International Journal of Business and Economic Affairs (IJBEA)
6(3), 118-129 (2021)
DOI: 10.24088/IJBEA-2021-63001
ISSN: 2519-9986

Software Cost Estimation in Global Software Development Business: A Review of Models and Cost Drivers for Economical Business

MEHMOOD AHMED [1], ATIF ATIQUE SIDDIQUI [2], SHIRAZ KHAN [3], MUHAMMAD JUNAID [4*]

[1,4] Department of IT, The University of Haripur, Haripur, Pakistan

[2,3] Department of Management Sciences, The University of Haripur, Haripur, Pakistan

Abstract: Global Software Development (GSD) has grown in popularity as important tool for ensuring the efficient use of resources in internationally distributed environments across multiple geographical locations for business. The most prominent features of GSD are lowering costs, speeding up growth, and gaining access to talented developers all over the world. However, there are a number of drawbacks that result from the distance between development teams including coordination and communication causing the hidden business costs involved in development process. In the context of GSD, it is necessary to focus on economic cost estimation models as estimating the needed resources and effort remains a difficult task. Software cost estimation has become a critical factor in determining software development effectiveness economically. There are many cost estimating models, including algorithmic, non-algorithmic, and hybrid for business development environment. Over the last decade, several studies have focused on economic cost estimate in GSD. To the best of our understanding, current cost estimation techniques/models do not take into account all the additional business cost drivers that are necessary to calculate accurate cost estimation in the GSD context. In this paper we present a comparative analysis of economic software cost estimation techniques along with cost divers used in GSD context. This paper summarizes the additional cost drivers in GSD and discusses open research topics in economic cost estimation in GSD based on the results review of the associated literature.

Keywords: Business development, GSD, Outsourcing, Software cost estimation, Economic cost drivers

Received: 21 January 2021 / Accepted: 19 June 2021 / Published: 29 July 2021

INTRODUCTION

GSD is a method of developing economic software in which stakeholders from various social backgrounds and locations participate in the business software development life cycle. From a newly introduced practice, this phenomenon started in the early years of this century and has developed into a broadly accepted and welcomed approach to software development. The rise of offshore software development outsourcing highlights the need to better comprehend the issues or difficulties that come with it (Aman & Nicholson, 2003). The popular software development companies strive to save time, cut costs, and improve the quality of their products (Joshi, 2018). As a result, they contract their work with a third party, such as a team, a partner, or an organization, to create components or the whole software product (Šmite & Borzovs, 2008). The percentage of projects that are internationally distributed has been constantly rising due to an expanding global marketplace, a tendency toward developing software in low-cost countries, and the growing complexity and size of software systems. (Davison, 2004). Many studies have now accepted the profound impact of globalization on software development. Business Process Outsourcing (BPO) has evolved into a natural evolution of the global market in this period (Smite, 2007). In order to thrive in a competitive market, businesses are increasingly outsourcing Information Technology (IT) services. The value of the offshore software development market has soared 25-fold over the past 10 years, according to a poll held in the United States in 2009 (Conchúir, Ågerfalk, Olsson, & Fitzgerald, 2009).

Although the globalization approach has a number of advantages that aid in the development of software products at a low cost, it also has a number of drawbacks that could stymie the success of

*Corresponding author: Muhammad Junaid
†Email: mjunaid@uoh.edu.pk

internationally distributed software development projects (Ågerfalk, Fitzgerald, Olsson, & Conchúir, 2008; Carmel, 1999; Conchúir et al., 2009; Hirschheim, Heinzl, & Dibbern, 2007; Šmite, 2006). As mentioned by Kile, Little, and Shah (2005) 60 percent of these projects failed to produce within time, budget, and desired quality, according to their research, which looked at the rate of project success in a globally distributed environment. Managing the internationally distributed environment is thus a key trait, either in and of itself, or in terms of its consequences. However, it is critical to maintain a high level of accuracy in effort estimating methods in order to effectively plan software development project activities (Shepperd, Schofield, & Kitchenham, 1996). However, there are a variety of reasons for adopting GSD that are appropriate for the purpose, but this study focuses primarily on the lower cost, which is one of the most important factors. The primary aim of globalization is to reduce the cost of development; this is seen as the primary reason for not developing locally. It can be deceiving if we don't consider the difficulties of this type of growth, such as the time zone and cultural differences (Suliman & Kadoda, 2017). As a result, if we do not consider GSD's considerations, it may take more time and effort. One of the most important problems is estimating the effort and expense in GSD (Ramasubbu & Balan, 2012) to see whether we can benefit from it or achieve importance through local production. Cost estimation can be done using a variety of methods and techniques. Because many models were created before the GSD concept, they lack the considerations and cost drivers that are associated with this growth. One of the most fundamental procedures in Software Project Management (SPM) is the identification of software cost qualities, which has been extensively discussed in the literature (Angelis, Stamelos, & Morisio, 2001; Chadli et al., 2016; Mukhopadhyay, Vicinanza, & Prietula, 1992).

