VFAST Transactions on Software Engineering
https://vfast.org/journals/index.php/VTSE@ 2024, ISSN(e): 2309-3978, ISSN(p): 2411-6246
Volume 12, Number 4, October-December 2024 pp: 267-281

Investigating Effort Estimation Techniques for Mobile Applications: An Efficient Approach

Mohammad Ayub Latif [1*], Muhammad Khalid Khan [1], Saad Akbar [2], Usman Khan [1]

[1]College of Computing and Information Sciences, Karachi Institute of Economics and Technology, Karachi, 75190, Pakistan; [2]Department of Computing, Faculty of Engineering Science and Technology, Hamdard University, Karachi, 75540, Pakistan

Keywords: *Software Effort Estimation, Mobile Applications, Software Estimation for Mobile Applications.*

Journal Info:
Submitted: December 01, 2024
Accepted: December 20, 2024
Published: December 31, 2024

Abstract Estimating the effort of mobile applications is essential because many of the applications are now working on mobile platforms. A need exists to understand the difference between Effort Estimation for mobile applications and other computer applications. The last decade has seen a revolution in the use of mobile applications, which has caused in an exponential increase in the total number of mobile phone users worldwide. The first objective of this work is related to the software industry, and that is to identify which techniques are used for calculating the effort of mobile applications. The first objective also dwells into the identification of the accuracy that was achieved by using those techniques. The second objective is to propose an efficient approach for the effort estimation of mobile applications. A 5+1 methodology is suggested which should be accommodated when proposing a model for the effort estimation of mobile applications. The proposed methodology is validated through intensive investigation of the literature and it is believed that if this 5+1 methodology is adopted, the proposed model will surely bring excellent results in terms of accuracy of the predicted effort that the proposed model will attain. A small case study is also mentioned as a starting point for the validation of the 5+1 proposed methodology and it shows how the methodology can be utilized for the effort estimation of a simple mobile application.

***Correspondence author email address:** malatif@kiet.edu.pk
DOI: 10.21015/vtse.v12i4.2018

1 Introduction

Smartphones and the use of mobile applications have revolutionized the world. Looking at some figures in the year 2024, it is known that 88 percent of the time spent by a mobile user is on a mobile application. It is expected that over 935 billion USDs will be generated in revenue in the year 2024 through mobile applications.

Almost 1.96 million applications are available for download at the Apple Application store, and the mobile applications available for download at Google Application store exceeds 2.87 million. The application opening figures in a day, range from 10 to 50 times a day for different smartphone users. Almost 70 percent of US digital media time comes from mobile applications [1]. These statistics clearly indicate that mobile application development will increase in the coming years. Just like the development of software, there is a pressing need for the development of models that are specific for calculating effort of mobile applications. Many mobile applications are also created for visually impaired users, a Systematic Literature Review (SLR) identifies and review sixty such application to give directions for future for such mobile applications[2]. Researchers have already started working in this direction to formulate effort estimation models for mobile applications. This work is also an attempt to investigate all the work that has been carried out in the domain of effort estimation for mobile applications.

The two major categories of effort estimation models are algorithmic and non-algorithm. Table 1 below classifies the popular algorithmic and non-algorithmic techniques and briefly defines them in a sentence or two.

Although it won't be wrong to state that a third category of models for software effort estimation exist as machine learning based models. These machine learning based models are classified under the category of computational intelligence in some research work [3], [4]. In fact a recent work shows the use of deep learning in smell detection for mobile applications, which shows enhancement of user interfaces of the mobile applications and also helps the developers in fixing up the bugs [5].

The existing literature on effort estimation for mobile applications mainly focus on the agile development methodology. This work is not limited to the agile methodology as it is likely that any software development approaches can be accommodated for the development of mobile applications. The Opportunities and Research Directions section helps in identifying the open areas and provides a very smooth direction to the researchers who are willing to work in this domain. Three literature reviews were identified in the domain of effort estimation for mobile applications. They are worth mentioning in this work because they provide future directions for the effort estimation of mobile applications. In the first study, very few studies were identified as the selection criteria for the studies were two different SLRs from which these studies were extracted. However, this approach is not as per the suggestions of Kitchenhem [6]. The total number of selected studies in this Literature Review is 5 [7]. The second Literature review talks about effort estimation in the traditional and agile-based mobile application development and testing. This review accommodates both the methodologies and also includes testing [8].The third literature review is specific to the effort estimation of mobile applications domain, and was again confined to agile methodology, just like the first one. Another significant difference in the third study is that the research questions were answered through interviewing [9]. Table 2 depicts all the studies identified in this domain.

Table 2 shows that from the three previous literature reviews on effort estimation of mobile applications, only two used the systematic approach. Of the two systematic reviews, the work of Mendes et al. [7] is a merger of two SLRs and therefore cannot be included officially in the review list. This only leaves the work of Kaur et al. which follows the systematic approach but their work focused on agile methodology and also included test effort. The time frame of Kaur et al. [8] is till 2018 and as it is 2024 this prompts towards identifying the changes that have taken place in the investigated area of effort estimation for mobile applications.

Considering these shortcomings, this work is the latest in terms of investigations and this work is also open to all methodologies that can be used to develop mobile applications. The core focus on this research work is the effort estimation confining to mobile applications. The practices that can be used in effort estimation for mobile applications is explored in depth to fill the research gap about the specific techniques of effort estimation for mobile applications. Discussing the overview of the concepts, this current study makes the following contributions.

