SLIM, and COCOMO which are essential in planning, tradeoffs analysis, monitoring and estimating resources and fund required for a software project.

4.1.1. FUNCTIONAL POINT ANALYSIS (FPA)

FBA is used for measuring complexity and size of software [19]. The idea of function pints is assessing the size of the system in functions form they perform, and the idea transfers to an object-oriented approach. Therefore, FPA focus on the use case and class diagram in order to estimate any software project, FPA can be applied by first counting number use cases connected to actors or extend in the first set that connected to an actor and secondly counting the number of classes with no exception. Moreover, FPA adjustments factors are used to get actual weight which normally ranges around 4 to 7 for use scenario and 7 to 15 for classes [26]. These valued depends on one’s opinion of the technicality of adjustments factors. Table 3 shows the equation and parameters used in this method. Function point can be converted into the effort by the first convert it to lines of code for particular language being used which is based on experiences [26]. Function point produces the same volume of code on average based on industrial experiences.

4.1.2. CONSTRUCTIVE COST MODEL (COCOMO)

COCOMO is one of the algorithmic approaches and it is introduced by. Boehm. COCOMO used to calculate schedule and size, effort and duration. COCOMO is made up of a hierarchy of three progressively in depth and exact forms; fundamental Model, transitional Model and in-depth Model [6]. Basic COCOMO estimates the effort for a software project that varies from small to medium which type in fast and unsystematic approach. Intermediate and Detailed COCOMO methods convert code size (KLOC) into the effort in person-month and optimal project duration in a month [7]. COCOMO acknowledges three software project classes. Table2 shows the parameter values for it which are based on analysis of many software projects.

Table 2. Parameters values for COCOMO

Software type	Parameters			
	a	b	c	d
Organic	2.4	1.05	2.5	0.38
Semi-detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

4.1.3. SOFTWARE LIFE-CYCLE MANAGEMENT (SLIM)

Putnam approach is considered as an experimental software effort testing model. It is seen by Lawrence H. Putnam in 1978 as revolutionary work in a section of Processes regarding software Modelling [1]. This approach of software estimation elaborates the effort and time that is supposed to be applied for a project of specified size. Software life-cycle management (SLIM) is closely related in software parametric to COCOMO model.

4.2. EXPERTISE BASED

The data that are quantified by the expert are used when there is no empirical. These approaches use the skills and skills of particular domain experts. The examples of this kind of estimation are the work breakdown, Delphi, and rule-based method. The major limits of this approach are that it is based on expert judgment; hence it may be partial depending on the expert interest.

4.2.1. DELPHI

Delphi is a Greece word donating to place [1]. This technique is a structured communication approach that was created as a systematic model of interactive forecasting that depends on the decisions from the experts' panel.

4.2.2. RULE-BASED

In the rule-based method, each rule can be taken as an independent unit with the same format for all but need careful treatment to avoid infinitive loop possibility of conflict between rules.

4.3. LEARNING BASED

These forms of methods use past data and the existing knowledge to create a model for estimating the effort of software development [20]. Neural network approximation and analogy are some of the examples for this categorization [23]. The previous method is based on learning phase, training and the future one is based on the estimation of an expert and comparison with the outcome obtained from the past data of the same software project [21]. Those types of methods use past data and existing knowledge to create representative for the effort of software.

4.3.1. NEURAL NETWORK

An essential neural network is made up of some inputs added by some weights, integrated to provide an output. The results from the output are added back to inputs to fix the applied weights and for network training purpose. This shape of the neural networks is useful in solving non-linear, practical and making decisions easily.

4.4. REGRESSION TECHNIQUES

Regression-based models are popular for developing cost approaches. Regression techniques are easily and widely accepted approaches and are utilized by some other methods such as COCOMO II and SLIM checkpoint. The regression-based methods used together with model-based techniques. Ordinary Least Squares and robust regression methods are under this categorization. However, it needs big data, and there should be no data items missing.

4.4.1. ORDINARY LEAST SQUARES (OLS)

Ordinary least squares is an approach, used to estimate parameters that are uncertain in a linear regression model, OLS aims to minimize the differences between the experimented value feedback and predicted value in some of the big dataset by using linear approximation of the data (addition of the vertical distances between each point in the set and the related point on the regression line). So the estimator can be interpreted by a simple formula.

