SOFTWARE COST ESTIMATION IN GLOBAL SOFTWARE DEVELOPMENT USING HYBRID APPROACH

Abdullah Gani, University Malaysia Sabah
Adnan Akhunzada, University Malaysia Sabah
Muhammad Junaid, The University of Haripur

ABSTRACT

GSD (Global software development) has surged in current years. Numerous software ventures currently use asynchronous association amongst physically dispersed crews in different time regions. Just like to aspects such as exertion consumed in team-structure and awareness handover in creating a software invention construction that can be effortlessly dispersed and reduces cross-tie communiqué, facilitating communiqué among distant teams cooperating on consistent parts of the structural design, and their day-to-day operations, estimating software costs for such projects becomes difficult. We describe the additional cost drivers of distributed development in this article and assess how essential each of these variables is to the global cost of the software expansion plan. We propose various enhancements to COCOMO-II, the utmost frequently in use software expansion cost estimating exemplary, to accommodate for these new challenges.

Keywords: Software, Cost Estimation, Hybrid Approach

INTRODUCTION

The Expanding worldwide software marketplace, a tendency toward built-up software in less-expensive countries, and the growing intricacy and scale of software program structures, the proportion of developments that are worldwide dispersed has constantly increased (Kodmelwar, 2019). “Siemen-Corporate-Research”, Inc. (S.C.R.) has been researching to acquire a better knowledge of the issues and impact of various GSD practices since GSD raises the necessities for growth processes, project-management techniques, construction, excellence, associated tools, and so on (GSD).

According to a 1999 Standish Group report (Gharehchopogh, 2014), project or team size has a substantial link with project performance. We believe that one of the primary factors of this association is physical team separation, which occurs more frequently as projects grow larger. This physical separation necessitates extra activities and exertion, such as team edifice, knowledge transmission for the asynchronous association, developing a disseminated architecture that reduces cross-site Communiqué and enabling Communiqué between teams

1532-5806-25-S4-16

Citation Information: Gani, A., Akhunzada, A., & Junaid, M. (2022). Software cost estimation in global software development using hybrid approach. *Journal of Management Information and Decision Sciences, 25(S4),* 1-25.

working on consistent parts of the architecture (Sagar et al., 2018; Boehm, 1984; Chirra et al., 2019). This extra effort results in momentous planning, synchronization, and control above your head in the day-to-day governance of GSD activities.

Most organizations, however, are collaborating with software development firms in "Eastern Europe, South America, and Asia" to save money due to these companies' reduced labor expenses. However, given the other factors that influence offshore sourcing, this could be misleading. We describe an approach for enhancing the exactness of C.O.C.O.M.O. cost estimates for GSD in this work. Then we suggest some improvements to COCOMO II. Finally, we offer some findings, noting both shortcomings in our approach and potential research avenues.

Following a brief overview of COCOMO II, an extensively used framework for estimating software development costs, we analyze the reasons for challenges in distributed software development projects and how successfully C.O.C.O.M.O. represents them. Our study's findings can be utilized to evaluate trade-offs when selecting whether to consolidate or distribute software development. The overarching purpose of our research on these themes is to give managers with decision-making frameworks when confronted with such decisions."

Background and Purpose of the Research

Globalization has aided the growth of international software initiatives. However, different studies forecast that the number of offshore developments will rise over time (Khan et al., 2021). Initiatives for global software development (GSD) in nations such as India and China are expected to rise by 20 to 30 percent. Several software companies in the West have moved their operations to Asia and Eastern Europe because of lower labor costs in these states (Chirra et al., 2019). Firms are using GSD for a variety of reasons, including improved speed-to-market due to time zone differences and the employment of highly experienced virtual teams (Sagar et al., 2018).

Cost estimates are one of the most challenging aspects of project management for managers in a different project. It becomes considerably more crucial in the case of GSD because task and forecast resource allocation is much more difficult due to their dispersion (Kodmelwar et al., 2019). Additional cost drivers are not taken into account in the current state of the art to raise the projected accuracy for GSD (Rajeswari, 2018). The purpose of this article is to classify the cost drivers that influence GSD's cost approximation process in this regard. The evaluation of these cost drivers can help practitioners deal with the cost drivers and improve GSD's cost approximation technique (Suliman et al., 2017).

According to Suliman & Kadoda in 2017, the core objective of globalization, and the main rationale for not expanding domestically, is to reduce development costs, which makes it critical to consider time zone and cultural differences as development barriers. As a result, if GSD factors aren't taken into account, more time and effort may be necessary [10]. One of the most pressing challenges is determining whether GSD would benefit or gain importance if it were produced locally (Venkataiah et al., 2019).

Cost estimating can be done using a variety of tools and approaches. Several models were established previous to the GSD idea, and in essence, the models and approaches absent the features and cost-drivers connected with this expansion, leading to ambiguity about the GSD concept's application (Tsung et al., 2018). Various estimating methodologies are used to generate estimation models (Khan et al., 2021). The cost estimating models given here could aid practitioners in improving estimations because they have been expanded to fit the needs of GSD. Cost overhead, S.O.C.E.M., and Analogy are among the models (Ahmad et al., 2020).

1532-5806-25-S4-16

Citation Information: Gani, A., Akhunzada, A., & Junaid, M. (2022). Software cost estimation in global software development using hybrid approach. *Journal of Management Information and Decision Sciences, 25(S4),* 1-25.

However, regardless of which model is utilized, if the cost factors that affect the total cost of GSD are not taken into account, we will not obtain the desired outcomes. GSD believes that cost projections continue to receive insufficient attention. It's also worth mentioning the importance of discovering factors that influence GSD estimation and cross-checking those factors using existing increase methodologies (Wickramaarachchi et al., 2017). When a project is done correctly, the cost is only one of the driving forces (Khan et al., 2021), and when software development is accurately appraised, efficient expenditures are made.

Software development success is predicated on effective resource management and planning, which is often based on cost and time estimation, which many software engineers have found difficult (El Bajta et al., 2020). In the context of Global-Software Development (Jain et al., 2018), this underscores the possible necessity for reliable cost assessment methods. Many studies have focused on various methods for addressing the cost estimation problem that plagues the engineering industry, but these methods and cost estimation techniques fail to take into account the additional cost drivers that are frequently used to calculate accurate estimated costs in the context of Global Software Development (Ionescu, 2017; Sagar et al., 2018). The current study only gives a rough idea of the cost factors that can affect the GSD project. The purpose of this study is to find and optimize GSD-based Cost Drivers that will assist cost estimating models improve. To answer the challenge of Software Cost Estimation, Boehm employed models like COCOMO II. In this research, a hybrid strategy is employed to answer the problem of Software Cost Estimation by designing and implementing COCOMO II and a Machine Learning Approach. To answer the research issues, and S.L.R. and empirical methods will be used.

In this research, we will find out the S.C.E. through the GSD-based hybrid model to check that how cost estimation works in different stages and through different techniques (Ionescu, 2017).

This research is based on a study to determine which technique is best for estimating software costs and what problems exist with various techniques. Also, identify the elements that contribute to the failure of cost estimation strategies. Why aren't they able to come up with an accurate cost estimate using these methods? Our goal is to identify GSD cost drivers that can influence the estimating process.

This paper addresses the paucity of information and research on the cost drivers that may affect a GSD project, preventing project managers from accurately estimating the cost of any software project. The goal and goal of this paper are to find the best approaches to estimate proper software project costs and cut expenditures on software projects during development by adopting a hybrid approach to find a solution to software cost estimates in the context of global software development. Using COCOMO II, we will also investigate various methods for determining the cost of the software. We'll also use the COCOMO II model to plan the design and implementation phase of a software cost estimation.

Research Questions

RQ1: Which cost-approximation technique is best to find the actual cost of a software project?
RQ2: What factors do current cost estimation techniques fail to incorporate?
RQ3: What are the driver of the GSD cost that could also influence its Estimate?
RQ4: How can the cos- drivers that are defined be categorized?

1532-5806-25-S4-16

Citation Information: Gani, A., Akhunzada, A., & Junaid, M. (2022). Software cost estimation in global software development using hybrid approach. *Journal of Management Information and Decision Sciences, 25(S4),* 1-25.

LITERATURE REVIEW

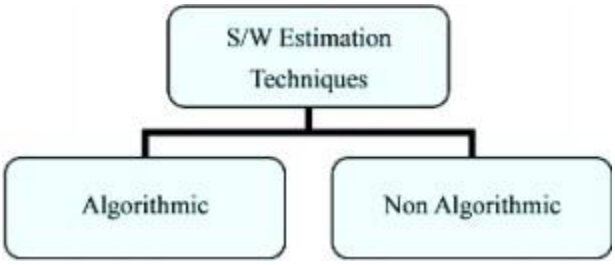

FIGURE 1
TYPES OF SOFTWARE COST ESTIMATION TECHNIQUES

The methodologies of cost estimate are mostly of two types: algorithmic and not algorithmic. To determine software cost estimates, algorithmic techniques use a formula. The formula is constructed using models that combine cost factors. In addition, for model construction, the statistical technique is applied. Non-algorithmic techniques do not employ a software cost estimate formula.