Table 1. Software Cost Estimation Models

Model Type	Description
Algorithmic Models	
Lines of Code (LOC)	Uses the basic LOC for major functions of the software being developed. After identifying the overall LOC of the software, it uses historical data to predict the cost and schedule of the software which is undergoing development.
Function Points (FP)	A model that uses five information domains (e.g., number of inputs, number of outputs) and assigns weights to calculate a total. It applies adjustment factors to convert function points into meaningful estimates using historical data.
COCOMO 81	The Constructive Cost Model (COCOMO) by Barry Boehm offers three modes: organic, semi-detached, and embedded. It also provides three levels of estimation choices that range from size-only to more detailed factors.
Non-Algorithmic Models	
Expert Judgment	Estimates cost using recommendations from specialists with prior experience in similar projects. Also known as expert opinion.
Analogy-Based Techniques	Attributes from a similar past project are used to estimate the cost of a new project by referencing historical data archives.
Proxy Techniques	Uses simpler proxy measures related to the target quantity for estimation. Proxies are easier to count than the actual measures.
Pricing to Win	Cost is determined based on the amount the customer is willing to pay comfortably.

Table 2. Comparative Summary of Existing Surveys

Ref.	Survey Type	Year	No. of Papers	Coverage
[7]	Systematic	2014	5	This study is a merger of two different SLRs, therefore two different coverages are shown: Global Software Development SLR (2001-13) and Effort Estimation in Agile Development (2001-13).
[8]	Systematic	2018	21	2008-18
[9]	Interviewing	2019	Interview of 20 experts from 18 different organizations.	No specific year, but the interviews were conducted before 2019 as the paper was published in 2019.

I. Investigation of the methods/techniques that are
II. Investigation of the metrics that have been used to measure the accuracy of effort estimation methods/techniques in Mobile Applications.
III. The investigation of the datasets that are used by the researchers/practitioners for estimating the effort of mobile applications.
IV. Identification of a 5+1 methodology before proposing a framework or a model for the effort estimation of mobile applications. As starting point a simple case study is also mentioned that shows how the 5+1 methodology can be accommodated for effort estimation of a mobile application.
V. Identification of the future directions for effort estimation of mobile applications.

The remaining paper is structured as follows: Section 2 provides the background of effort estimation of mobile applications. Section 3 goes into the investigations and the results that were extracted from those investigations for the effort estimation of mobile applications. Section 4 discusses the future directions of the research work that has been investigated, which provides an eye opener for the future work in effort estimation of mobile applications. Section 5 presents opportunities, research directions along with a 5+1 methodology that has been proposed for creating a model or a technique for the effort estimation of mobile applications. It is believed that this 5+1 methodology will lay the foundation for future models or techniques proposed for effort estimation of mobile applications. Finally, the conclusion section along with the future directions comes under section 6.

2 Background

This section talks about some important terms which are helpful in understanding the concept of software effort estimation and also introduces the concept of effort estimation for mobile applications. A table is provided which discusses the pre-requisites for knowing the complete effort estimation process. These prerequisites are commonly used in the relevant literature for industrial models or by the practitioners. The terms discussed in Table 3 are also used by the mature software process models, which will help the readers better understand all the terms used in effort estimation of software. Although the terms introduced in Table 3 are related to software effort estimation and not specific to mobile applications, but as mobile applications are also software, therefore, they will also have their applicability in the mobile applications domain.

2.1 Effort Estimation of Mobile Applications

As the number of mobile application users is increasing, it will not be wrong to claim that in near future, all computer-based activities will be performed on mobile applications, so it is believed that computer-based applications will be obsolete at that point of time. It is expected that the number of mobile application users will pass 250 million in 2015 [15]. This trend will keep increasing at an exponential speed, and, naturally, this will pace up the domain of mobile application development. Researchers and practitioners are actively working on devising techniques that will confine to the effort estimation area of mobile applications [16], [17]. As per the data of 2017, the available IOS applications were over 2 million [18] and over 3 million applications were available for Android [18]. It is also evident that people are transitioning from legacy cell phone to smartphones [19]. Because of this, the mobile applications now enjoy a global impact in the world [20]. Initially, when the mobile applications were getting popular, the practitioners tried to accommodate the existing effort estimation techniques of the desktop applications to calculate the effort of the mobile applications [16], [21], [22]. There has also been a debate as to whether the available desktop techniques are suitable for the effort estimation of mobile applications [23]. It is also concluded that the existing techniques are too cumbersome and cannot be applied for the estimation of mobile applications [24], [25], [26]. In a recent Systematic Literature Review the authors have investigated the issues arising from inaccurate estimation while using the story-based estimation in agile software development. They believe that by incorporating the challenges by the interplay between team dynamics, task complexity, requirements engineering and recommending proper solutions for them software development process can avoid failures [27]. For accommodating comparatively newer development that also includes the mobile applications; a work has proposed and developed a hybrid approach for the project management where agile software development (ASD) project management can be used along with the traditional project management [28].

2.2 Major difference between effort estimation for mobile applications and traditional software

Many research manuscripts have identified the differences between the effort estimation process for traditional software and mobile applications. Some of the major differences and the role they play are mentioned in this section.

Emphasizing the need for metrics that account for human activities for the effort estimation is particularly relevant when comparing traditional software development to mobile application development. The study introduces a novel technique for classifying manual activities into Developer Work Elements (DevWE), which aids in accurately estimating effort. This is crucial for mobile applications, where development comprises of challenges which are unique for mobile applications effort estimation such as varying device specifications and user interactions that differ from traditional software [29].