4.4.2. ROBUST

Robust regression is a form of regression analysis that purposed to overcome some disadvantages of conventional non-parametric and parametric techniques. Regression analysis requires catching the relationship among one or more than dependent variables and an independent variable. Most

of the utilized approaches of regression, such as ordinary least squares, has features, if its assumptions are true, however, can provide wrong results if the assumptions used are wrong; Robust regression methods are predetermined to be not affected by violations of the assumptions. Approaches for approximating the uncertain parameters in a linear regression representative with the aim of proper utilization the variance between the experimented feedback is some of the arbitrary set of data and the feedback forecasted by the linear estimation of the data (visually this is observed as the addition of the vertical distances between each data point in the set and the relating point on the regression line - the smaller the variance, the better the model fits the data). The estimator can be addressed by a simple formula, typically in the scenario of the right-hand side or a single regression.

4.5. COMPOSITE

Composite based approaches are the integration of multi techniques (two or more) that formulate the best functional type for estimation.

4.5.1. COCOMO II

COCOMO II is an approach give capacity [1,6] for one to approximate the cost, schedule, and effort that is required when planning the development of new software. The extension of the original COCOMO was published in 1981. It made up of three sub-models each sub-model is providing fidelity during the planning processes and configuring processes. The sub-models are called application comparison, early design and post-architecture model in the listed fidelity.

Table 3. Software Estimation Techniques Formula

Method	Formula
FPA	$FP = Fu \times Wu + Fc \times Wc$, Where; Fu: number of use cases; Wu: weighting of use cases; Fc: number of classes with no exception and Wc: weighting of classes. $Adjusted = FP \times (0.65 + 0.01 \times \sum F_j)$ Where; FP: function point, $\sum F_j$: sum of fourteen factors, each value range (0-5)
COCOMO	$E = a(KLOC)^b$ Where; E: effort, a, b: come from table of COCOMO parameter KLOC:(FPA * average of LOC per F) 1000 $D = cE^d$ Where; D: duration, c, d: come from table of COCOMO parameter E:effort calculated above
SLIM	$S = E \times (Effort)^{\frac{1}{3}} t_d^{\frac{4}{3}}$ Where; td is software delivery time, E environment factor. $Effort = D_0 \times t_d^3$ Where; D0:manpower build-up parameter,
COCOMO II	$PM = 2.94 \prod_{i=1}^{17} EM_i S^{0.91+0.01 \sum_{i=1}^5 w_i}$, $E = B + 0.01 \times \sum_{j=1}^5 SF_j$ Where PM: is effort expressed in person per month, S is the effort expressed in person size of the software project. EMi: effort multiplier, wi : expand scale factor. E: Scaling base-exponent, SF: 5 Scale Factors, B: Scaling base-exponent.
OLS	$R_{ols}^2 = 1 - \frac{\sum_{i \in 1 \dots n} (y_i - ax_i - b)^2 / n}{\sum_{i \in 1 \dots n} (y_i - \bar{y}_{ols})^2 / n}$ Where R_{ols}^2 is R ² usual sued in OLS $\sum_{i \in 1 \dots n} (y_i - ax_i - b)^2 / n$: is the mean of squared residual. \bar{y}_{ols} : is the estimator of central tendency for the multiset.
Robust	$R_{LMS}^2 = 1 - \frac{med\{ y_i - ax_i - b \}}{med\{ y_i - \bar{y}_{LMS}\}}$, $R_{LQS}^2(q) = 1 - \frac{v(q)\{ y_i - ax_i - b \}}{v(q)\{ y_i - \bar{y}_{LQS}(q)\}}$

4.6.DYNAMICS BASED

Dynamic based is a technique to estimate changes in effort during development of a system [25]. The model has been built to predict changes in the cost, staffing team needs, schedules and time of project as far as the proper initial values of the project are available to estimate. Madnick applied the technique in the context of the system reuse, demonstrating in the interesting result of getting the benefit of the relationship between reuse software components and project personal productivity. The system has dynamic continuous simulation modeling methodology [24]. Therefore, the result and behavior display as a graph or information that change over time. The model is difficult to calibrate and give results as it appears as a graph, so it is difficult to measure and get the accuracy level. Therefore, we exclude this technique from our comparison study.