Many studies recommended several software cost estimation models. Numerous models have been projected and developed to uncover alternatives, improve or backing existing models. One of the first methods that have been developed by researchers for S.C.E. is the algorithm-based method (Hasanluo et al., 2016). Such methods were brought onto the scene in late 1970 among which include, COCOMO II a pattern-based model, Checkpoint model by Jones in 1997, SLIM by Putnam and Myers in 1992, PRICE-S model by Park in 1989 (Hasanluo et al., 2016), but these techniques fail to consider various factors that affect cost estimation such as time and effort (Khan et al., 2021)**.** Ascertaining the number of resources for software development for ages has been looked into by numerous researchers, and as such, there has seen an introduction of non-algorithm techniques such as machine learning algorithms in the software engineering field by researchers in an attempt to mitigate this problem by using machine learning algorithms (Hasanluo et al., 2016). Nevertheless, numerous methods have been presented to solve this emerging problem.

According to Low and Jeffery in 1990 (Low et al., 1990), the purpose point counts seem to be a more compatible and more occurring prior amount of software size than foundation lines of code. The discernment in the application of function points is a significant aspect of their successful application. Mukhopadhyay and Kekre in 1992 have established that the most important of the previous work in software effort assessment has put colossal importance on L.O.C. or other design features as primary variables for cost models. The appraised size be able to use in combination with the existing models to generate early appraisals of development effort.

The hybrid COCOMO II model with fuzzy logic was employed by researchers like Gharehchopogh et al. [21] to overcome the problem of S.C.E. The above work shows that the fuzzy systems are integrated with triangular, trapezoidal, Gaussian and widespread Bell intermediate COCOMO II functions. For these model's assessment criteria are employed, for the evaluation of hybrid approaches, such as M.M.R.E., M.R.E., PRED, MARE, V.A.F., VARE, B.R.E. In comparison to the COCOMO II model, results generated from the hybrid COCOMO-II model have shown greater performance.

1532-5806-25-S4-16

Citation Information: Gani, A., Akhunzada, A., & Junaid, M. (2022). Software cost estimation in global software development using hybrid approach. *Journal of Management Information and Decision Sciences, 25(S4),* 1-25.

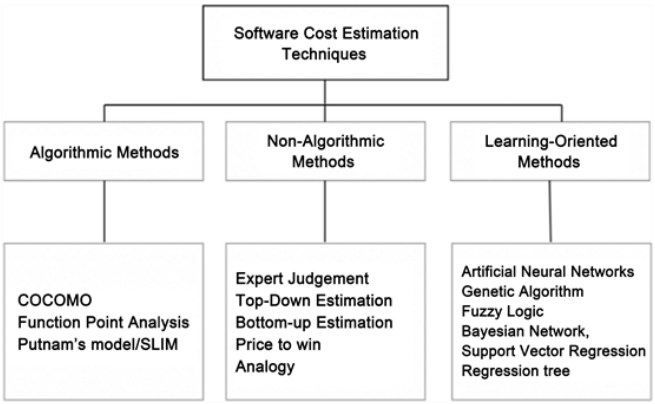

FIGURE 2
FURTHER TYPES OF SOFTWARE COST ESTIMATION TECHNIQUES

In solving the Software Cost Estimation problem, Ziauddin et al. (2013) used the fuzzy logic model. In this, the COCOMO II model parameters were first transformed to fuzzy numbers, and the numbers were emitted from the fuzzy state. Comparison with the COCOMO II and AlaaSheta models was achieved with the proposed method. In the proposed strategy, the comparison indicates an improved performance. In addition to the COCOMO II and AlaaSheta, the proposed procedure had a more marginal error of M.M.R.E., PRED (N), and V.A.F. [22]. In a research study by Reddy et al. in 2011 (Reddy et al., 2011), several PSOs were utilized to optimize the COCOMO II model parameters. In small and big initiatives, the proposed model was tested independently. Following the conclusions, MARE amounts are 16,1306 percent and 9,0143 percent correspondingly for micro-projects in the COCOMO II model and the suggested model. Furthermore, the MARE figures for COCOMO II and the suggested model were 18.1548 and 20.9717%, respectively. The results revealed that concerning the COCOMO II models, the model was higher performing.

Cost Drivers

These are the types of cost drivers which can affect the S.C.E. Structural cost drivers depend on the economic condition of the company. It also depends on the organizational scale and scope, which can affect the S.C.E. (El Bajta et al., 2020). Executional or operational cost drivers are the factors that influence a company's cost position and are dependent on its ability to successfully "execute" its operations or activities. We may evaluate three key structural cost factors using data from the I.M.V.P. survey: automation, plant scale, and product mix complexity (Jain et al., 2018). The number of resources consumed by an activity is measured by the resource cost driver. It's used to assign a resource's cost to an activity or cost pool. The frequency and intensity of demand imposed on activities by cost objects are measured by the Activity Cost Driver. It's used to assign costs to cost objects based on activities. The labor, maintenance, and other variable costs are influenced by the activity cost driver. A.B.C., a branch of managerial accounting that allocates an activity's indirect costs, or overheads, requires cost drivers (Ziauddin et al., 2013). These all cost drivers can affect the S.C.E., due to which the cost may be increased or decreased. But according to the GSD researchers trying to find the method of cost-cutting and best for the organizations.

Citation Information: Gani, A., Akhunzada, A., & Junaid, M. (2022). Software cost estimation in global software development using hybrid approach. *Journal of Management Information and Decision Sciences, 25(S4),* 1-25.

Review of Traditional Software Cost Estimation Methods

One of the most difficult aspects of software project management is estimating how much money should be spent, and this amount should be monitored throughout the project's life cycle and iterated at particular milestones. In some circumstances, we may obtain real data from prior projects and have a reliable estimate based on several and old initiatives, which can be beneficial because estimating software costs would be more difficult without any sample data at the start. Some of the criteria indicated in the first two tables are invisible and intangible at the start of the project process, such as database size and source line of code complexity of functions, as well as the ability of developers to address them.

C.O.C.O.M.O.

Barry Boehm (Rahikkala et al., 2018) proposed the Constructive Cost Model in 1981. (C.O.C.O.M.O.). COCOMO II, an upgraded version, was introduced in the year (1985) and has been altered since then (Ali et al., 2019). COCOMO-II is the utmost widely use approach for predicting software project costs at the moment [23]. It is widely used in all sizes of enterprises all around the world. COCOMO II is a functionally explicit algorithmic estimating model comparable to the Function Point method (Khan et al., 2021; Wickramaarachchi et al., 2017; El Bajta et al., 2020). It connects the independent variables to the dependent-variable exertion in person months. The total project expenses are calculated by multiplying the number of person-months by the total quantity of person & months. The basic formula is as follows:

$$Effort = A \cdot x \cdot (Size)^B \quad (1)$$

The size is measured in K.D.S.I. ("kilos. of delivered. Source. Instructions."). To standardize COCOMO-II to the unique C.O.C.O.M.O. scheme database, the coefficient A is set to 3.0. The scaling factor, indicated in the equivalence by the exponent B, models the frugalities and dis-economies of scale that ascend as the scope of a software plan develops. When B equals one, the scale economies and diseconomies are balanced, resulting in linear extrapolation. This method is used to calculate the cost of modest projects. However, most of the time, 0 B >1. is expected, indicating that the project suffers from scale dis-economies. As a project expands in scale, so does the overhead for planning, Communication, and integration (which can be reduced to some extent by initial risk management, using methodically authenticated design stipulations, or steadying requirements). If B = 1.0, the project aids from financial prudence of scale as a result of specialization gains, such that doubling the project size costs less work than doubling the effort. When examined more closely, B is the product of the following additive factors:

$$B = 1.01 + 0.01 \sum_i W_i \quad (2)$$

Where Wi represents the five precedent factors: progression flexibility, architecture and design resolution, team cohesion, and process maturity. The nominal person-month obtained Estimates can be considerably improved by multiplying them by a number of cost factors known as effort multipliers (E.M.). They are estimated separately and then combined, yielding:

$$PM\ adjusted = PM\ nominal \cdot x \cdot (\prod i\ EMi) \quad (3)$$

As shown in the table, all additional GSD operations must be covered by a single project factor termed Multi-site Expansion. This cost driver is determined by being around two variables: multi-site apposition (from fully collocated to international) and multi-site infrastructures (varieties from some mail to collaborating multimedia). As we'll see in the following section, this may not be enough to produce a more specific cost estimate for GSD projects. The making

1532-5806-25-S4-16

Citation Information: Gani, A., Akhunzada, A., & Junaid, M. (2022). Software cost estimation in global software development using hybrid approach. *Journal of Management Information and Decision Sciences, 25(S4),* 1-25.

selection, for example, has a substantial impact on exertion and project price since it determines the allocation and direction of work across sites. This is one of the features that the Multi-site Expansion factor lacks."