Cost estimation plays an important role in supporting the budgeting, scheduling and planning activities and decisions during software development (Usman, 2018). Many cost estimation methods and techniques have been suggested for collocated environment during last decades before emergence of GSD concept. They are broadly divided into categories i.e., Algorithmic models, Artificial Intelligence and Expert estimation based techniques (Usharani, Ananth, & Velmurugan, 2016). Algorithmic methods like COCOMO (Constructive Cost Model), Putnams Model, Function Point based models and SEER-SEM Models make use of input like source line of code, function point and cost driver to perform the process of estimation by using mathematical equations. COCOMO 81 proposed by Barry Boehm in 1981 is an extensively and most widely used for cost estimation. With enhancement of new development approaches and software development lifecycles Boehm improved and published COCOMO II in 2000 based on statistical analysis of past projects data such as cost drivers with its Effort Multipliers (EMs) and Scale Factors (SFs).

GSD is much more complicated than collocated development in terms of cost analysis and calculation. (Wickramaarachchi & Lai, 2017) mentioned that many models were created before the GSD concept but they lack the considerations and cost drivers that are associated with this growth. The models proposed for GSD are amplified from the collocated cost estimation techniques. This paper also investigates software cost factors that have been used in the literature to address software cost estimation related to the management of GSD projects. The goal of this research is to make a comparative study to identify cost estimation issues in GSD. A Literature Review focusing on the methods faced by project managers of GSD projects was used to compile the list of software cost characteristics. We plan to do so by addressing the following Research Question.

RQ: In the context of GSD, what are the state of the art cost estimation models and additional cost drivers?

The following is a breakdown of the paper's structure. Section 2 explains the cost estimation techniques in GSD contexts. Section 3 highlight the cost drivers impact GSD cost estimation. Section 4 presents the finding of this study.

LITERATURE REVIEW

There has been a lot of work done in the area of cost estimating in a collocated environment, but there has only been a small amount of work done in the area of GSD cost estimation. GSD cost estimate differs from cost estimating in a collocated setting because GSD includes extra cost variables. The bulk of software cost estimating methods used in business are based on expert judgement (Azzeh, 2012). Because of the growing popularity of GSD in recent decades, researchers must concentrate on cost estimation in GSD. The most recent contribution in this subject is reviewed in this publication. Based on the development method, the selected models may be classified into three groups. Fig. 1 depicts research works in the realm of cost estimate in the context of GSD.

Amplification of The COCOMO II

Stefanie's art was first published in 2007. (Betz, 2008) developed a COCOMO II adaptation for estimating the effort of internationally distributed projects, as well as additional effort multipliers to make it more effective. Because this model is an expansion of COCOMO II, it has three stages: early prototyping, early design, and post architectural design. This expansion focuses on Post Architectural, which is a popular and precise technique.