The adaptation of COSMIC Function Point and UML techniques for estimating the size of mobile applications, emphasizing the unique challenges posed by feature complexities and size in mobile development is discussed. A research work introduces a new measurement technique, Unadjusted Mobile COSMIC Function Points (UMCFP), which incorporates Mobile Complex Factors, illustrating a tailored approach for mobile applications that differs from traditional software estimation methods [30]. A nonlinear multiple regression model which is specifically designed for estimating efforts during the planning phase of mobile application development, highlights the need for more sophisticated modeling techniques due to the dynamic and varied nature of mobile application

Table 3. The mandatory pre-requisites to know before estimating a project

Term	Description
Size	LOC is a direct size metric, and FP is an indirect size metric. The idea behind direct and indirect metrics is that to measure through direct metric one can directly measure by using the metric. To measure through indirect metrics, one needs to perform some mathematical operations generally based on regression.
Effort	The metric used for software effort is persons-month or staff months. If software has an effort of 12 staff months, this would mean that if 12 people work on a project for a month then the project will be completed in a month's time. This does not mean that the project will be completed in a month but that's how the metric works for calculating the effort.
Data	After getting the outputs from a model, data is required for calibration so that the estimates can give appropriate meanings. Some form of data is required by all the models and the three types of data are the industrial data which is the data of some other organization other than the one which is estimating a particular software. Second is the historical data which is old data of some other project of a company, which is now estimating for a newer case. Finally project data is actually the data of the same project which is undergoing estimation. The pre-requisite of using project data in estimation is that the software development methodology should be iterative, as only then the data of the first iteration can be used for later upcoming iterations [10]. Researchers have already formed a consensus that best accuracy is of the project data, followed by historical data and then industrial data. This simply means the lower the variance between estimated value and actual value the better the accuracy and project data will always have the lowest variance from the actual effort [11].
Parkinson's Law	Allocated work is always extended to the actual time which was assigned for the allocation of a task, this is the definition of Parkinson's law, even though if the work was much lesser than the allocated time. This simple means that if a task is assigned to a practitioner which is of two days, but if he/she is given seven days to accomplish the task then now the task will take seven days to finish [12], [13].
Flat or Dynamic Staffing Model	When the software development from the requirements phase to the implementation phase uses the same number of staff members, the estimation model is known as the flat estimation model. On the other hand, if different number of team members are allowed in the model during different phases, then it is known as the dynamic staff model. This simple means that the size of the development team can comprise three persons in the analysis phase and 8 persons in the development phase [10], [11], [14].
Development Time	This is the completion time for a software project, generally specified in calendar months.

requirements as they are different than compared to traditional software [31]. The need for tailored estimation techniques due to the distinct characteristics of mobile applications, which includes the varying device capabilities, user interface considerations, and rapid technological advancements are discussed in a work. The work emphasizes that traditional estimation methods may not adequately address these factors, leading to potential inaccuracies in project planning and resource allocation [32].

The differences in effort estimation between traditional software and mobile applications, particularly in the context of visual aesthetics assessment are thoroughly discussed in a research work and it show how the use of deep learning can help in lessening down the estimation time. The proposed deep learning model has solely been proposed for mobile applications and it automates the assessment process which helps in effort reduction [33]. A work discusses the limitations of traditional models like COCOMO when applied to modern projects, including mobile applications. It emphasizes that traditional estimation methods often struggle with the complexity and rapid evolution of mobile software development, which requires more adaptive and precise approaches [34]. A new technique utilizing the standard deviation technique for better prediction of software effort using analogy-based estimation if proposed for mobile application. A case study has been included that shows the step-by-step procedure for the effort estimation of a mobile application and how to the technique can help in better prediction than compared to traditional approaches [35]. The next sections go in details of the information that was extracted from different research work carried out for the effort estimation of mobile applications.

3 Investigation Results

When investigating through the literature for the effort estimation of mobile applications there were two major categories of the work. The first category included all the literature that included some form of model or techniques for estimating the effort of mobile applications. The other category of the literature was on the validation of some existing techniques proposed for the effort estimation of mobile applications. The categorization of the techniques that were used for the effort estimation of mobile applications are shown in table 4. The investigation results for this research work are different to that of an SLR on current and future trends in software cost and effort

Table 4. Techniques, models mostly used for the Effort Estimation of Mobile Applications

Technique Name	References
COSMIC, CPEEM	[36], [25], [37], [38], [39], [40], [41], [30]
Machine Learning	[42], [43], [44], [31], [45], [46], [47]
UCP	[48], [49], [50], [40], [51]
FPA	[48], [52], [53], [40]
General Features, Factors [57], [58]	[54], [55], [56] COCOMO, COCOMO2
DELPHI	[59], [38]
Sakhrelia method	[52]
FiSMA (Finnish Software Measurement Association)	[16]
Analogy	[22]
SLIM	[51]

estimation because as per their SLR, Artificial Neural Networks ANN and COCOMO based approaches are most favored techniques for effort and cost estimation of software [3]. As per them, MMRE is the most widely used accuracy metric. They have identified difference in preferred effort estimation technique for a general software and for the effort estimation of mobile application, which validates and strengthens the proposed 5+1 methodology of this research work.