Product	Required Software Reliability Database Size Product Complexity Required Reusability Documentation match to life-cycle needs
Platform	Execution Time Constraint Main Storage Constraint Platform Volatility
Personnel	Analyst and Programmer Capability Application Experience Platform Experience Language and Tool Experience Personnel Continuity
Project	Use of Modern Programming Practices Use of Software Tools Multisite Development Required Development Schedule Classified Security Application

FIGURE 3
EFFORT-MULTIPLIER COST DRIVERS

Litoria and al. in 2012 (Litoriya et al., 2012) examined cost drivers that directly affect the accuracy of the cost estimate model and have proved the cost drop by Agile COCOMO II for drivers with the closest values. A new study based on the C.O.C.O.M.O. model was presented in 2017 by Saljoughinejad and Khatibi (2005) to improve the precision of the software cost estimate process. Due to its flexibility and its adaptability to different sorts of projects, the C.O.C.O.M.O. model was picked. During the investigation, multiple meta-heuristic techniques have been used to analyze the cost drivers. An efficient choice of the components and coefficients employed in the model was used for the development procedure. The enhancement included cost-drivers and estimates of constants. The results presented that when a judgment between the suggested C.O.C.O.M.O. model and other models was made; explicit superiority was achieved."

Chen et al. in 2005 (Chen et al., 2005) focused on the assessment of software costs by using dissimilar models such as C.O.C.O.M.O. and how the WRAPPER functionality in D.M. may be enhanced. It depends essentially on the selection of a subset of features (F.S.S.), where it focuses mostly on the greatest auspicious fields in the dataset. So, they concluded with the use of the WRAPPER feature that the C.O.C.O.M.O. model may be improved and its findings efficient. To select appropriate Artificial Intelligence (A.I.) techniques necessary for new projects, Chalifelu and Gharehchopogh in 2012 introduced numerous software cost approximation models based on data mining approaches. Their major purpose was to examine several ways of data mining about the exactness of the forecast and compare them with intermediate C.O.C.O.M.O. models. The results achieved were promised and seen.

In addition to numerous researches on and improvement of the C.O.C.O.M.O. model, other studies are undertaken which attempt to offer new models and techniques that could offer worth to the area. "Whigham-et-al. in 2015" (Whigham et al., 2015) proposed a basic model that

1532-5806-25-S4-16

Citation Information: Gani, A., Akhunzada, A., & Junaid, M. (2022). Software cost estimation in global software development using hybrid approach. *Journal of Management Information and Decision Sciences, 25(S4),* 1-25.

would be essential for all projects under development, and a study on the Estimate of software work would be necessary. When using a new or existing approach to study the prediction process, it is highly crucial to compare the outcomes with the baseline. Their proposal was known as the Automatic Transformations Linear Model (A.T.L.M.) as a basis for comparing various methods for estimating the software work.

Sarro et al. in 2016 (Sarro et al., 2016) have implemented a new effort estimating algorithm based on the investigation and calculation of the mean complete error combination (M.A.E.). Based on three factors, the proposed algorithm was tested and assessed, although the findings were extremely promised. The tests were performed using the data set acquired from over 700 software programs. A new software methodology for feature selection has recently been developed by Masoudi-Sobhanzadeh et al. in 2019 (Masoudi-Sobhanzadeh et al., 2019). It may be used in several sectors, including drug design, biology, and processing of images. The model begins with a subgroup of features/factors, which are based on optimization algorisms that are eventually passed on to graduates and students. SVM, ANN, and Decision-Tree are among the learners that can be utilized for "regressions-and-datasets." Researchers can use two types of optimization procedures and three classifiers to implement this model called Feature Select. The chosen feature was evaluated on eight different size and nature datasets, producing astonishing results.

Machine Learning

Machine learning algorithms, on the other hand, are considered crucial. In numerous research lessons, ML approaches are frequently used, and their outcomes are credible and dependable (Vig et al., 2018). An investigation of 25 releases encompassing 100 classes was carried out in an in-depth study by Vig & Kaur (Vig et al., 2018) in 2018 to examine the effort indicators. The study found that IBk, KStar, Additive Regression, and Multilayer Perceptron, out of the 18 machine learning algorithms used, could accurately predict the test effort. Furthermore, a forecast model for the timeframes of software processes using ML algorithms was proposed by Khalid et al. in 2017 [34]. There are two training models used for testing and evaluating the F.F.N.N. and R.B.N.N. algorithms, Levenberg–Marquardt (L.M.) and the Bayesian regularization backpropagation (B.R.). The two models revealed a slightly better comparison of B.R. results. Furthermore, B.R. is more desired since it is cost-effective in its application. The two machine learning techniques were proposed by Yeh and Deng in 2012 (Yeh et al., 2012) to anticipate the software product life cycles. The research developed a more accurate and widespread product cost estimation approach. “

Research on breast cancer was conducted in 2019 to provide modeling and analytical signals on the survival rate of breast cancer using ML algorithms (Ganggayah et al., 2019). Forecast algorithms have been established by the use of the thrilling boost, conclusion tree, neural networks, random vector provision machines, and logistical regression to control the main characteristics of breast cancer survival. All the algorithms were extremely great and close, and the highest algorithm was haphazard forests that can conclude from predictive models for breast cancer investigations. Software cost estimating processes, such as technology and creativity aspects, frequently take on some key problems. In addition, unpredictable difficulties such as a shortage of resources or an increase in O.S. costs may arise during deployment. Based on the combination of COACuckoo and K.N.N., Kumari & Pushkar, in 2018 (Kumari et al., 2018), devised a hybrid algorithm for further assess S.C.E. The hybrid algorithm is performed on six

Citation Information: Gani, A., Akhunzada, A., & Junaid, M. (2022). Software cost estimation in global software development using hybrid approach. *Journal of Management Information and Decision Sciences, 25(S4),* 1-25.

different datasets and assessed with eight criteria analyzed. The total results demonstrate an enhanced cost estimate accuracy.

A technique based on machine learning algorithms has been developed by Pospieszny, et al. in 2018 to reduce the gap between new research and true application. The findings obtained for the projected model were quite exact, and the writers feel that it provides real results for practical projects about software and time estimation. Pandey in 2013 also investigated most of the methodologies and approaches utilized in the Estimate of software costs. He tried to outline some of everyone's significant pros and disadvantages. He stated that all the project components are crucial to assess, estimate and differentiate the cost of a project from one project to the next. Quality elements are the major aspects to be considered in estimates: team knowledge, the atmosphere, and culture; project size and accessible resources are the quantitative components. Başkeleş, et al., in 2017 conducted numerous software effort assessment studies based on three primary and differing data sets utilizing several machine learning algorithms. In conclusion, they discovered that software effort is estimating parametric process mod.

Various mathematical models, such as algebraic equations, differential equations, and finite state machines, can be used to represent systems. In a fuzzy model, the relationships between variables are defined by if-then rules. In essence, a fuzzy model is a handy tool for investigating and displaying the estimated and vague nature of the real world. When typical quantitative techniques cannot be utilized to study the systems or when the available information on the systems is imprecise or wrong, a fuzzy model is extremely beneficial. By linking qualitative standards of one adjustable to qualitative standards of added, the rules, for example, start reasonable links in-between the system's variables. Linguistic words are used to refer to qualitative values with a distinct linguistic interpretation. The term fuzzy identification (Ziauddin et al., 2013) typically refers to the techniques and strategies used to generate fuzzy models from data. There are two methods for creating a fuzzy model from data:

1. Expert knowledge expressed verbally and interpreted into a set of if-then rubrics. A specific model assembly can be formed, and its limitations, such as membership purposes and rule weights, can be altered using input and output data.

2. To begin, no previous knowledge of the system under training is used to develop the rules, and a fuzzy model is built from data using a specific algorithm. System behavior is predictable to be explained by extracted rubrics and membership purposes. An expert can alter or create new rules grounded on his or her own involvement. Skilled alteration is optional in this method.

Our research will be focused on the second technique, in which fuzzy models are produced automatically from data. We used the Takagi–Sugeno fuzzy model (Mukhopadhyay et al., 1992; Gharehchopogh, 2014) in our work, and the data can be numerical or linguistic. A fuzzy model's two essential components are membership purposes and instructions. The algorithms for association purposes, the Takagi-Sugeno fuzzy model, and rule abstraction will be covered in the sections that follow."

METHODOLOGY

The absence of united cost approximation techniques to use in the software development procedure can be explained as the motivation for such a methodology. In today's submissions, accessible cost approximation techniques are mostly used as stand-alone solutions. The majority of the methods are organization-exact, as are the perimeters used in those methods. There is a

Citation Information: Gani, A., Akhunzada, A., & Junaid, M. (2022). Software cost estimation in global software development using hybrid approach. *Journal of Management Information and Decision Sciences, 25(S4),* 1-25.

need for a more comprehensive and integrated approach that incorporates the most widely accepted and used cost estimation techniques. It is more generic and used to the complete software life cycle in this context. The technique essentially recommends the use of a series of cost estimation tools in a step-by-step, integrated manner to improve project planning and scheduling.