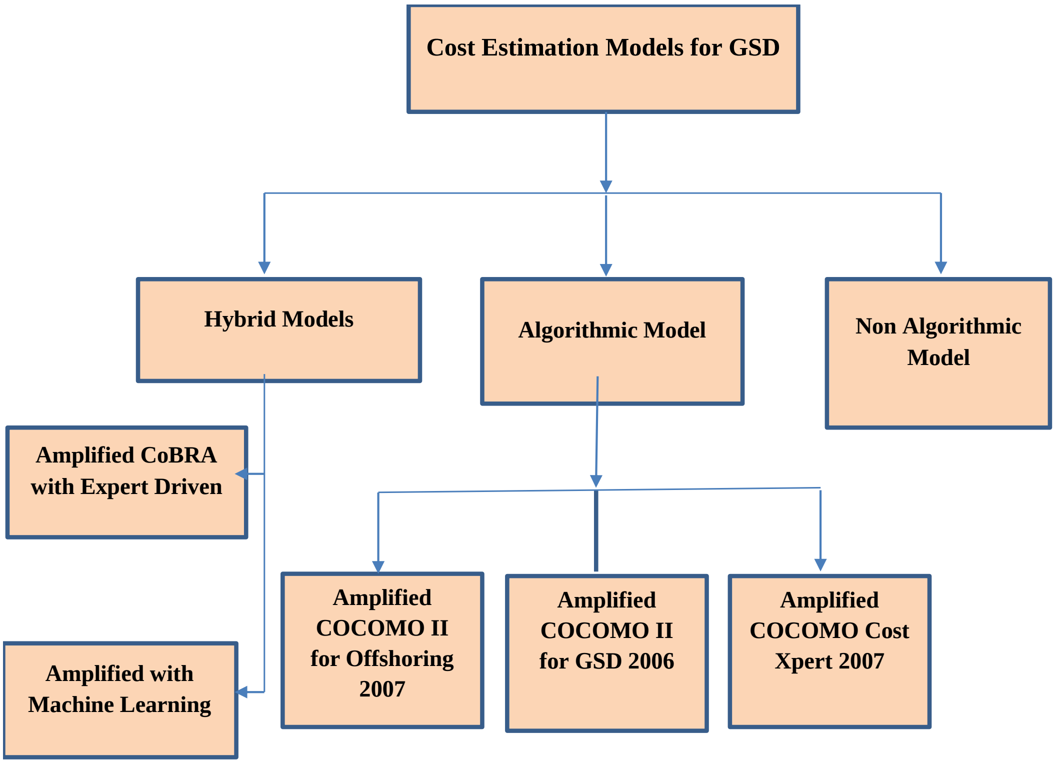

Figure 1. Research studies depicting GSD

A: Constant (2.94), Size: KLOC, E: Scale factors & EM Effort Multiplier COCOMO II modification consists of three phases, the first of which is the discovery of additional cost drivers. Several variables (Effort Multiplier and Scale) were discovered to be impacted by offshore software development through research and qualitative surveys. The second stage is categorising and quantifying these elements into four groups. The classification was based on theoretical considerations, literature, and expert views. This model is more extensive than the previous one, but it has certain drawbacks, such as the lack of a systematic approach to factor quantification, the limiting of collocation between two firms, and the fact that it has yet to be evaluated.

Estimating effort in GSD projects using machine learning techniques

Machine Learning is a novel addition to researchers' toolkits, particularly for experimenting with new techniques since machine learning approaches provide more precise effort estimation. Humayun and Gang (2012) released a paper that included an overview of several machine learning approaches for cost estimate in GSD. Artificial Neural Network (ANN) is a computer or mathematical model that is stimulated by the biological human brain, and is one of the most often utilised approaches. This can be set up for a specific purpose, such as pattern recognition or data classification via learning. In a different technique, neurons are employed in the creation of a Feed-Forward Neural Network (FFNN), and these neurons are connected to each other through a specific network design. The FFNN's primary objective is to convert inputs into meaningful outputs. The primary distinction between RBNN and FFNN is that the radial basis layer of RBNN is positioned in the same place as the Hidden layer in FFNN design. Another machine learning approach is Case-Based Reasoning (CBR), which is a four-stage cyclic procedure. Retrieval of instances that are comparable ii). Using the previously retrieved cases to solve the problem iii). If required, revise the suggested solution; and iv). Retaining the solution in order to create a new case.

The comparison in this paper is between three widely used machine learning algorithms, namely ANN, CBR, and RI. A total of 81 software projects from a Canadian software firm in the late 1980s were included in this comparison. This research shows that, depending on the context, existing methods can be utilised to account for GSD features. This model is extremely reliant on the data given, which can lead to incorrect estimates because accuracy is largely reliant on the data used for training. In the context of GSD, cost drivers/variables are increasingly significant, yet models do not address cost drivers.

Software cost estimation based on use case points for GSD

Relativity In GSD, there is a quick and straightforward method for estimating costs. Azzeh (2013) published a paper on the Use Case Point (UCP) method. Analyzes the possibilities of the Use Case Point estimate model for worldwide projects and utilises it as a springboard to explore three recommended elements (global team trust, global team composition, and culture value) that will aid in managing the development of global software projects. The goal of this article is to expand the widely used Use Case Points UCP paradigm to enable outsourced projects. The UCP model is a frequently used method for estimating project work at an early stage. UCP is a relatively new and straightforward method. This method is based on the Use Case Diagram and assists project managers in determining the size of a software application at an early stage. Six stages are involved in calculating use case points.

Step 1

In a use case diagram, identify and classify the different types of actors and use cases. This will aid in determining the size of the system based on its complexity.

Actors are categorised into three groups:

1. Easy to understand (Another system).
2. Average (Actor interacting through protocols such as TCP/IP, FTP, and so on)
3. Complicated (person interacting through GUI).

Step 2

Weighting factors is assigned to each factor including Simple as (1), Average as (2) and Complex as (3).

Step 3

Calculate Unadjusted Weighted Actors (UWA) by multiplying the quantity of each actor type by its appropriate weight and then combining the results.

UWA = (nS + 2 x nA + 3 x nC)

The number of each actor type is multiplied by its appropriate weight, and then these numbers are added to get UWA, where nS stands for simple, nA for average, and nC for complex actor.

Step 4

Unadjusted Use Case Count (UUC) is determined in the same way as UWA; each use case is categorised into one of three kinds based on the number of transactions involved: simple, average, or complicated. Simple use cases have no more than three transactions; typical use cases have four to seven transactions; and complicated use cases include more than seven transitions. Simple (5), Average (10) and Complex (50) are the complexity weights (15).