It is evident that the most widely used techniques are COSMIC and CPEEM techniques, considered the second-generation methods of FSM (Functional Size Measurement), followed by the machine learning techniques. Next is the Use case point technique, which is used in five research studies. Usecase points technique is one of the techniques that has been derived from the UML diagrams. There are many other techniques which are derived from the UML diagrams to measure the size of software systems, Predictive Object Points (POP) [60], Class Points [61], UML Points [62], Fast&Serious method [63], the metrics for eServices [64]. Variants to the use case techniques are also proposed in different research works [65, 66]. At number 4, the most widely used technique is FPA which is also classified in FSM but under the first-generation methods [67]. Figure 1 shows the histogram of the model/techniques with their occurrences in research studies that have been investigated. An important consideration to note here is that the frequencies give an overview of the mostly used model for the effort estimation of mobile applications. They do not include all the work as including all the research work is not possible, so Table 4 and Figure 1 gives a good idea about the models and techniques mostly used for the effort estimation of mobile applications.

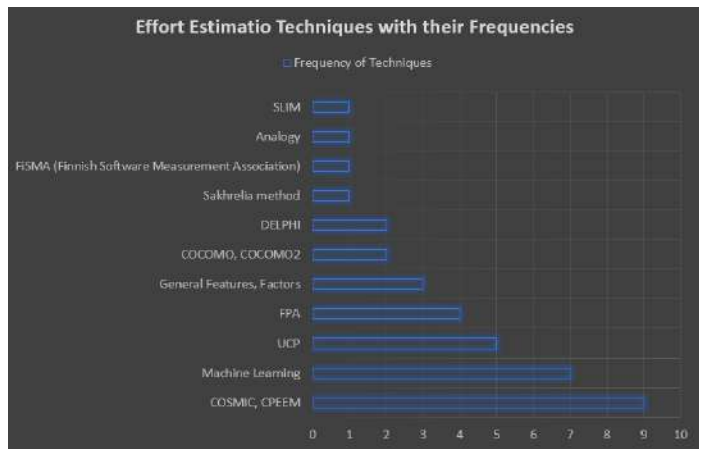

Figure 1. The frequencies of techniques mostly used for the effort estimation of mobile applicationse

Another investigation which was performed in this research work was to identify the accuracy metrics that were used for the validations of the techniques and models which are used for the effort estimation of mobile applications. The selected studies were analyzed deeply to identify the accuracy metrics which were used to validate the results. The investigated studies, have either proposed the methods and techniques or have provided validation of previously proposed techniques, not all but some of them used the accuracy metrics. Table 5 presents the accuracy metrics used in different studies along with their references. A brief definition of the accuracy metrics that had the highest usage frequencies in the investigation of the research work is also provided for the

Table 5. Widely used accuracy metrics for Effort Estimation of Mobile Applications

Accuracy Metric	References
Pred	[25], [37] , [57], [53], [50], [68], [41], [44], [46]
Mean Magnitude of Relative Error (MMRE)	[25], [37], [57], [50], [68], [40], [41], [46], [47]
Median Magnitude of Relative Error	[25], [37], [57], [41]
(MdMRE)	[48], [53], [44], [46]
Magnitude of Relative Error (MRE)	[40]
Mean Balanced Relative Error (MBRE)	[41]
Mean Absolute Error (MAE)	[45]

understanding of the readers.

Pred. and MMRE are the most widely used accuracy metric in the investigated research work. In many research work to strengthen the validation case for accuracy, more than one accuracy metrics are used. Figure 2 shows the graph for the frequency of accuracy metrics in different research work.

Table 6 briefly defines the mostly used accuracy metrics in the research work that were investigated for the effort estimation of mobile applications.

Another investigation that was done in this research work was the identification of the datasets on which a software effort estimation model or a technique is tested. When a new method/technique for effort estimation is proposed or a change is suggested in any effort estimation model, a dataset is required to validate the proposed work. These datasets can be used for some new method/technique, or the dataset will be required to validate mobile applications' effort

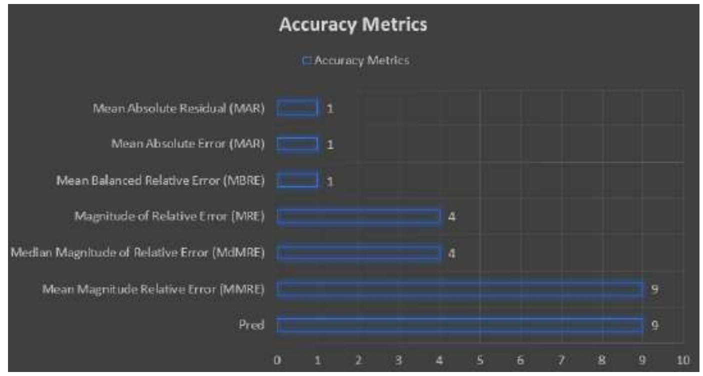

Figure 2. The frequencies of accuracy metrics mostly used for the effort estimation of mobile applications

based on previously proposed techniques or models. Table 7 shows the details about the datasets used in different research work that were investigated.

The next section incorporates and discusses the future directions pertaining to effort estimation for mobile applications which were investigated by an in-depth analysis of the research work pertaining to the mobile applications' effort.

4 Future Directions identified by the research work related to effort estimation of mobile applications

The in-depth study of all identified research work pertaining to the effort estimation of mobile applications points towards some future directions. In this section, an understanding of the identified future directions is presented. This also points toward the open research questions and areas where work is required in future. Almost all the studies that are referred in this work have given some future directions. In Table 8, the essential future directions as per the research work are identified.

Figure 3 depicts the details of the future directions identified through the investigation of the research studies about effort estimation of mobile applications. The details of all the future directions are depicted in Table 8, but the most important future directions are shown in Figure 3.