Step 1: Determine the Size of the Project

The basic activity in this step is to estimate the size of the software product under consideration (i.e., the number of non-commentary basis declarations). The methodology recommends employing the previously described methods of Function Points and Conversion of Function Ideas into N.C.S.S. for size estimation. Depending on the methodology and stage of development, various more unique size estimating techniques can be applied. The N.C.S.S. estimate for the software product that will be developed is the primary output of this step.

Step 2: Calculate the Cost of Effort and Time

It is proposed that COCOMO II and its variants be used to estimate effort and time in terms of work and months and months, respectively.

Step 3: Spread out the Exertion and Time Costs across the Life Cycle

For distribution, some methods founded on involvement in the sol, ware manufacture environmental, as well as some strategies such as those provided in the COCOMO II technique, may be used.

Step 4: Convert the Costs to Calendar Time

The method should include some criteria for systems forecasters and other expansion workers. The outputs are supposed to be calendar period and monetary charges."

Afterward, Step 4 is accomplished, one of the automatic project administration programs can be utilized to manage and schedule the project. It is regarded as successful when an estimates approach is simple to comprehend, can be adjusted during the software expansion process, and the early evaluation is within +/- 30% of the definite final cost. As a result, the cost estimation procedure should be employed multiple times during the development stage. It should be carried out at various phases of the development procedure. After each repetition, the estimated findings should be advanced by associating them with earlier estimations. The steps of smearing the cost approximation methodology may differ depending on a variety of elements such as the development model and/or methodology utilized, the scope of the software invention, and so on. The section that follows explains the methodology's general use during software development utilizing Yourdon's Structured Design/Development approach.

The Nasa software project data set [26] is used as a benchmark for assessing the proposed approach. It consists of two independent features, Developed Lines (DL) and Methodology (M.E.), as well as one dependent feature, Software Cost Estimation, which will be predicted/estimated (S.C.E.). To provide a more accurate size measurement, the L.O.C. (lines of code) value of the software is used to calculate the DL feature, which takes into account both

1532-5806-25-S4-16

Citation Information: Gani, A., Akhunzada, A., & Junaid, M. (2022). Software cost estimation in global software development using hybrid approach. *Journal of Management Information and Decision Sciences, 25(S4),* 1-25.

implemented and re-used lines of code. It is worth noting that the DL measure takes into account comments as well as 20% of re-used lines (Litoriya et al., 2012). The ME functionality is provided in accordance with the methodologies used in software project development. The ME feature is calculated by taking the following things into account: Official test plans, certification, and training are all employed (Litoriya et al., 2012).

To address the software cost estimating (S.C.E.) problem, the morphological rank linear hybrid. The design (M.R.L.H.D.) method is planned. It employs an evolutionary search technique (Chen et al., 2005) to locate the ingredients required for the construction of a precise system. I the fundamental data needed to approximation the cost (the dimensionality or number of structures required to label the data), and (ii) the model construction, parameters, and training procedure. Because the model must be as simple as feasible, the data with the fewest dimensions must be examined.

The planned method leverages a modified. genetic. algorithm (MGA) to generate morphological-rank-linear (MRL) perceptron's (Whigham et al., 2015). (which employs optimal genetic operators to accelerate convergence). The M.G.A. (Sarro et al., 2016) is used to calculate the exact dimensionality of data and the correlating specific characteristics to show the information (first, a maximum dimensionality (MaxDim) is classified by the number of individual functionalities, and then the M.G.A. can select any significance in the span (Chalifelu et al., 2012)for each element in the community), as well as to search for the right initial MRL back propagation parameters.

Framework

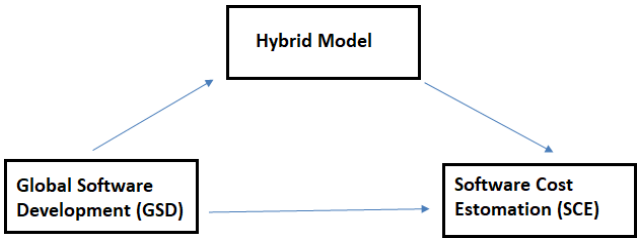

FIGURE 4
HYBRID MODEL OF SOFTWARE COST ESTIMATION

There are two hypotheses of this research:

H1 Global software development impact on software cost estimation positively.
H2 Global software development impact on software cost estimation, mediation role of hybrid model.

The framework illustrates how we will check the flow of research and which factors impact which factor. Through the given framework, we will prepare a hypothesis, in which we will check the validity of the framework and hypothesis. In this framework, we introduce an independent variable, dependent variable, and mediation in between variables. The first variable is GSD (Global Software Development) is an independent variable, and we will check the impact of GSD on the others variables. S.C.E. (Software Cost Estimation) is the dependent variable that can affect GSD, or we can say due to the GSD, the scenario can be changed. There is one mediation available, and due to the mediation of the Hybrid model may be the effect on S.C.E.

1532-5806-25-S4-16

Citation Information: Gani, A., Akhunzada, A., & Junaid, M. (2022). Software cost estimation in global software development using hybrid approach. *Journal of Management Information and Decision Sciences, 25(S4),* 1-25.

values will be changed. So we will check the framework by creating a questionnaire and solve that questionnaire by the respondent. Respondents can tell us the effects of these variables by giving their opinions.

Through this framework, we create two hypotheses. In the first hypothesis, we will check the impact of global software development on the software estimation cost positively. So, if due to the GSD if is there any positive change in the software cost estimation or not. We will check this change in positive aspects. In the second hypothesis, we will check the impact of GSD on the S.C.E. by the mediating role of the hybrid model. Is there any effect on cost estimation due to the hybrid model or not?

Overhead costs are chosen as the basic concern in the context of GSD. Afterward, we carried out a planned S.L.R. to derive GSD-specific cost estimate models and cost drivers. The results achieved using S.L.R. will then be confirmed by publicly available data (GSD-based). Because of this, we have emphasized the important cost drivers by conducting several statistical analyses. Lastly, based on our results, we provided a conceptual model. The ultimate purpose of this research and its future direction is to formalize the proposed model to help practitioners to examine the additional issues of cost assessment.

Table 1
SURVEY OF CONDUCTED RESEARCH IN VARIOUS REPOSITORIES

Databases	Papers extracted through Search Terms	Papers extracted based on introduction and conclusion	Papers extracted based on title and abstract	Papers extracted based on full texts
Google Scholar	213	150	197	50
IEEE Xplore	189	112	98	70
ResearchGate	300	198	223	90
ACM	178	134	101	39
Springer Link	183	126	112	57
Science Direct	90	97	76	88

The research used the inclusion/exclusion criteria for article selection to do away with those that do not fall under the desired qualification, including those that gave ambiguous information. The criteria for inclusion and exclusion in our research were based on Kitchenham's recommendations in 2004 (Kitchenham, 2004). The following table highlights Kitchenham's inclusion criterion (Kitchenham, 2004) that the research adopted when choosing the viable articles.

Table 2
INCLUSION CRITERIA

1.	The selected primary study should be a journal, conference or book chapter
2.	The studies that focused on cost estimation in the GSD context
3.	The studies that discuss the challenges or cost drivers affecting cost estimation in GSD
4.	The studies that present GSD specific cost estimation models or techniques
5.	The selected studies must be available full-text articles, specifically in the English language
6.	Most of the studies published between 2010-2021

Citation Information: Gani, A., Akhunzada, A., & Junaid, M. (2022). Software cost estimation in global software development using hybrid approach. *Journal of Management Information and Decision Sciences, 25(S4),* 1-25.

Table 3 below further represents the exclusion criteria that were used to eliminate the studies and literature that the research employed and adopted.

Table 3 INCLUSION CRITERIA	
1.	The studies that do not answer the research questions
2.	The studies that do not discuss the cost estimation process in the GSD context
3.	The studies that do not highlight the challenges or cost drivers of cost estimation in GSD
4.	The studies that do not highlight the various cost estimation models
5.	The studies that were not written in English

RESULTS

To gain the industrial view of the cost drivers derived from literature, an online questionnaire was devised. The project managers of multinational software businesses were the target respondents to the questionnaire. A total of 175-project-managers, all of whom were multinationals, were embattled. In addition, the questionnaire for the legitimacy of the findings was distributed in over 20 nations. Each identified cost driver used five Likert scales for the questionnaire ("Strongly agree, Agree, Neutral, Disagree, strongly disagree"). The results were transformed into percentages and then enhanced utilizing the tools of data analysis.

Table 4 DEMOGRAPHIC CHARACTERISTIC OF THE SAMPLE GENDER					
		Frequency	Percent	Valid Percent	Cumulative Percent
	Male	50	100	100	100
	Female	0	0	0	00
	Total	50	100.0	100.0	

Table 5 MODEL SUMMARY OF LINEAR REGRESSION										
Model		**Square**	**Adjusted R Square**	**Std. The error of the Estimate**	**Change Statistics**					**Durbin-Watson**
					R Square Change	**F Change**	**f1**	**f2**	**Sig. F Change**	
1	454[a]	206	0.203	822.768	0.206	64.351		48	.000	2.160
a. Predictors: (Constant), GSD										
b. Dependent Variable: S.C.E."										