Software estimation techniques differ regarding the formula used, and characteristics, summaries of each technique appear in Table 3.

5. COMPARISON CRITERIA

To evaluate estimation techniques, compare the performance and proposed the optimal techniques be used based on software project situation. We set three criteria to see the accuracy, suitability, and usage of these techniques.

5.1.ACCURACY CRITERIA

We have studied the accuracy of each estimation technique based on a literature review that taken up in this paper. That to get the accuracy level of each estimation parameters, the most used are measure the average of estimation accuracy of the Mean scale of Relative Error (MRE), where the MRE of each estimate defined as [15]. Also, comparison result among these techniques described in the following comparison section

$$MRE = \frac{|Actual\ Effort - Estimated\ Effort|}{Actual\ Effort} \times 100$$

Table 4. General Comparison of Algorithmic and Non-Algorithmic Models

Type	Based on	Technique	General Advantage	General Disadvantage
Algorithmic	Model-based	COCOMO, Slim, FPA	<ul style="list-style-type: none"> • Able to repeatable estimations. • Easily able to advance input data, improve and convert formulas. • Efficiently support sensitivity analysis. • Evaluated based on previous experience. 	<ul style="list-style-type: none"> • Resulted in some inaccurate estimation due to poor input sizing and inaccurate cost. • Not capable of dealing with exceptional circumstances. • Such expert judgment cannot uneasily quantify.
	Composite	COCOMO II		
	Regression Tech.	OLS, Robust		
Non-Algorithmic	Expertise	Delphi, Rule-based	<ul style="list-style-type: none"> • Useful when historical data absence • Useful in lacking project scope. • Can get a clear and quick result. • Simply it can be predicted 	<ul style="list-style-type: none"> • expert team should agree on agreements otherwise it useless • Time consuming as many participants. • Expensive method.

6. COMPARISON STUDY

Software estimation techniques used to estimate the effort and cost of a software project as well as software project managers must estimate the resources, effort and time need also the cost to complete and deliver the project properly. However, the project manager has to choose a suitable technique and apply it to his project to manage and deliver the project as scheduled.

6.1. ADVANTAGE AND DISADVANTAGES OF EACH TECHNIQUE

Table 4 shows the characteristics and drawbacks of estimation techniques taken up in this paper. Moreover, it also shows general advantage and disadvantage of algorithmic and Non-algorithmic methods. Table 5 below displays summarization advantages and disadvantages for algorithmic.

Table 5. Advantage and disadvantages of algorithmic techniques.

Method	Advantages & disadvantages	
COCOMO	Advantage	<ul style="list-style-type: none"> • Transparent technique • Drivers are very useful for estimating the project cost.
	Dis-Advantage.	<ul style="list-style-type: none"> • Not useful to measure the size. • Demand for historical data which not always available.
SLIM	Advantage	<ul style="list-style-type: none"> • Using linear programming, can developing constraints. • No need many parameters.
	Dis-Advantage.	<ul style="list-style-type: none"> • So sensitive • Inconvenient with small projects
FPA	Advantage	<ul style="list-style-type: none"> • Reliability about effort • Measuring projects in different languages ex. java, C# etc.
	Dis-Advantage.	<ul style="list-style-type: none"> • Lack of research dataset on FPA. • Can't be performed before creating the design.
COCOMOII	Advantage	<ul style="list-style-type: none"> • Provide replicable and objective estimation • Suitable, produce more accuracy.
	Dis-Advantage.	<ul style="list-style-type: none"> • Still, use waterfall process. • Not suitable for effort in all SDLC phases.
Robust	Advantage	<ul style="list-style-type: none"> • Great effect on quality. • Not affected by violations of the assumptions.
	Dis-Advantage.	<ul style="list-style-type: none"> • Consume more time, due to a lot of things to be considered. • Required promise data history.
OLS	Advantage	<ul style="list-style-type: none"> • Very efficient in using and analyze the dataset.
	Dis-Advantage.	<ul style="list-style-type: none"> • Sensitivity to the data that come out the range of data set. • Required data history.