The next section presents opportunities and research directions along with a 5+1 methodology that

Table 6. Definitions of commonly used accuracy metrics for effort estimation

Accuracy Metric	Definition
Pred (25)	The argument values can be changed with respect to the percentages. Pred (25) and pred (30) are most commonly used. This PRED (25) includes the prediction values which are within 25 percent of the actual values.
Mean Magnitude of Relative Error	Calculates the mean of the Magnitude of Relative Error, which measures the difference between actual and estimated effort of a project.
(MMRE)	The significant advantage of MdMRE over MMRE is that it handles the issue of sensitivity to the outliers. This makes it a more appropriate central tendency measure for data from a skewed distribution [69].
Median Magnitude of Relative Error	For a single measure, MRE is used, and for calculating the mean, MMRE is used.

Table 7. Details of datasets used in different research work.

Reference	Dataset Details
Ferrucci et al. [25]	Dataset of 13 different mobile applications generated through the play store
D'Avanzo et al. [37]	One complete example application and a dataset of 8 mobile applications
Francese et al. [42]	A dataset is used containing information of 23 android projects of a course
Shahwaiz et al. [57]	They applied their model to 44 projects from the dataset. Dataset details are not available
Kaur et al. [48]	Uses a dataset of 10 different mobile applications
Lusky et al. [59]	A dataset exists for features only.
Arnuphaptrairong et al. [53]	A dataset of 17 mobile applications is used
Qi et al. [50]	A 22 records-based dataset is used to validate their results. The dataset is mainly of mobile applications. It shows improvement compared to its parent model of UCPs. Paper of Barry Boehm as the second author.
Mushtaq et al. [41]	It uses a dataset of 36 projects and can be used in other papers.
Pandey et al. [44]	A dataset of 19 projects, the lowest MMRE is of GA. 6 papers in literature review of mobile effort estimation.
Pandey et al. [45]	SAMOA Dataset is used
Qi et al. [56]	Dataset of 34 open-source android projects

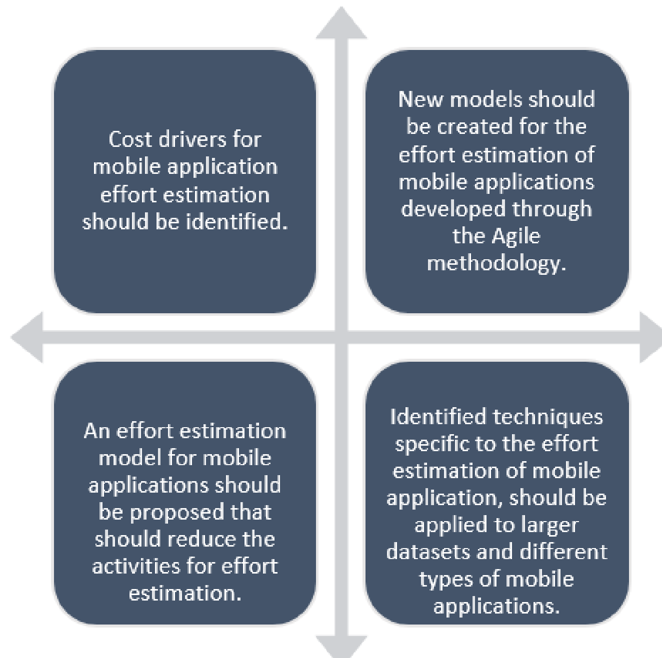

Figure 3. The future directions for effort estimation of mobile applications.

is proposed for creating a model or a technique for the effort estimation of mobile applications. This 5+1 methodology is proposed in lieu of the intensive and rigorous investigations that has been performed through the research work pertaining to the effort estimation of mobile applications.

5 Opportunities, Research Directions and the proposed 5+1 methodology

In this section, the complete analysis is discussed, formulated from the investigation that was conducted for

Table 8. Future Directions Identified Through Different Research Work.

Reference	Future Directions
[16]	The proposed model should be applied to several development projects for mobile applications.
[22]	What problems can be faced when functional requirements are not known?
[25, 37, 40, 41, 45, 52]	Applying the technique to larger datasets and different forms of mobile Applications.
[42]	Applying their method to other development models.
[57]	The model can be validated by using data from other countries.
[48]	Other models can be applied to their methodology. It only uses Function points and Use Case points presently.
[59]	Continuous data collection processes should carry on and techniques other than simple Delphi can be used in future research.
[38]	Develop a model that can make use of the suggested metrics.
[53]	Calibration on own data should be used to achieve better accuracy.
[43]	They plan to apply their proposed model in a case study for future work.
[54]	They identified a few factors for effort estimation and planned to identify factors specific to a particular phase of the Software Development Lifecycle.
[50]	The model should be tested on larger projects with more team members.
[8]	New models should be created for mobile application estimation through the Agile methodology.
[68]	Investigate the accuracy of the proposed model on mobile application effort predictors for more mobile applications.
[9, 46]	A method should be proposed that will reduce the activities of effort estimation.
[20]	Cost drivers should be identified related to mobile applications.
[56]	More data needs to be collected related to transactional and effort from the specific domains of Android projects and calibrate specific models for the specific domains.

the effort estimation of mobile applications. The core findings of the analysis are given as under:

i. Variation to the existing models that were proposed previously are lesser then compared to the new techniques proposed for the effort estimation of mobile applications. This is a positive point, which means researchers are working to develop techniques that will be specific for the effort estimation of mobile application. This points to the fact that soon the stakeholders will see better accuracy in effort estimation for mobile applications in future.

ii. Techniques based on COSMIC, Machine Learning, and Use Case points were dominant in the overall number of studies that were investigated. This means more investigations into these techniques can lead to a breakthrough for the effort estimation of mobile applications.