Step 5

Unadjusted Use Case Count is determined by multiplying the number of each use case type by its weight, then summing the results:

UUC = (5 x suc + 10 x auc + 15 x nuc)

The Unadjusted Use Case Point UUCP is calculated by adding UWA and UUC

UUCP: UWA + UUC

Step 6

In order to improve the accuracy of the final size estimate, the UCP technique employs two types of correction variables. Technical Complexity Adjustment Factors (TCAF) and Environmental Factors are the two types of factors (EF). There are 13 factors of technical complexity and eight aspects of environmental complexity.

TCAF: 0.6 + (0.01 x TF)

Points for the reworked use case The unadjusted use case count UUCP is multiplied by TCAF and EF to get the UPC:

UPC = UUCP x TCAF x EF

This study examines the potential of using additional environmental elements in the final calculation to assist globalisation.

- Global Team Trust (GTT)
- Global Team Composition (GTC)
- Culture value

There is no technical knowledge required for this method, and it is typically utilised in environments where the application size is approximated by the UCP. Use cases are necessary to ensure precise estimating information. This model has yet to be tested.

Analogy-based software development effort estimation in GSD

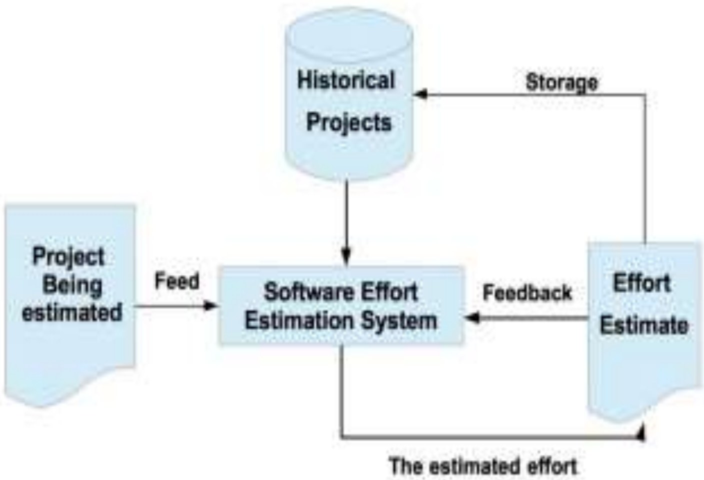

Figure 2. Framework for the effort estimation system (El Bajta, 2015)

El Bajta (2015) analogy-based cost estimating methodology employs a case-based reasoning approach to predict costs for GSD projects of comparable kind. This methodology is highly effective if you've done comparable nature projects before, but it won't work if you don't have any experience with similar nature projects. A new method for using analogy-based reasoning to improve cost estimating efficiency in distributed or mixed software projects that deal with numerical and categorical data. The suggested technique will be empirically validated using the following dataset from the International Software Benchmarking Standards Group.

Steps involved in analogy based estimation:
- Choosing the right analogy
- Investigating similarities and differences
- Examining analogy quality
- Providing the estimation

Steps of the proposed model

1. Relevant attributes: To begin, we must define a collection of qualities that are both relevant and independent.
   Similarity function: Determine the degree of similarity between the candidate project and each database project. There are two widely used similarity functions:
   - Euclidean similarity (ES)
   - Manhattan similarity (MS)
2. Performance evaluation: The following high-level needs were established as a first step in creating the model:
   - Calculation of the input cases' complexity measure.
   - Prediction based on analogy, which employs a variety of similarity metrics.
   - Keeping the database up to date as new instances come in.
   - Modification or deletion of a specific record.

The procedure for adding new similarity measures and extra parameters has been updated.

Cost estimation for GSD

Keil, Paulish, and Sangwan (2006) proposed a COCOMO II-based model as a decision-making framework for calculating the tradeoff between data and GSD estimate.

Factors relating to multisite communication and coordination are included in the model.
- Factors affecting the product: (Precedentedness, Architectural Adequacy).
- Factors affecting personnel: (Cultural Fit, Skill Level, Shared Understanding, Information Sharing Constraints).
- Factors affecting the project: (Novelty of Collaboration Model, Tools and Infrastructure, Physical Distance).

The lack of a systematic method to factor extraction is a flaw in this paradigm. Furthermore, the model has yet to be tested.

Software Outsourcing Cost Estimation Model (SOCEM)

(Ahmad, Khan, & Qasim, 2018) sought to identify and address the problems faced by the vendor organisation in GSD using a systematic literature review protocol. This is not a generic model, but rather one that focuses on organisation. Without GSD-based cost drivers, this is not a comprehensive model. This model is still being tested.