Table 6 ANOVA[a] TEST						
Model		**Sum of Squares**	**f**	**Mean Square**	**F**	**Sig.**
	Regression	43562490.310	1	43562490.310	64.351	000[b]
	Residual	167883071.700	48	676947.870		
	Total	211445562.000				

1532-5806-25-S4-16

Citation Information: Gani, A., Akhunzada, A., & Junaid, M. (2022). Software cost estimation in global software development using hybrid approach. *Journal of Management Information and Decision Sciences, 25(S4),* 1-25.

			49			
a. Dependent Variable: SCE						
b. Predictors: (Constant), GSD						

The following table shows the results of an F test. The hypothesis in F-linear regression testing is that the model explains zero variation in the dependent variable (in other words, R Square = 0). Because F testing is critical, we can conclude that this model explains a large portion of the project's fluctuating success. In the above table, the significance value is less than 0.05, indicating that the hypothesis is accepted.

Table 7
MODEL SUMMARY

Model		R Square	Adjusted R Square	Std. The error of the Estimate	Change Statistics					Durbin-Watson
					R Square Change	F Change	f1	f2	Sig. F Change	
	763[a]	582	.579	598.084	.582	172.059		47	.000	2.229
a. Predictors: (Constant), GSD, HM										
b. Dependent Variable: SCE"										

Table 8
ANOVA[a] TEST

Model		Sum of Squares	f	Mean Square	F	Sig.
	Regression	123092413.700	2	61546206.860	172.059	000[b]
	Residual	88353148.290	47	357705.054		
	Total	211445562.000	49			
a. Dependent Variable: SCE						
b. Predictors: (Constant), GSD, HM						

Coefficient of determination

Since the value of Adjusted-R-Square is .579, which means the modal is predicting the dependent variable by 58% approximately.

The **Durbin-Watson statistic** is 2.229, which is between 1.5 and 2.5 and therefore, the data are not autocorrelated.

ANOVA

H_1:The modal is insignificant
H_2:The modal is significant

The sig. value of ANOVA is less than 0.05, so we reject $\boldsymbol{H_1}$ and accept $\boldsymbol{H_2}$, which means the modal is significant.

1532-5806-25-S4-16

Citation Information: Gani, A., Akhunzada, A., & Junaid, M. (2022). Software cost estimation in global software development using hybrid approach. *Journal of Management Information and Decision Sciences, 25(S4),* 1-25.

Sobel Test for Mediation

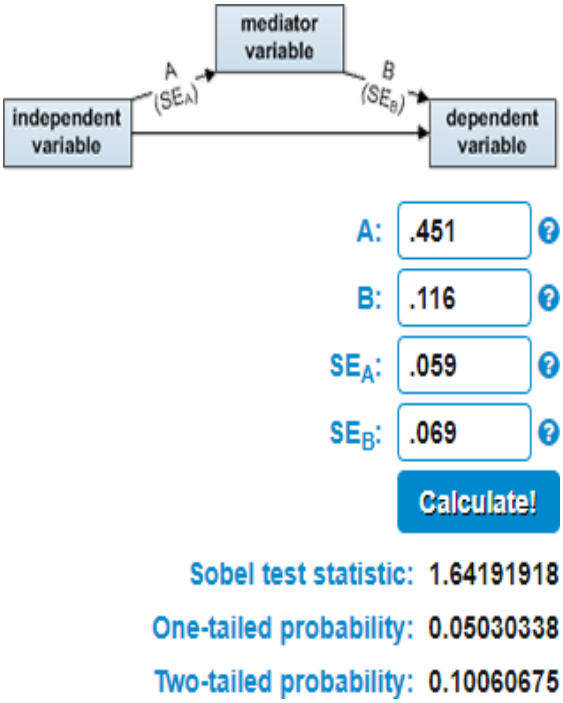

FIGURE 5
SCORES OF VARIABLES USED

Table 9
RELIABILITY ANALYSIS

Sr.	Variable Name	No. of Items	Cronbach's Alpha
1	Software Cost Estimation Techniques	5	.731
2	Software Cost	6	.855
3	Product Performance Flexibility of Software	4	.798

Above the given table is explaining the reliability of all the variables used in the model. With references to the no. of items, the total questions were asked from the respondents to evaluate the variable. Cronbach's Alpha is acceptable with the value of .700. In the above-given tables of Reliability analysis, all the variables are acceptable with their Cronbach's Alpha values as; Software Cost Estimation Techniques Cronbach's Alpha value is .731, Software Cost Cronbach's Alpha value is .855, Product Performance Flexibility of Software Cronbach's Alpha value is .798.

DISCUSSION

According to our framework, and to prove the hypothesis, we develop a questionnaire and collect the data of respondents. After the collection of data, we applied some tests on data, and, according to those tests, this proved that our hypothesis was correct. The GSD has an impact on the S.C.E. positively because we have to already follow the instructions defined in the GSD for the cost estimation. The second hypothesis tells us that GSD also has an impact on

1532-5806-25-S4-16

Citation Information: Gani, A., Akhunzada, A., & Junaid, M. (2022). Software cost estimation in global software development using hybrid approach. *Journal of Management Information and Decision Sciences, 25(S4),* 1-25.

S.C.E. due to the hybrid model of GSD. So we will always follow the guidelines which are available in the GSD models.

In the GSD context, there are numerous other cost drivers concealed from the estimate process. Our investigation found 21 initial cost factors, which were subsequently refined by the empirical investigation carried out. In the GSD context, there are numerous other cost drivers concealed from the estimate process. Our investigation found 21 initial cost factors, which were subsequently refined by the empirical investigation carried out. There are additional obstacles that relate to the cost estimate which are not mentioned in the literature, such as higher authorities' pressure, unforeseen impediments, and the engagement of many vendors. Time differences can be calculated through overlapping time. The metrics need to be developed with these aspects in mind. Finally, two basic groupings of these factors have been discovered. For compensating for the additional elements, 4P and P.M.B.O.K. might be utilized. P.M.B.O.K. is used to classify high-level drivers, while 4P's could be used to categorize small-cost drivers. The determining factors for improved visualization and analysis for the knowledge fields are offered in 4P's taxonomy.

Most of the methodologies employed in the GSD are complemented by current methods of cost estimating to estimate the projects collaborating. The techniques are recognized early and are not measured. The methodologies are not verified, and the number of organizations is restricted. Lacking technical cost components can lead to incorrectly estimated outcomes. There is, then, a difference in improving these models for the GSD context for researchers and practitioners. Certain existing models are context-specific and are therefore built for a certain setting such that the GSD context cannot generalize them.

Characteristics of Complexities in GSD

The development of global software in recent years has gained momentum. Many software projects now have many time zones apart, involving asynchronous collaboration between geographically distant teams. Due to factors such as team-building effort and knowledge transfer, developing a software architecture that is easy to distribute and reduces cross-site Communiqué, facilitating Communiqué among remote teams working with interconnected architecture, and daily work, estimating software costs for such projects becomes difficult (Sangwan, 2017). A wide range of research is carried out in remote and worldwide collaboration, but only a few papers contain problems and procedures derived from genuine projects (Lassenius et al., 2003). Our most validated findings have been presented in a variety of Siemens dispersed projects (Herbsleb et al., 2005). Furthermore, most of the literature focuses on Communication, team growth, or tool support. There is no doubt that all of these elements are significant, but we are particularly interested in the interaction of these factors and their impact on project costs.

True cooperation and information exchange across geographically remote teams were challenging and infrequent, uniform with a wide range of association technologies and face-to-face contacts among chosen team followers (Rahikkala et al., 2018). As a result, regardless of the method used to generate it, the initial cost estimate will be excessively low once one component of project members is relocated to a dissimilar physical position. Estimates become substantially more imprecise when teams differ in time, culture, and language. As a result, the sources of "intra-project variance" must be investigated directly.

Citation Information: Gani, A., Akhunzada, A., & Junaid, M. (2022). Software cost estimation in global software development using hybrid approach. *Journal of Management Information and Decision Sciences, 25(S4),* 1-25.

Product Elements/Factors

Historical Precedent: new software development, demonstrating the level of creativity that is closely proportional to the extent of spontaneous Communication and the need for specific domain skills and understanding, as well as the occurrence of unplanned changes (Sharma, 2017).

Architectural Worthiness: the tasks of Organization and functional connectivity between software units have to be carefully crafted (Silhavy, 2019). Consequently, architecture plays an important role in influencing the coordination of the phase of development. Architectural appropriateness indicators may include modularity, interface match and dependencies, architectural Communication, etc. Architectural appropriateness indicators may include modularity, interface match and dependencies, architectural Communication, etc.