iii. The Pred and MMRE were the most dominant accuracy metrics used by most studies for the validation of their results.

iv. Proposed techniques require validation and for validation datasets are needed. This points to the fact that the use of datasets should increase in future. This also addresses the fact that datasets about mobile applications should be easily made available to researchers and practitioners.

v. There is a crucial need for more effort estimation models specific to mobile applications.

vi. Efforts are required to investigate further the factors that can help in the early effort identification of mobile applications.

vii. Larger datasets should be available for practitioners and researchers so that the proposed methodologies and models can be tested more rigorously for accuracy.

viii. Cost drivers that are specific to mobile applications should be identified.

ix. Effort estimation of mobile applications for complex scenarios where the functional requirements are not clearly known, where the customers are busy and refrain from giving more time and details to the developers should be considered. Again, this might lead to a new mechanism for the estimation effort of mobile applications.

It is expected that this research work will have a very positive impact on the practitioners and researcher working in the effort estimation domain for mobile applications. It sums up the major effort estimation problems in mobile application effort estimation, which require the attention of practitioners and researchers. The clear direction for the researchers is also defined to give them a way to further investigate in the area of effort estimation of mobile applications.

The investigations and citations of the available datasets can help them work more rigorously to achieve any breakthrough in this domain of study. This work will also help keep the expectation level of the practitioners realistic as they will know about the actual state of the investigation in concern. Considering the investigation into the literature about effort estimation of mobile applications and the identification of the future directions, a 5+1 steps basic methodology is proposed that should be adopted for estimating the effort of mobile applications. The 5+1 steps are listed with explanations and clarified with a complete flow graph of the 5+1 steps which are proposed by this methodology in Figure 4.

- **Step 1:** After having complete understanding of the cone of uncertainty, the first consideration for effort estimation for mobile applications should be to cope up with the cone of uncertainty. Li et al. [63] in their work have demonstrated how they have coped up with volatile requirements for software system that did not allow volatile requirements to refrain the narrowing of the cone of uncertainty, as the narrowing of the cone of uncertainty is very important for project completion. The experts should identify a commitment time for the customers who are getting mobile applications developed. This should be at a point where the cone of uncertainty has narrowed to an acceptable range and a commitment can be made to the customer.
- **Step 2:** Proper understanding of the application development methodology and the application type that is being created will be integrated into the methodology. It's important to note here that agile is not the only methodology that can be used for the development of mobile applications. By type of application, it means whether, a system solution or an application solution is created, as the productivity of practitioners can vary largely for the type of application being created. In a state-of-the-art for effort estimation, the variance caused in effort estimation because of productivity is discussed in detail [70].
- **Step 3:** In the introduction section of this work, the effort estimation models were classified as algorithmic models and non-algorithm models. For implementing this methodology, it is important that the model type is identified, and the cost drivers that can fudge the effort in one direction or the other, needs to be identified particularly for the mobile applications. Researchers have debated about the idea that cost drivers can create an element of subjectivity. However, the use of cost drivers is still very common in many effort estimation models [10], [11].
- **Step 4:** The management is curious to know about the effort of a mobile application as soon as possible. To cater for this need, factors that can help in identifying the effort at an early stage should be accommodated in the methodology. Many research studies have incorporated this requirement in their proposed model. This proposed methodology will identify the factors that can help in early identification of effort [38], [42], [52], [46].
- **Step 5:** Data is required at different stages for calibrating. This proposed methodology will be very simple and allow easy calibration to reach to the expected effort of the mobile applications. Data is required for effort estimation in both types of models, if the models are algorithms, then the calibration comes at a later stage, but for non-algorithm models, they are needed at the initial stage as in analogy-based estimates [71] and other such models.
- **+1 Step:** This methodology will be open, allowing the practitioners to make relevant changes to the cost drivers and let them make decisions at the calibration time. The COCOMO model by Barry Boehm is an entirely open model that allows tailoring things at different stages [72]. As all the steps will require some form of custom manipu-

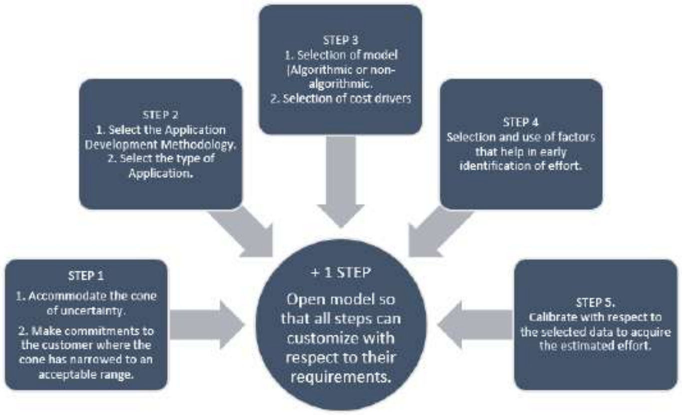

Figure 4. The 5+1 proposed methodology for developing an effort estimation model or framework for mobile applications.

lation, that's why this step sits in the center of all other steps so that the other steps can use the +1 step as needed.