Parametric of global development that is distributed cost modeling

(Madachy, 2007) introduced a new technique called Cost Xpert, which uses COCOMO II to model costs in a distributed setting. Because the characteristics of scattered locations in a GSD project may change, this model provides effort multipliers for each site. The task allocation in Cost Xpert differs from the other models in that it is based on phase rather than module or specific function. It allows several teams in a project to have their own calendars. To better estimate internationally distributed projects, this model was created with the help of the University of Southern California (USC) Centre for Systems and Software Engineering. Using phase-sensitive effort multipliers, the novel approach extends standard cost estimating formulae for dispersed teams. The distribution of software work by phase per team may be used to describe a project. For a more thorough and precise assessment, the unique qualities of each team are also included in the computations. To capture the variation owing to varied team characteristics by phase, the model uses the phase-sensitivity of effort multipliers. This approach is appealing because it takes into account cost elements linked to individuals in different teams (labour calendars, labour categories, and rates in local currencies) at various stages of the computations. However, collaboration and coordination, which have a significant influence on the global environment, were not taken into account in the calculation, and no extra effort drivers were included.

Estimating the effort overhead in GSD

The Cost Overhead Model for GSD (CoBRA) (Lamersdorf, Münch, Fernandez-del Viso Torre, Sánchez, & Rombach, 2010) is based on causal relationships and influencing variables. This causal link can aid in the comprehension of the elements' significance and relationships. This is a company-specific model, not a generic one. The Goal Question Metrics Paradigm is used to evaluate this model.

Scheduling based cost estimation model

Ramacharan and Rao (2016) concentrates on scheduling and productivity factors, including line-by-line cost estimation. This model is compared to other models, which is a unique feature because all prior models lacked this type of evaluation.

There appear to be several elements that influence the price of GSD. Geographic and temporal distance, variations in language and culture, social variables, issues originating from organisational structure, procedures, and projects, and infrastructure and product architectural obstacles are among the most relevant considerations. The majority of cost considerations associated with GSD have a negative influence on productivity. The variables may also interact, raising the risk of a detrimental influence on the development process. When a team is located in the same physical place, geographic distance has a substantial detrimental influence on informal ad hoc coordination, communication, and interaction. These are essential for effective software development since developing a shared knowledge of the programme under development necessitates a great deal of collaboration, communication, and interaction. Even a relatively short physical separation, such as when team members are in separate buildings, has an influence on the physical separation. The possibility of synchronous communication, which is a crucial communicational quality for real-time problem solving and design activities, is reduced by temporal distance. The temporal distance has a U-shaped influence on productivity, and the greatest production speed is when there is no temporal distance or no overlap between the teams. Temporal remoteness is also having an increasingly detrimental influence on communication accuracy.

Cultural and linguistic differences have a larger influence since they have been identified as important obstacles to communication. These distinctions can result in significant and long-term misconceptions and communication issues. Differences in culture and language also increase the amount of time and effort required for communication. There are two types of cultural issues: organisational and national culture. As a result, issues might emerge even inside a single country. Fear and distrust can have a detrimental influence on motivation, trust, cooperation, communication, and knowledge sharing with remote colleagues, and hence have a direct impact on the success of adopting GSD. These social issues are exacerbated by geographic, temporal, and cultural isolation. Outsourcing, cooperation, and partnership arrangements make these concerns much more complicated. When, by whom, and how should activities and tasks be completed and assigned are all organisational, process, and project concerns. The most productive teams are said to be those that are colocated and cohesive. Within the corporation, the structure also provides communication limits and barriers. Due to misconceptions, integration, and interoperability issues, different project management methods and processes might result in rework or data loss. Communication obstacles are created by infrastructure configuration and disparities, which must be addressed in order to allow information transfer amongst interdependent team members. Inadequate knowledge management infrastructure might prevent project teams from developing a shared understanding. The lack of efficient information exchange channels and poorly kept documentation raises the risk of knowledge management challenges, which can result in decreased productivity, quality, and other concerns.