Table 6 displays summarization advantages and disadvantages of Non-algorithmic estimation methods.

Table 6. Advantage and disadvantages of Nonalgorithmic techniques.

Method	Advantages & disadvantages	
Delphi	Advantage	<ul style="list-style-type: none"> • No need for historical data
	Disadvantage	<ul style="list-style-type: none"> • Need agreement with all team members before applying it. • Consume more time.
Rule-based	Advantage	<ul style="list-style-type: none"> • Each rule is taken as an independent unit. • Same format for all knowledge.
	Disadvantage	<ul style="list-style-type: none"> • Need treatment carefully to avoid the infinitive loop. • Possibility, conflict among rules

6.2. COMPARE THE ACCURACY OF ESTIMATION TECHNIQUES.

Table 7 shows the accuracy, Usage, and suitability of estimation techniques.

Table 7. Accuracy, Usage, and Suitability of Estimation Techniques.

Method	Accuracy	Usage	Suitable for
COCOMO	Good	very useful for project cost	Estimate effort and time
SLIM	Good	useful in big projects	Estimate effort and time
FPA	Low	specify complexity of the project	Estimate size and effort
COCOMOII	Good	used to produce more accuracy	Estimate effort, size, cost, and schedule.
Robust	Good	used for valid, significant univariant	Estimate effort on four projects from the dataset.
OLS	Low	used only for relatively small project	Estimate effort from dataset
Delphi	Based on experts experience	used when lacking in project scope	Estimate effort, size, cost, and schedule.
Rule-based	Based on team experience	used to cognitive estimate based on the rule	Estimate effort, size, cost, and schedule.

7. FUTURE RESEARCH OPPORTUNITIES AND PROPOSED MODEL

The study finds that software estimation techniques are easy to understand and applied to the simple project, but it is failure led to not accurate and wrong estimate. Estimation techniques can be enhanced by using hybrid and combined two or more techniques to improve the accuracy. In order to come up with that comparison among existing hybrid model techniques based on certain criteria will improve the accuracy as it proved by many research that it gives more accurate result than using the single technique. Where comparing and evaluating the results of existing

techniques can pick and predict differences in accuracy of estimation techniques. Therefore, a hybrid model is proposed integrating existing estimation techniques to enhance the accuracy of software estimation. The proposed model is shown in Figure 2. The proposed model will enhance the accuracy of existing estimations techniques. The combination process of the proposed model is based on project evaluation and review technique (PERT) [27]. These values are as a distribution lies between the minimum (optimistic), maximum (pessimistic), and nominal (most likely) values which in between the minimum and maximum values [27-28]. PERT equation is as follows.

$$PERT = \frac{(\text{Optimistic} + 4(\text{Most Likely}) + \text{Pessimistic})}{6}$$

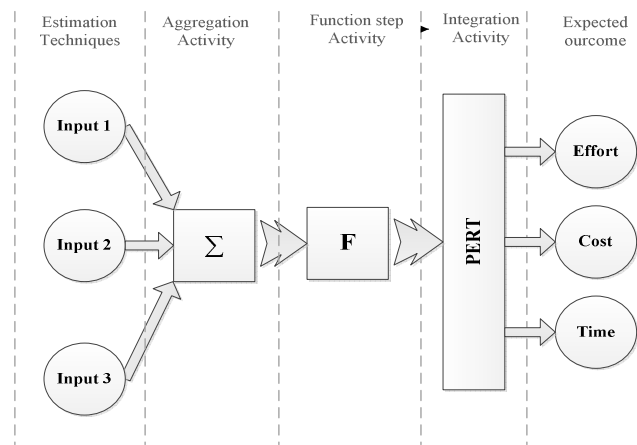


Figure 2. Proposed Software Estimation Model

As per the proposed model, the researcher should choose three of existing estimation techniques. These techniques are considered as input1, input2, and input3. Second, aggregate the selected techniques, for example, time, effort and cost of each selected technique. That means to aggregate the estimated time, effort and cost for input1, input2 and input3 after apply each estimation technique separately. A researcher puts inputs in a different block for each category (for example; Time for input 1, 2 and 3 in one block and so on), to prepare it as input for functional activity in next step. Table 8 shows the aggregation activity of estimation techniques input.