5.1 Case study to understand the rational of 5+1 methodology

To apply the 5+1 methodology in a software house that is developing mobile application for a client that is into retail business. Initially the development team will analyze the cone of uncertainty, considering that requirements will evolve as the project development will progress. When the cone was narrowed to an acceptable range the commitment time for the customer was set. Considering the interactivity and the high probability of changed the agile software development methodology was chosen for the development of this project. The cost drivers such as the complexity of the user interface and integration with existing retail systems were considered to identify the impact on effort estimation. Early effort identification factors, such as the availability of modules which can be used in the development of this mobile application are incorporated to give the strategic management an early estimate of required effort. As data on past projects is available, the team calibrates the model for the current project, adjusting cost drivers and estimating effort more accurately to give better prediction. The +1 step ensures flexibility which allows the software development team to handle the effort estimation methodology with respect to the project's unique needs whenever required.

The 5+1 methodology accommodates the cone of uncertainty with a customized mechanism to effort estimation for mobile applications. The core emphasis of the methodology are early-stage effort identification, adaptability through its +1 step for customization of the methodology and it can utilize both algorithmic and non-algorithmic models. It distinguishes the existing frameworks for effort estimation on the grounds of a completeness, flexibility and easy to adapt effort estimation solution.

Finally in the last section, the research work is concluded with the future directions.

6 Conclusion and Future Directions

This research work provided an in-depth investigation of the research, and highlights the work carried out for the effort estimation of mobile applications. It provides a comprehensive classification of techniques and models proposed for the effort estimation of mobile applications and also investigates the validation done for the proposed techniques for the effort estimation of mobile applications. This work also identifies that computer application users are converting into mobile application users, and soon, a time will be seen where computer applications will be completely obsolete. A 5+1 steps basic methodology is proposed that should be adopted for estimating the effort of mobile applications and which is believed to form the foundation for proposing a model or a framework for the effort estimation of mobile applications.

The limitations and open issues pertaining to the effort estimation of mobile applications are also classified, which paves the way for the future for the researchers and the practitioners. This work also discusses that there is a much greater need for effort estimation model/techniques for mobile applications.

As part of future work, research can commence to validate the actual results that have performed best accuracy for the effort estimation of mobile applications. The diseconomies of scale factor, the project size, and the application type can also be investigated for the mobile application domain to see what difference it creates in the estimated effort for mobile applications. It is also planned to propose a model specific to effort estimation for mobile applications.

Author Contributions

Mohammad Ayub Latif: Conceptualization, Methodology, Analysis, Investigation **Muhammad Khalid Khan**: Supervision, Project Administration, Investigation. **Saad Akbar**: Visualization, Writing, Review and Editing. **Usman Khan**: Project Coordination.

Compliance with Ethical Standards

It is declare that all authors don't have any conflict of interest. It is also declare that this article does not contain any studies with human participants or animals performed by any of the authors. Furthermore, informed consent was obtained from all individual participants included in the study.

References

[1] Lauren, "Mobile App Download Statistics and Usage Statistics (2024)," Mobile App Download Statistics and Usage Statistics (2024). [Online]. Available: https://buildfire.com/app-statistics/

[2] M. Al-Razgan et al., "A systematic literature review on the usability of mobile applications for visually impaired users," PeerJ Comput. Sci., vol. 7, p. e771, 2021.

[3] C. H. Rashid et al., "Software Cost and Effort Estimation: Current Approaches and Future Trends," IEEE Access, 2023.

[4] I. F. de Barcelos Tronto, J. D. S. da Silva, and N. Sant'Anna, "An investigation of artificial neural networks based prediction systems in software project management," J. Syst. Softw., vol. 81, no. 3, pp. 356–367, 2008.

[5] A. Ali et al., "Mobile-UI-Repair: a deep learning based UI smell detection technique for mobile user interface," PeerJ Comput. Sci., vol. 10, p. e2028, 2024.

[6] B. Kitchenham, O. P. Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, "Systematic literature reviews in software engineering–a systematic literature review," Inf. Softw. Technol., vol. 51, no. 1, pp. 7–15, 2009.

[7] R. Britto, M. Usman, and E. Mendes, "Effort estimation in agile global software development context," in Agile Methods. Large-Scale Development, Refactoring, Testing, and Estimation: XP 2014 International Workshops, Rome, Italy, May 26-30, 2014, Revised Selected Papers 15, Springer, 2014, pp. 182–192.

[8] A. Kaur and K. Kaur, "Effort estimation in traditional and agile mobile application development testing," Indones. J. Electr. Eng. Comput. Sci., vol. 12, no. 3, pp. 1265–1272, 2018.

[9] A. Altaleb and A. Gravell, "An empirical investigation of effort estimation in mobile apps using agile development process," J. Softw., vol. 14, no. 8, pp. 356–369, 2019.

[10] S. McConnell, Software estimation: demystifying the black art. Microsoft press, 2006.

[11] M. A. Latif, M. Y. Khan, and K. Bashir, "Practices for Achieving Accuracy in Software Costing and Estimation," KIET J. Comput. Inf. Sci., vol. 1, no. 1, p. 13, Jul. 2018, doi: 10.51153/kjcis.v1i1.13.

[12] C. Northcote Parkinson, Parkinson's Law and Other Studies in Administration. Houghton Mifflin, 1957.

[13] C. F. Kemerer, "An empirical validation of software cost estimation models," Commun. ACM, vol. 30, no. 5, pp. 416–429, 1987.

[14] B. Boehm, C. Abts, and S. Chulani, "Software development cost estimation approaches—A survey," Ann. Softw. Eng., vol. 10, no. 1, pp. 177–205, 2000.

[15] A. Nitze, "Measuring mobile application size using COSMIC FP," in DASMA Metrik Kongress, 2013, pp. 1–13.

[16] L. S. De Souza and G. S. de Aquino Jr, "Estimating the effort of mobile application development," in Proceedings of Second International Conference on Computational Science and Engineering, 2014, pp. 45–63.