Table 1: Review matrix of existing GSD specific cost estimation models

‘ Paper Title	Researcher Name	Cost Estimation Model	Technique	Main Idea	Limitation & Future Work
Analogy-based software development effort estimation in global software development	El Bajta (2015)	Analogy Based Model	Similarity Functions	Software development effort estimation is based on an enhanced analogy-based model using similarity functions.	Consideration of Risks and Uncertainties
Amplification of the COCOMO II regarding offshore software projects. Offshoring of software development	Betz (2008)	Extended COCOMO II	Effort Multiplier	Extend COCOMO II to estimate the additional sources of cost and effort for the globally distributed projects.	1. No systematic approach is used for the quantification of the factors 2. Furthermore this model limits the collocation between two companies. 3.Model is yet to evaluate
Software cost estimation based on use case points for global software development	Azzeh (2013)	Use Case Point	Weight Values	Examined the Use Case Point estimation model and discussed three additional factors (Global team composition, Global team trust and Culture value)	1. To ensure accurate estimation details, use cases are required. 2. This model is yet to evaluate
Scheduling based cost estimation model: An effective empirical approach for GSD project	Ramacharan and Rao (2016)	Scheduling based Estimation	Calibration parameters	Primary study presented the consequences of intercultural factors.	Combining the findings of committees of automatically produced experts may work better in noisy or uncertain environments than depending on a single expert.
Cost estimation for global software development	Keil et al. (2006)	Extended COCOMO II	Effort Multiplier	Model provided factors related to multisite communication and multisite coordination	No systematic approach for factor extraction. No quantification of complexity factors. The model is not yet evaluated
Estimating the effort overhead in global software development	Lamersdorf et al. (2010)	Effort Overhead Estimation	Causal Model	Presented specifically designed model for developing individual cost models using expert estimations and project data.	1. Could not be used for context specific environment. 2. Model focuses on the rankings of the chosen variables by interviewees. The majority of the quantifications have a high Std. Dev. 3. Study is exclusive ro single remote location ventures. 4. Model not validated. 5. GSD Factors (Time & currency differences, coordination) not considered.
Distributed global development parametric cost modeling	Madachy (2007)	Cost Xpert-2007	Phase-sensitive Effort Multiplier	Cost Xpert an extended COCOMO II for distributed environments. Unlike the other models, Work allocation in Cost Xpert based on phase rather than module or specific function.	1. Model might not be appropriate here work allocations are based on modules. 2. Cost drivers (collaboration and coordination) not considered. 3. Not is not validated and quantified. 5. No additional cost drivers by COCOMO II.
Software Outsourcing Cost Estimation Model (SOCEM). A Systematic Literature Review Protocol	Ahmad et al. (2018)	Software Outsourcing Cost Estimation		Presented model to identify the challenges faced by the vendor organization regarding cost estimation	An abstract level estimation model No consideration on cost drivers Not generalized model it is specific to an organization
Estimating effort in global software development projects using machine learning techniques	Humayun and Gang (2012)	Machine Learning		Overview of different machines learning techniques for effort estimation have been presented.	No consideration on cost drivers Accuracy of estimation highly dependent of data used for training purpose there is more chance of inaccuracy .

Table 2: Identified cost drivers of cost estimation (GSD)

‘ Sr.	Cost Driver	Impact of Cost	Sr.	Cost Driver	Impact of Cost
1.	Time zone difference	Critical	22	Development Productivity	Low Significant
2	Language and Cultural Difference	Critical	23	Defect Density	Low Significant
3	Communication Infrastructure and Process	Critical	24	Rework	Moderate
4	Process Model	Moderate	25	Project Management Effort	Critical
5	Travel Cost	Moderate	26	Reuse	Moderate
6	Competence Level	Low Significant	27	Code Side	Low Significant
7	Requirement Legibility	Moderate	28	Product Complexity	Low Significant
8	Process Compliance	Critical	29	Platform Volatility	Low Significant
9	Response delay	Moderate	30	Task Allocation	Moderate
10	Team Trust	Critical	31	Geographic Distance	Critical
11	Client Unawareness	Low Significant	32	Social Factors	Low Significant
12	Shared Resources	Moderate	33	Product Architecture	Low Significant
13	Team Structure	Low Significant	34	Unviability of Concerned Personnel	Low Significant
14	Work Dispersion	Low Significant	35	Exchange Rate Fluctuation	Low Significant
15	Work Pressure	Moderate	36	Unrealistic Milestones	Low Significant
16	Range of Parallel Sequential Work Handover	Low Significant	37	Training Session/Meeting	Low Significant
17	Client Specific Knowledge	Low Significant	38	Rules/Laws	Low Significant
18	Lack of client involvement	Moderate	39	Process maturity	Moderate
19	Design and Technology newness	Moderate	40	Organizational Difference	Low Significant
20	Team size	Moderate	41	Overoptimism	Low Significant
21	Project Effort	Moderate			

Source: Khan, Khan, Iqbal, and Rehman (2021)

Table 3: Mapping of critical & moderate cost drivers of cost estimation (GSD)

‘ Sr.	Cost Estimation Model (GSD) Reference S. Table II	Critical Cost Driver Reference Table III	Moderate Cost Driver Reference S. Table III	Remarks
1	22	Nil	Nil	
2	31	Nil	4 , 7	
3	30	Nil	Nil	
4	34	1 , 2 , 31	Nil	
5	24	Nil	Nil	
6	25	2 , 3 , 8	15 , 39	
7	23	Nil	Nil	
8	32	Nil	Nil	
9	33	Nil	Nil	

The work required to coordinate the development process is influenced by the software architecture and modularity. It's tough to manage dependencies among developers working on the same modules in GSD since communication across locations is severely hampered. When organisational design and task allocation are aligned with software architecture, it helps to reduce the need for communication and coordination between geographical teams, which boosts productivity. The cost variables for global software development described in the research literature do not include cost elements that arise outside of software engineering. Set-up factors (contracting to a foreign country, establishing an office in a foreign country), illegal (corruption in many developing nations), and international trade-related factors are examples of these (export and import taxes and tariffs). These considerations might potentially have a big influence on the total cost of a software development project.