Table 8. Aggregation Activity of Estimation Techniques

Inputs	Block 1	Block 2	Block 3
Input 1	Est. time	Est. effort	Est. cost
Input 2	Est. time	Est. effort	Est. cost
Input 3	Est. time	Est. effort	Est. cost

Third, a researcher should apply a function to each aggregated block separately. This function will specify the biggest, lowest and nominal values. Fourth, a researcher is able to integrate the output of function values by using PERT method. Fifth, a researcher applies PERT method to get the estimated time, effort and cost of composed techniques.

8. CONCLUSION AND FUTURE WORK

The project managers always faced problem at the beginning of a project to select a suitable cost estimation technique to calculate the cost. This is because there are several cost estimation techniques. This research studied the existing techniques in term of accuracy, usage, and suitability. Therefore, our comparison can facilitate and help the project managers to distinguish and compare among techniques to choose the optimum technique according to project type and requirements. Also, we have proposed a model to help a researchers, and project managers, by combining three existing estimation techniques to improve the accuracy. The challenges that are being anticipated, and covered by using the proposed model include the errors that can result due to the single approach failure. The wrong estimation is another challenge that can be faced when the proposed model is employed in future researches. Hence, improvement, performance, and efficiency of the proposed model will be evaluated in future.

REFERENCES

- [1] Suri, Ranjan, (2012) “Comparative Analysis of Software Effort Estimation Techniques”. International Journal of Computer Applications (0975 – 8887) Volume 48– No.21.
- [2] Patil, Waghmode, Joshi & Khanna, (2014) “Generic Model of Software Cost Estimation: A Hybrid Approach”. IEEE International Advance Computing Conference (IACC).
- [3] Waghmode, Patil & Joshi (2013). “A Collective Study of PCA and Neural Network based on COCOMO for Software Cost Estimation”. IJCA (0975 – 8887) Volume 74–No.16.
- [4] Madheswaran, M., & Sivakumar, D. (2014). “Enhancement of prediction accuracy in COCOMO model for software project using a neural network”. In Computing, Communication and Networking Technologies (ICCCNT), International Conference on (pp. 1-5). IEEE.
- [5] Saroha, Sahu, (2015). “Tools & Methods for Software Effort Estimation Using Use Case Points Model – A Review”, IEEE, ISBN:978-1-4799-8890-7/15.
- [6] Musilek, Pedrycz & Sun, (2002), “On the Sensitivity of COCOMO II Software Cost Estimation Model”. Proceedings of the Eighth IEEE Symposium on Software Metrics (METRICS.02) 0-7695-1339-5/02.
- [7] Kuashik, Chauhan, Mittal, Gupta,(2012) .“COCOMO Estimates Using Neural Networks”, MECS.
- [8] Gupta, A., Mishra, N., & Kushwaha, D. S. (2014). “Rule based test case reduction technique using decision table”. In Advance Computing Conference (IACC), IEEE International, pp. 1398-1405.
- [9] V. Anandhi, R. Manicka Chezian, (2014), “REGRESSION TECHNIQUES IN SOFTWARE EFFORT ESTIMATION USING COCOMO DATASET”. 978-1-4799-3966-4/14 \$31.00 © 2014 IEEE DOI10.1109/ICICA.2014.79.
- [10] Lilja, Laakso & Palomäki (2011). “Using the Delphi Method”, IEEE, 978-1-890843-23-6/11.
- [11] Klaus Nielsen, (2013), “Software Estimation using a Combination of Techniques”, PMI Virtual Library, Project management institute.
- [12] Dr. N. Balaji, N. Shivakumar & V. Vignaraj Ananth (2013). “Software Cost Estimation using Function Point with Non-Algorithmic Approach”, Volume 13 Issue 8 Version 1.0.
- [13] Luigi Lavazza and Sandro Morasca, (2012), “Software Effort Estimation with a Generalized Robust Linear Regression Technique”, IET, ISBN: 978-1-84919-541-6.
- [14] Arun Kumar Marandi and Danish Ali Khan (2015), “An Impact of Linear Regression Models for Improving the Software Quality with Estimated Cost”, Procedia Computer Science Volume 54, Pages 335-342.
- [15] Karna, Gotovac, (2010), “Modeling Expert Effort Estimation of Software Projects”, IEEE, ISBN: 978-9-5329-0052-.
- [16] Poonam Pandey, (2012), “Analysis Of the Techniques for Software Cost Estimation”, IEEE, 978-0-7695-4941-5/12.
- [17] Bhuvanewari, Gayathri, & Priyadharshini, (2014),“Activity estimation using regression”, inter. conference intelligence and communication networks, CICN, IEEE, 978-1-4799-6929-6/14.
- [18] Ajay Jaiswal, & Meena Sharma, (2012), “An Expert Estimator Tool to Estimate Project Cost and Risk with early stage of function points”, International Journal of Software Engineering & Applications (IJSEA), Vol.3, No.5.