Personnel Elements/Factors

Cultural Conformity*:* compatibility of the mental models of the team members (Herbsleb et al., 2005) impacted by the combination of participating countries, international team experience, etc.

Skill Qualifications: educational0level and linguistic skills showing remote teams' formal skills (Singh et al., 2020).

Shared Understanding: implicit and inferred knowledge needed, reflecting the documentation level and specification thoroughness, and the common understanding of the objectives (Herbsleb et al., 2005; Sharma et al., 2019). Information sharing constraints, i.e., when working with external suppliers or in safety-sensitive contexts, reflect competitive restrictions in information distribution.

In the vast area of research on distant and global collaboration, the relevance of the less formal variables, commonly called "soft skills," was made obvious. Hersleb and Grinter (1999) found it sensitive to imperfect foresight and unforeseen events while organizations try to coordinate cross-site work. This requires informal Communication and negotiating skills from workers to resolve these problems. In addition, GSD makes it harder to do all these activities; because of the lack of delicate communications mechanisms, remote coworkers are generally less likely to be useful, less able to adapt to changes rapidly, and more projects are delayed (Hersleb et al., 1999). GSD needs more planned patterns of Communication and, ironically, greater flexibility and process adaptively. These aspects have been attempted by us in the offered Personnel Factors/elements.

The project Factors

Collaboration Model: starting search and negotiation of contracts with offshore partners (Woznicki et al., 2019).

Tools and Infrastructure: the uniformity of the tool chains used in all locations and the possible expense of building infrastructure in remote locales. The results are noted based on the research questions.

Citation Information: Gani, A., Akhunzada, A., & Junaid, M. (2022). Software cost estimation in global software development using hybrid approach. *Journal of Management Information and Decision Sciences, 25(S4),* 1-25.

RQ1; Which Cost Estimation Technique is Best to find the Actual Cost of a Software Project?

This study took the industrial standpoint of GSD environment cost estimate models. To do so, we identified the models through a practitioner's questionnaire for estimating the costs in the context of GSD. It is noted that corporations are not dependent on a particular cost estimate model but are using many models to estimate a project (Mohan et al., 2020). The study (Mohan et al., 2020) also shows that corporations and industries gain from the combined use of multiple models. A mixture of cost estimate models in GSD is used by many enterprises (Mohan et al., 2020). Our expert judgment and analogy models are still very frequent; thus, we still have deficient an adequate formal model. GSD-based organizations, because the results of the formal model in this respect are not yet adequate, nevertheless choose non-formal cost estimates to be used; the additional GSD costs drivers are not taken into account.

Although cost estimates for the Estimate of a project are diverse, the author Koskenkyla in 2012 (Koskenkyla, 2012) examined the algorithmic and non-algorithmic cost estimating methodologies; however, those methodologies do not fit for GSD owing to further obstacles posed in the development of global software. GSD approaches are increased by adding more components to the current cost estimate. In the majority of the research papers, the algorithmic methodology has been extended, i.e., the COCOMO II model for the S.D.C. context was expanded by Keil et al. in 2006 (Keil et al., 2006), Betz and Makio in 2007 , Madachy in 2007. However, some other strategies are also introduced through overhead analysis. By augmenting the COBRA basic model, (Lamersdorf, 2010) established a model based on the whole cost. These strategies are validated early and not properly. There are several cost drivers not identified in the literature with these procedures. Thus, a gap could be filled for investigators if we take into account the aspect missing: the changes in the time zone. Given such elements, the efficiency of the Estimate is very vital, or the hidden factors can overrun our budget.

RQ2; What Factors do Current Cost Estimation Techniques Fail to Incorporate?

With all the above different GSD-specific cost-estimation prototypes, the GSD-specific cost estimation model still has a number of shortcomings and lacks certain factors. A model for cost estimation, COCOMO II, was built for estimating the GSD effort as provided by Betz and Makios in 2007. To make it effective, new effort multipliers were introduced. The model has various limitations, among which is its limitation to the partnerships to two. There is no systematic methodology for quantifying variables leading to ambiguous numerical values. A cost-overhead model was created based on manipulating factors affecting costs and influencing is the creation, and causal connections were introduced by Lamersdorf in 2010. The model was GSD's overhead cost model (CoBRA Based). The causal model can help to comprehend the significance and relationships of the components. In one Organization, all interviews were performed. Its shortcoming was that, as a company-specific model, this concept could not be generalized.

In 2012 Mamoona Humayun showcased an outline and review of various methods to be used when estimating costs of machine learning in GSD. The techniques handle GSD's distributed features, and depending on the circumstances; they are available. Factors that drive the cost are not discussed, though. The precision and correctness of these models depend heavily

1532-5806-25-S4-16

Citation Information: Gani, A., Akhunzada, A., & Junaid, M. (2022). Software cost estimation in global software development using hybrid approach. *Journal of Management Information and Decision Sciences, 25(S4),* 1-25.

on the sort of data they are trained on, and the inaccurate estimates may lead to inadequate data. The Manal El Bajta in 2015 analogy-based cost estimating methodology refers to case-based justification where it is anticipated that similar projects will cost similar but first need to find similarities to calculate their costs in the instance of software development. The model is relevant only if we know similar projects. The problem with this form of Estimate is when there are no data or no similar projects in the past. Ramachandran's scheduling-based cost estimates in 2016, which identifies scheduling parameters and productivity parameters and calculates costs based on a code line. By differentiating it and contrasting it with other models that cannot be done in any other model, the efficiency of the model is assessed. Its shortcoming is that no change in data is taken into account, but as stated in future research work.

S.O.C.E.M., Software Outsourcing Model Cost Estimate (S.O.C.E.M.) by Jamshed Ahmad in 2018, a model that identifies the vendor organization difficulties in the GSD cost estimate. The model addresses the cost estimating difficulties. There is an abstract estimate model. No debate on cost drivers and cost drivers by the model and its Organization specific. Madachy in 2007 suggested a phase-sensitive multiplier model (Cost Xpert). The model takes into account the cost considerations for the individuals in various teams. In addition, with industrial cooperation, the model has been developed, and effort multipliers are sensitive to phases. The proposed model is not quantified and validated. COCOMO II's additional cost drivers, i.e., Communication and collaboration, are not included. To compute the trade-off between the collocated Estimate and GSD, a decision-making context is provided in the model of COCOMO II Based by Keil in 2006. The model introduced multi-site Communication and multi-site collocation factors. The complexity factor in this model is not quantified, and there is no systematic technique to extracting factors.In addition to the confirmation of the literature cost-driving, the project managers have expressed interest in presenting us with their experiences with several further difficulties which can have an impact on the cost estimate in GSD. This includes other cost drivers;

Customer participation is an important aspect of project management. Projects without client participation can lead to problems, and in time or budget, the project can exceed. Process maturity also shows that our processes develop and standardize themselves to fulfill a particular project. If there is no formal procedure, we will be compensated for our time and costs. The effort to manage projects is not as important as the literature for the industry. We assessed that practitioners take into account the measurable parameters which may be calculated and used. The effort invested in the running and management of a project, however, is considered as a theoretical concept, very abstract; nor can it be assessed correctly. Thus, from an industrial point of view, its share is below 50 percent. The only reason that some cost drivers' percentages vary is that practitioners have a more quantitative element, while literature takes into account aspects that are not quantitative. These are ultimately just measurable cost drivers which the cost assessment models require.

Adaptability to emerging technologies of the Organization. Selection of a project on the most recent technologies can sometimes influence cost estimates owing to the lack of understanding. Higher authority pressure. Pressure. The costs and effort of the project are adjusted because of the pressures from higher authorities concerning "Winning a project." Engagement of Multiple-Sellers. Cost estimates are impacted when several suppliers engage in their various corporate politics. Over rental. Over rental. This isn't typical; however, if an enterprise employed engineers that are not needed for the running of the current project, the price and costs may be affected. It matters to choose the appropriate quantity. Failure to work on the

1532-5806-25-S4-16

Citation Information: Gani, A., Akhunzada, A., & Junaid, M. (2022). Software cost estimation in global software development using hybrid approach. *Journal of Management Information and Decision Sciences, 25(S4),* 1-25.

Q.A. The work needed to ensure quality should not be overlooked, or the total cost of a product could otherwise be affected. Unforeseen obstacles. These industries ought to have a strategy lay-out and further prepare a budget and plan for unforeseen obstacles to adequately address the existing status of Covid-19. Otherwise, these unexpected obstacles may impair the development process.

RQ3; What are the Drivers of the GSD Cost that could also Influence its Estimate?

Cost drivers may be good or negative elements that affect the worldwide development of software in a multidimensional way. Cost drivers are features of the creation of software that influence effort in the realization of a given project (Suliman et al., 2017). Cost drivers are picked on the basis that they have a linear effect on effort in comparison to the scale factors. In the COCOMO II model, the efforts of development are adjusted by 17 effort multipliers (E.M.s). Each multiple cost driver is given the same rating level as the combination of allocated weights (Rhmann et al., 2021). The transitional rating levels and weights for the effort multipliers can be assigned, as noted by Boehm (1984). They are also eligible for a mean value which indicates a fairer figure. Although the model sets a limited number of cost drives, COCOMO II enables the user to create a set of efforts to better match the prevailing circumstances in each development. Cost drivers are judged and based on a firm justification that gives details of a significant source of effort and/or productivity discrepancy autonomously (Tomáset al., 2017). Nominal values do not affect effort, while below/over value reduces/increases effort.