Cost drivers are elements that have a multi-dimensional impact on software development, which can be favourable or negative. COCOMO II models calculate costs using 5 scale factors and 17 effort multipliers, and are suitable for collocated projects. Because of the dispersed nature of this development, these cost drivers are concealed in nature in GSD. Because of GSD's scattered structure, these cost drivers are frequently overlooked, resulting in cost overruns later in the project. For a reasonable forecast of effort and resources, these elements should be addressed throughout the estimating process.

Khan et al. (2021) presented a systematic strategy for identifying new cost drivers by evaluating related activities and testing alternative cost estimating techniques. For the GSD cost calculation provided in Table 3, Khan et al. (2021) found 41 cost factors. Based on the frequency in the articles, cost drivers are classified as having low relevance (24), moderate (16), or crucial (8). According to related studies (Betz, 2008; Kile et al., 2005; Lamersdorf et al., 2010; Madachy, 2007; Prikladnicki, Audy, & Evaristo, 2006; Muhairat, Aldaajeh, & Al-Qutaish, 2010), only 22.2 percent of studies focus on critical cost drivers (Language and cultural differences, Process competence, Communication infrastructure, and time zone differences), while 22.2 percent focus on moderate cost drivers (Process model, Response delay, work pressure, and design and technology newnes).

Summary of the findings

As stated in the introduction, cost estimate is important for project budgeting, scheduling, and planning, as well as making decisions throughout software development. Even if the ultimate aim of GSD is cost-effective software development, it is still unable to satisfy expectations due to the following:

1. There has been little study in the field of GSD cost estimate, and most of the presented models are unsuitable for application due to a lack of calibration.
2. However, no matter whatever model we employ, we are not obtaining the expected outcomes since present models only target a small number of extra cost drivers and do not account for all of the potential cost drivers in a GSD project.
3. Because to the nature of GSD, proposing a model that can be utilised all over the world is challenging, if not impossible. Because GSD circumstances are so diverse, what works in one may not work in another. As a result, there is no such thing as a one-size-fits-all solution, and it will help no one. As a result, a model that can be customised depending on the criteria that are essential to the domain where the programme is created is required. As a result, the most widely utilised technology for global software development must be chosen, taking into consideration the GSD cost factors.

CONCLUSION

The study included a review of cost estimates in GSD business as well as a thorough discussion of the proposed models. The relevance, advantages, and problems are addressed in a comparative research. The methodologies employed, dataset utilised for validation, constraints in business, economical cost drivers used, and so on have all been explored in the literature. Reduced manufacturing costs of business are one of the primary causes for GSD. According to the literature evaluation, however this study claims that there are no models to estimate the real costs of business. As a result, greater management overhead, more time to build teamwork, travel expenditures, and trip limitations to cut costs may result in difficulties such as loss of trust, which can lead to a drop in efficiency. Finally, this study found that the GSD context lacks formal cost estimate models and requires a formal model that takes into account all GSD economical cost drivers, since the proposed formal models are still in the early stages of development and cannot be used by business industry.

REFERENCES

Ågerfalk, P. J., Fitzgerald, B., Olsson, H. H., & Conchúir, E. Ó. (2008). Benefits of global software development: the known and unknown. In *International Conference on Software Process,* Berlin, Heidelberg.

Ahmad, J., Khan, A. W., & Qasim, I. (2018). Software Outsourcing Cost Estimation Model (SOCEM). A systematic literature review protocol. *University of Sindh Journal of Information and Communication Technology*, *2*(1), 25-30.

Aman, A., & Nicholson, B. (2003). The process of offshore software development. In *Organizational information systems in the context of globalization.* Boston, MA: Springer.

Angelis, L., Stamelos, I., & Morisio, M. (2001). Building a software cost estimation model based on categorical data. In *Proceedings Seventh International Software Metrics Symposium,* London, UK.

Azzeh, M. (2012). A replicated assessment and comparison of adaptation techniques for analogy-based effort estimation. *Empirical Software Engineering*, *17*(1), 90-127. doi:https://doi.org/10.1007/s10664-011-9176-6

Azzeh, M. (2013). Software cost estimation based on use case points for global software development. In *5th International Conference on Computer Science and Information Technology,* Amman, Jordan.

Betz, a. M. J., S. (2008). Amplification of the cocomo ii regarding offshore software projects. In *Offshoring of software development.* Munich, Germany: KIT Scientific Publishing.