- [19] Archana Srivastava, Dr. Syed Qamar Abbas, Dr.S.K.Singh, (2012), "ENHANCEMENT IN FUNCTION POINT ANALYSIS", International Journal of Software Engineering & Applications (IJSEA), Vol.3, No.6.
- [20] Satyananda Reddy and KVSN Raju, (2012), "An Adaptive Learning Approach to Software Cost", NCCCS, IEEE, 978-1-4673-1953-9/12.
- [21] Srinivasan, K., & Fisher, D. (1995), "Machine learning approaches to estimating software development effort", Software Engineering, IEEE Transactions on, 21(2), 126-137.
- [22] Algabri, Saeed, Mathkour, & Tagoug, (2015), "Optimization of Soft Cost Estimation using Genetic Algorithm for NASA Software Projects", IEEE, 978-1-4799-7626-3/15.
- [23] Kaushik, Soni & Soni, (2012), "An Adaptive Learning Approach to Software Cost Estimation", , , IEEE, 978-1-4673-1953-9/12.
- [24] Tarek Abdel-Hamid, (1991), Stuart Madnick Book, Published by Prentice Hall, ISBN 10: 0138220409 / ISBN 13: 9780138220402.
- [25] Dnoald J. Reifer, (2006) Software Management 7th, ISBN-13978-0-471-77562-1, copyright ©2006 IEEE computer society.
- [26] Mohammad D. Aljohani, & M. Rizwan J. Quresh (2016), "Management of Changes in Software Requirements during Development Phases", International Journal of Education and Management Engineering (IJEME), Vol.6, No.6, pp.12-26.
- [27] Mete MAZLUMa, & Ali Fuat GÜNERİ, (2015), "CPM, PERT and Project Management With Fuzzy Logic Technique and Implementation On A Business", Procedia - Social and Behavioral Sciences 210 348 – 357.
- [28] Dr. P.K.Suri, & Pallavi Ranjan, (2012), "Estimating the Probability of Project Completion by SIM_DEL Estimator", IJCSIT, Vol. 3 (4), 4938-4945.

AUTHORS

Mohammed D. Aljohani is currently doing master in the department of IT, at King Abdulaziz University, Jeddah, Saudi Arabia. Aljohani obtained his B.Sc. Degree in Information Technology and computing from Arab Open University, at Saudi Arabia in the year of 2012. His interest areas are software engineering, requirements engineering, agile software engineering, Access control systems, cybersecurity, and computer networks.



Dr. M. Rizwan Jameel Qureshi received his Ph.D. degree in Computer Science from National College of Business Administration & Economics, Pakistan 2009. He is currently working as an Associate Professor in the Department of IT, King Abdulaziz University, Jeddah, Saudi Arabia. This author is the best researcher awardees from the Department of Information Technology, King Abdulaziz University in 2013 and 2016. He is also honored as the best researcher in Computer Science discipline from seven campuses of COMSATS Institute of Information Technology, Pakistan in 2008.