The drivers of costs in the GSD context are in their nature concealed due to their distributed characteristics. These cost factors are typically disregarded because of the distributed structure of GSD and production overhead costs later in the project. During the estimating phase, these aspects need to be examined to predict energy and means realistically. A completely of 17 of 23 papers targeted the cost-drivers, while several papers examined GSD cost estimation models specifically. Cost drivers were discussed, but actual proof to substantiate these cost drivers was lacking. At first, we got a list of literature's cost drivers, and the analysis was carried out. With the empirical investigation, the resulting cost drivers are validated to overcome the deficiencies of prior work. Table 10 presents the cost drivers derived from the literature. The name of the cost driver and its cost estimation impacts are on the labels.

Table 10
COST DRIVERS OF SOFTWARE COST ESTIMATION

Cost Driver	Low Impact on Cost	Significant Impact on Cost	Moderate Impact on Cost
The differences in time zones		XX	
Diversity in culture and language		XX	
The methods and process of Communication and infrastructure		XX	
The model of the process			XX
Cost of travel			XX
The degree of skill and Competence		XX	
The legibility of the requirement			XX
Compliance with the process		XX	
The delay in response			XX
Trustworthiness of the Team		XX	

1532-5806-25-S4-16

Citation Information: Gani, A., Akhunzada, A., & Junaid, M. (2022). Software cost estimation in global software development using hybrid approach. *Journal of Management Information and Decision Sciences, 25(S4),* 1-25.

Lack of client awareness	XX		
Resources that are shared			XX
The foundation of the team			
The dispersion and distribution of work			
The pressure involved in work			XX
Range of parallel sequential work handover	XX		
Client Specific knowledge	XX		
Lack of Client involvement			XX
Design and technology newness			XX
Team size			XX
Project effort			XX

Table 10 above further identifies the cost drivers that were obtained from the literature that was analyzed. Table 10 further highlights the effects these cost drivers have on cost. The cost drivers are categorized to have either a critical impact or a low impact on cost and a moderate impact, as pointed out in Table 4 above.

RQ4; How can the Cost Drivers that are defined be Categorized?

In most research, the elements are missing from appropriate groups, while some are represented using a typical grouping approach. Around 70% of the articles do not have the factor categorization. The lasting 30 percent of the papers grouped the items grounded on the standard P.M.B.O.K. 4P'S ("Process, project, staff, and product"). P.M.B.O.K. is an assembly of project management practices (Jain et al., 2018). The categorization of these cost drivers can be used to offset the extra GSD cost drivers. For these studies, 4P's were picked to categorize the cost drivers because the components that are defined and laid down to all 4P's categories are generalized and do not include specific cost estimating sub-areas when speaking on the P.M.B.O.K. (Sharma et al., 2019). Presenting the cost drivers in 4Ps could help project managers better understand how knowledge areas require cost estimating elements to be considered for improved estimates.

CONCLUSION

Despite geographical and time disparities, Global Software Development (GSD), which is scattered around the world and works in conjunction with partner companies, is becoming more widespread. The primary benefit of GSD is that it makes more human resources available at lower costs. The distance that separates development teams, on the other hand, causes several issues. When software development teams are spread out across the country, coordination and Communication become more difficult, resulting in hidden costs. As a result, GSD's work estimation models for collocated software are insufficient. As a result, assessing effort in GSD becomes a major research topic. Several researchers have focused on estimating effort in GSD throughout the last decade. This study presents the findings of a comprehensive literature review, including a summary of the expenses hidden in GSD with and without a hybrid model and an examination of unresolved research questions in GSD effort assessment.

1532-5806-25-S4-16

Citation Information: Gani, A., Akhunzada, A., & Junaid, M. (2022). Software cost estimation in global software development using hybrid approach. *Journal of Management Information and Decision Sciences, 25(S4),* 1-25.

REFERENCES

Ahmad, J., & Khan, A.W. (2020). Software outsourcing quality evaluation management model (S.O.Q.E.M.M.). 2020 3rd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET).

Ahmad, J., Khan, A.W., & Qasim, I. (2018). Software Outsourcing Cost Estimation Model (S.O.C.E.M.). A Systematic Literature Review Protocol, 2(1).

Ali, A., & Gravino, C. (2019). A systematic literature review of software effort prediction using machine learning methods. Journal of Software: Evolution and Process, 31(10).

Arslan, F. (2019). A review of machine learning models for software cost estimation. Review of Computer Engineering Research, 6(2), 64-75.

Attarzadeh, I., & Ow, S.H. (2011). Improving estimation accuracy of the COCOMO II using an adaptive fuzzy logic model. 2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011).

BaniMustafa, A. (2018). Predicting software effort estimation using machine learning techniques. 2018 8th International Conference on Computer Science and Information Technology (C.S.I.T.).

Baskeles, B., Turhan, B., & Bener, A. (2007). Software effort estimation using machine learning methods. 2007 22nd international symposium on computer and information sciences.

Betz, S., & Mäkiö,, J. (2007). Amplification of the COCOMO II regarding offshore software projects,. Offshoring Softw. Dev. Methods Tools Risk Manag.

Blaifi, S., Moulahoum, S., Benkercha, R., Taghezouit, B., & Saim, A. (2018). M5P model tree based fast fuzzy maximum power point tracker. Solar Energy, 163, 405-424.

Boehm, B.W. (1984). Software engineering economics. IEEE Transactions on Software Engineering, SE-10(1), 4-21.

Budgen, D., & Brereton, P. (2006). Performing systematic literature reviews in software engineering. Proceedings of the 28th international conference on Software engineering.

Chen, Z., Menzies, T., Port, D., & Boehm, B. (2005). Feature subset selection can improve software cost estimation accuracy. Proceedings of the 2005 workshop on Predictor models in software engineering - PROMISE '05.

Chirra, S. M., & Reza, H. (2019). A survey on software cost estimation techniques. Journal of Software Engineering and Applications, 12(06), 226-248.

El Bajita, M. (2015). Analogy-based software development effort estimation in global software development. Proc. - 2015 IEEE 10th Int. Conf. Glob. Softw. Eng. Work. I.C.G.S.E.W. 2015, 51– 54.

El Bajta, M., & Idri, A. (2020). Identifying software cost attributes of software project-organization in global-software-development. Minutes of the 13th International Conference on Intelligent Systems: Theories and Applications.

Ganggayah, M.D., Taib, N.A., Har, Y.C., Lio, P., & Dhillon, S.K. (2019). Undefined. B.M.C. Medical Informatics and Decision Making, 19(1).

Gharehchopogh, F.S. (2014). A novel PSO based approach with hybrid of fuzzy C-means and learning automata in software cost estimation. Indian Journal of Science and Technology, 7(6), 795-803.

Hasanluo, M., & Gharehchopogh, F.S. (2016). Software cost estimation by a new hybrid model of particle swarm optimization and K-Nearest Neighbor Algorithms. Journal of Electrical and Computer Engineering Innovations, 4(1).

Herbsleb, J.D., & Grinter, R.E. (1999). Splitting the organization and integrating the code: Conway's law-revisited. Minutes of the 21st international Conference on Software Engineering (1999). I.C.S.E. '99. IEEE Computer Society Press, Los Alamitos, CA, 85-95."

Herbsleb, J.D., Paulish, D.J., & Bass, M. (2005). Global software development at Siemens: experience from nine projects. In Minutes of the 27th international-Conference on Software Engineering. ICSE '05. A.C.M. Press, New York, NY, 524-533.

Humayun, M., & Gang, C. (2012). 111-K0003.

Ionescu, V. (2017). An approach to software development effort estimation using machine learning. 2017 13th IEEE International Conference on Intelligent Computer Communication and Processing (I.C.C.P.).

Jain, R., & Suman, U. (2018). A project management framework for global software developmen-t. ACM SIGSOFT Software Engineering Notes, 43(1), 1-10.

Keil, P., Paulish, D.J., & Sangwan, R. S. (2006). Cost estimation for global software development.

Khalid, A., Latif, M.A., & Adnan, M. (2017). An approach to estimate the duration of software project through machine learning techniques. Gomal University Journal of Research, 33, 1-13.

Citation Information: Gani, A., Akhunzada, A., & Junaid, M. (2022). Software cost estimation in global software development using hybrid approach. *Journal of Management Information and Decision Sciences, 25(S4),* 1-25.

Khalifelu, Z.A., & Gharehchopogh, F.S. (2012). Comparison and evaluation of data mining techniques with algorithmic models in software cost estimation. Procedia Technology, 1, 65-71.