Carmel, E. (1999). *Global software teams: Collaborating across borders and time zones.* Hoboken, NJ: Prentice Hall.

Chadli, S. Y., Idri, A., Ros, J. N., Fernández-Alemán, J. L., de Gea, J. M. C., & Toval, A. (2016). Software project management tools in global software development: A systematic mapping study. *Springer Plus*, *5*(1), 1–38. doi:https://doi.org/10.1186/s40064-016-3670-7

Conchúir, E. Ó., Ågerfalk, P. J., Olsson, H. H., & Fitzgerald, B. (2009). Global software development: Where are the benefits? *Communications of the ACM*, *52*(8), 127-131.

Davison, D. (2004). *Offshore outsourcing: Market overview* (Tech. Rep.). Westborough, MA: META spectrum Inc.

El Bajta, M. (2015). Analogy-based software development effort estimation in global software development. In *10th International Conference on Global Software Engineering Workshops,* Ciudad Real, Spain (pp. 51–54).

Hirschheim, R., Heinzl, A., & Dibbern, J. (2007). *Information systems outsourcing: Enduring themes, new perspectives and global challenges.* Berlin, Germany: Springer Science & Business Media.

Humayun, M., & Gang, C. (2012). Estimating effort in global software development projects using machine learning techniques. *International Journal of Information and Education Technology*, *2*(3), 208-214.

Joshi, B. R. (2018). Factors affecting the software developers performance. *Journal of Administrative and Business Studies*, *4*(6), 286–294. doi:https://doi.org//10.20474/jabs-4.6.3

Keil, P., Paulish, D. J., & Sangwan, R. S. (2006). Cost estimation for global software development. In *Proceedings of the 2006 international workshop on economics driven software engineering research.*

Khan, J. A., Khan, S. U. R., 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. doi:https://doi.org//10.1109/ACCESS.2021.3055858

Kile, J. F., Little, D., & Shah, S. (2005). *The importance of effective requirements management in offshore software development projects* (Unpublihed doctoral dissertation). Pace University, New York, NY.

Lamersdorf, A., Münch, J., Fernandez-del Viso Torre, A., Sánchez, C. R., & Rombach, D. (2010). Estimating the effort overhead in global software development. In *International Conference on*

Global Software Engineering, Princeton, NJ.

Madachy, R. (2007). Distributed global development parametric cost modeling. In *International Conference on Software Process,* Berlin, Germany.

Muhairat, M., Aldaajeh, S., & Al-Qutaish, R. E. (2010). The impact of global software development factors on effort estimation methods. *European Journal of Scientific Research*, *46*(2), 221-232.

Mukhopadhyay, T., Vicinanza, S. S., & Prietula, M. J. (1992). Examining the feasibility of a case-based reasoning model for software effort estimation. *MIS Quarterly*, 155-171. doi:https://doi.org/10.2307/249573

Prikladnicki, R., Audy, J. L. N., & Evaristo, R. (2006). A reference model for global software development: findings from a case study. In *International Conference on Global Software Engineering (ICGSE'06),* Florianopolis, Brazil.

Ramacharan, S., & Rao, K. V. G. (2016). Scheduling based cost estimation model: An effective empirical approach for GSD project. In *Thirteenth International Conference on Wireless and Optical Communications Networks (WOCN),* Hyderabad, India.

Ramasubbu, N., & Balan, R. K. (2012). Overcoming the challenges in cost estimation for distributed software projects. In *34th International Conference on Software Engineering (ICSE),* Zurich, Switzerland.

Shepperd, M., Schofield, C., & Kitchenham, B. (1996). Effort estimation using analogy. In *Proceedings of IEEE 18th International Conference on Software Engineering,* Berlin, Germany.

Šmite, D. (2006). Global software development projects in one of the biggest companies in Latvia: Is geographical distribution a problem? *Software Process: Improvement and Practice*, *11*(1), 61-76. doi:https://doi.org/10.1002/spip.252

Smite, D. (2007). Project outcome predictions: Risk barometer based on historical data. In *International Conference on Global Software Engineering (ICGSE 2007),* Munich, Germany.

Šmite, D., & Borzovs, J. (2008). Managing uncertainty in globally distributed software development projects. *University of Latvia, Computer Science and Information Technologies*, *733*, 9–23.

Suliman, S. M. A., & Kadoda, G. (2017). Factors that influence software project cost and schedule estimation. In *Sudan Conference on Computer Science and Information Technology (SCCSIT),* Elnihood, Sudan.

Usharani, K., Ananth, V. V., & Velmurugan, D. (2016). A survey on software effort estimation. In *International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT),* Chennai, India (pp. 505–509).

Usman, M. (2018). *Improving expert estimation of software development effort in agile contexts* (Unpublished doctoral dissertation). Blekinge Institute of Technology, Karlskrona, Sweden.

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