Kitchenham, B. (2004). Procedures for Performing Systematic Reviews Kitchenham, B., 2004. Keele, UK, Keele Univ, 33, 1–26.

Kodmelwar, M.K., Shashank, D.J., & Khanna, V. (2019). Software Estimation using deep learning Mr. Manohar K. Kodmelwar, Shashank D. Joshi, V. Khanna. International Journal of Innovative Technology and Exploring Engineering (I.J.I.T.E.E.), 8(6).

Koskenkyla, J. (2012). Cost estimation in global software development - Review of estimation techniques.

Kumari, S., & Pushkar, S. (2018). Cuckoo search-based hybrid models for improving the accuracy of software effort estimation. Microsystem Technologies, 24, 4767-4774.

Khan, J.A., Khan, S.U., Iqbal, J., & Rehman, I.U. (2021). Empirical investigation about the factors affecting the cost estimation in global software development context. IEEE Access, 9, 22274-22294.

Lamersdorf, A., Fernández-Del Viso Torre, A., Münch, J., Sánchez, C. R., & Rombach, D.(2010). Estimating the effort overhead in global software development. Proc. - 5th Int. Conf. Glob. Softw. Eng. ICGSE, 267–276.

Langsari, K., Sarno, R., & Sholiq, S. (2018). Optimizing effort parameter of COCOMO II using particle swarm optimization method. TELKOMNIKA (Telecommunication Computing Electronics and Control), 16(5), 2208.

Lassenius, C. and Paasivaara, M. (2003). Collaboration practices in global inter-organizational software development projects. Software Process Improvement and Practice, 8(4), 183-199.

Litoriya, R., Sharma, N., & Kothari, A. (2012). Incorporating cost driver substitution to improve the effort using Agile COCOMO II. 2012 C.S.I. Sixth International Conference on Software Engineering (C.O.N.S.E.G.).

Low, G., & Jeffery, D. (1990). Function points in the estimation and evaluation of the software process. IEEE Transactions on Software Engineering, 16(1), 64-71.

Madachy, R. (2007). Distributed global development parametric cost modeling. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics, 4470, 159–168.

Masoudi-Sobhanzadeh, Y., Motieghader, H., & Masoudi-Nejad, A. (2019). FeatureSelect: A software for feature selection based on machine learning approaches. B.M.C. Bioinformatics, 20(1).

Mohan, K.G., Babu, J.S., Raju, V.K., Kotharu, M., Gowthami, K., & Pramod, Y. (2020). Cost Estimation Using Hybrid Algorithm. International Journal of Scientific & Technology Research, 9(2).

Mukhopadhyay, T., & Kekre, S. (1992). Software effort models for early estimation of process control applications. IEEE Transactions on Software Engineering, 18(10), 915-924.

Mustafa, E., & Osman, R. (2018). An analysis of the inclusion of environmental cost factors in software cost estimation datasets. 2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C).

Pandey, P. (2013). Analysis of the techniques for software cost estimation. 2013 Third International Conference on Advanced Computing and Communication Technologies (ACCT).

Pinkashia, S. J., & Singh. (2017). Systematic literature review on software effort estimation using machine learning approaches. International Conference on Next Generation Computing and Information Systems, Jammu, India, 11-12 December 2017, 43-47.

Pospieszny, P., Czarnacka-Chrobot, B., & Kobylinski, A. (2018). An effective approach for software project effort and duration estimation with machine learning algorithms. Journal of Systems and Software, 137, 184-196.

Qi, K., & Boehm, B.W. (2019). Process-driven incremental effort estimation. 2019 IEEE/ACM International Conference on Software and System Processes (I.C.S.S.P.).

Rahikkala, J., Hyrynsalmi, S., Leppänen, V., & Porres, I. (2018). The role of organisational phenomena in software cost estimation: A case study of supporting and hindering factors. E-Informatica Software Engineering Journal, 12, 167–198.

Rajeswari, K. (2018). A critique on software cost estimation. International Journal of Pure and Applied Mathematics. International Journal of Pure and Applied Mathematics, 118, 3851-3862.

Ramacharan, S., & Rao, K.V. (2016). Scheduling based cost estimation model: An effective empirical approach for GSD project. F.I.P. IFIP International Conference on Wireless and Optical, 1-4.

Reddy, P.V., Hari, C.V., & Srinivasa, R.T. (2011). Multi objective particle swarm optimization for software cost estimation. International Journal of Computer Applications, 32(3), 13-17.

Rhmann, W., Pandey, B., & Ansari, G.A. (2021). Software effort estimation using ensemble of hybrid search-based algorithms based on metaheuristic algorithms. Innovations in Systems and Software Engineering.

Sagar, D., Moparthi, D., & Mandhala, V. (2018). Probabilistic estimation of software development struggle techniques using machine learning. International Journal of Engineering & Technology, 7(2.7), 1085.

Citation Information: Gani, A., Akhunzada, A., & Junaid, M. (2022). Software cost estimation in global software development using hybrid approach. *Journal of Management Information and Decision Sciences, 25(S4),* 1-25.

Saljoughinejad, R., & Khatibi, V. (2018). A new optimized hybrid model based On C.O.C.O.M.O. to increase the accuracy of software cost estimation. Journal of Advances in Computer Engineering and Technology, 4, 27-40.

Sangwan, O.P. (2017). Software Effort Estimation Using Machine Learning Techniques. In 7th International Conference on Cloud Computing (pp. 92-98). Data Science & Engineering-Confluence, Noida, India, 12-13.

Sarro, F., Petrozziello, A., & Harman, M. (2016). Multi-objective software effort estimation. Proceedings of the 38th International Conference on Software Engineering.

Sewak, M., Sahay, S.K., & Rathore, H. (2018). Comparison of deep learning and the classical machine learning algorithm for the malware detection. 2018 19th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (S.N.P.D.).

Sharma, P., & Sangal, A.L. (2019). Investigating the factors which impact S.P.I. implementation initiatives in software S.M.E.s—A systematic map and review. Journal of Software: Evolution and Process, 31(7).

Sharma, S. (2017). Applications of genetic algorithm in software engineering, distributed computing and machine learning. International Journal of Computer Applications & Information Technology, 9, 208-212.

Silhavy, R. (2019). Software engineering methods in intelligent algorithms: Proceedings of 8th computer science online conference 2019. Springer.

Singh, A.J., & Kumar, M. (2020). Comparative analysis on prediction of software effort approximation using machine erudition techniques. S.S.R.N. Electronic Journal.

Suliman, S.M., & Kadoda, G. (2017). Factors that influence software project cost and schedule estimation. 2017 Sudan Conference on Computer Science and Information Technology (S.C.C.S.I.T.).

Tomás, V., Ochoa, S.F., & Perovich, D. (2017). Survey of software development effort estimation taxonomies. Technical report. Computer science department, University of Chile, Chile.

Tsung, C., Yen, C., & Wu, W. (2018). A software defined-based hybrid cloud for the design of smart micro-manufacturing system. Microsystem Technologies, 24(10), 4329-4340.

Venkataiah, V., Mohanty, R., & Nagaratna, M. (2019). Application of hybrid techniques to forecasting accurate software cost estimation. International Journal of Recent Technology and Engineering (I.J.R.T.E.), 7(6), 408-412.

Vig, V., & Kaur, A. (2018). Test effort estimation and prediction of traditional and quick release prototypes using machine learning algorithms. Journal of Intelligent & Fuzzy Systems, 35(2), 1657-1669.

Vyas, M., Bohra, A., Lamba, D. C., & Vyas, A. (2016). "A review on software cost and effort estimation techniques for agile development process. International Journal of Recent Research Aspects, 5, 612-618.

Whigham, P.A., Owen, C.A., & Macdonell, S.G. (2015). A baseline model for software effort estimation. A.C.M. Transactions on Software Engineering and Methodology, 24(3), 1-11.

Wickramaarachchi, D., & Lai, R. (2017). Effort estimation in global software development - a systematic review. Computer Science and Information Systems, 14(2), 393-421.

Woznicki, S.A., Baynes, J., Panlasigui, S., Mehaffey, M., & Neale, A. (2019). Development of a spatially complete floodplain map of the conterminous United States using haphazard forest. Science of The Total Environment, 647, 942-953.

Yeh, T., & Deng, S. (2012). Application of machine learning methods to cost estimation of product-life-cycle. International Journal of Computer Integrated Manufacturing, 25(4-5), 340-352.

Ziauddin, S., Kamal, S., Khan, S., & Nasir, J. A. (2013). A fuzzy logic based software development cost estimation model. International Journal of Software Engineering and Its Applications, 7(2), 7-18.

Received: 08-Feb-2022, Manuscript No. JMIDS-21-9014; **Editor assigned:** 10-Feb-2022; PreQC No. JMIDS-21-9014 (PQ); **Reviewed:** 24-Feb-2022, QC No. JMIDS-21-9014; **Revised:** 03-Mar-2022, Manuscript No. JMIDS-21-9014 (R); **Published:** 10-Mar-2022