[17] N. A. S. Abdullah and N. I. A. Rusli, "Reviews on functional size measurement in mobile application and UML model," 2015.

[18] S. Costello, "How many apps are in the app store? Lifewire." 2017.

[19] P. Zheng and L. M. Ni, "Spotlight: the rise of the smart phone," IEEE Distrib. Syst. Online, vol. 7, no. 3, pp. 3–3, 2006.

[20] R. Islam, R. Islam, and T. Mazumder, “Mobile application and its global impact,” Int. J. Eng. Technol., vol. 10, no. 6, pp. 72–78, 2010.

[21] L. S. de Souza and G. S. de Aquino, “The applicability of present estimation models to the context of mobile applications,” in 2014 9th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE), IEEE, 2014, pp. 1–6.

[22] A. Nitze, A. Schmietendorf, and R. Dumke, “An analogy-based effort estimation approach for mobile application development projects,” in 2014 Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement, IEEE, 2014, pp. 99–103.

[23] G. Jošt, J. Huber, and M. HeriČko, “Using object oriented software metrics for mobile application development,” in 2nd workshop of software quality analysis, monitoring, improvement, and applications, 2013, pp. 17–27.

[24] B. Boehm, “Cost estimation with COCOMO II,” Cent. Softw. Eng., 2002.

[25] F. Ferrucci, C. Gravino, P. Salza, and F. Sarro, “Investigating functional and code size measures for mobile applications: A replicated study,” in Product-Focused Software Process Improvement: 16th International Conference, PROFES 2015, Bolzano, Italy, December 2-4, 2015, Proceedings 16, Springer, 2015, pp. 271–287.

[26] F. J. Heemstra, “Software cost estimation,” Inf. Softw. Technol., vol. 34, no. 10, pp. 627–639, 1992.

[27] M. Iqbal et al., “Exploring issues of story-based effort estimation in Agile Software Development (ASD),” Sci. Comput. Program., vol. 236, p. 103114, 2024.

[28] J. Leong, K. May Yee, O. Baitsegi, L. Palanisamy, and R. K. Ramasamy, “Hybrid project management between traditional software development lifecycle and agile based product development for future sustainability,” Sustainability, vol. 15, no. 2, p. 1121, 2023.

[29] P. Sopahtsathit, “Software analytics for manual activities using developer work elements,” J. Inf. Process., vol. 28, pp. 279–291, 2020.

[30] Z. Mushtaq and A. Wahid, “Revised approach for the prediction of functional size of mobile application,” Appl. Comput. Inform., vol. 20, no. 1/2, pp. 181–193, 2024.

[31] S. Prykhodko, N. Prykhodko, and K. Knyrik, “Estimating the Efforts of Mobile Application Development in the Planning Phase Using Nonlinear Regression Analysis.,” Appl Comput Syst, vol. 25, no. 2, pp. 172–179, 2020.

[32] N. Rusli, N. Abdullah, F. Abd Razak, and N. Mufriz, “Web-Based Parametric Effort Estimation for Mobile Application Development. Proceedings 2022, 82, 69,” in International Academic Symposium of Social Science 2022, s Note: MDPI stays neutral with regard to jurisdictional claims in published . . . , 2022, p. 131.

[33] A. L. de Souza Lima, C. G. von Wangenheim, O. P. Martins, A. von Wangenheim, J. C. Hauck, and A. F. Borgatto, “A Deep Learning Model for the Assessment of the Visual Aesthetics of Mobile User Interfaces,” J. Braz. Comput. Soc., vol. 30, no. 1, pp. 102–115, 2024.

[34] N. A. Zakaria, A. R. Ismail, N. Z. Abidin, N. H. M. Khalid, and A. Y. Ali, “Optimized COCOMO parameters using hybrid particle swarm optimization,” Int. J. Adv. Intell. Inform., vol. 7, no. 2, pp. 177–187, 2021.

[35] M. A. Latif, M. K. Khan, and U. Hani, “Using Standard Deviation with Analogy-Based Estimation for Improved Software Effort Prediction.,” KSII Trans. Internet Inf. Syst., vol. 17, no. 5, 2023.

[36] N. A. S. Abdullah, N. I. A. Rusli, and M. F. Ibrahim, “Mobile game size estimation: Cosmic fsm rules, uml mapping model and unity3d game engine,” in 2014 IEEE Conference on Open Systems (ICOS), IEEE, 2014, pp. 42–47.

[37] L. D’Avanzo, F. Ferrucci, C. Gravino, and P. Salza, “Cosmic functional measurement of mobile applications and code size estimation,” in Proceedings of the 30th annual ACM symposium on applied computing, 2015, pp. 1631–1636.

[38] G. Catolino, P. Salza, C. Gravino, and F. Ferrucci, “A set of metrics for the effort estimation of mobile apps,” in 2017 IEEE/ACM 4th International Conference on Mobile Software Engineering and Systems (MOBILESoft), IEEE, 2017, pp. 194–198.

[39] M. Bachiri, A. Idri, L. Redman, A. Abran, J. M. C. de Gea, and J. L. Fernández-Alemán, “COSMIC functional size measurement of mobile personal health records for pregnancy monitoring,” in New Knowledge in Information Systems and Technologies: Volume 3, Springer, 2019, pp. 24–33.

[40] F. Ugalde, C. Quesada-López, A. Martínez, and M. Jenkins, “A comparative study on measuring software functional size to support effort estimation in agile.,” in CIbSE, 2020, pp. 208–221.

[41] Z. Mushtaq and A. Wahid, “Inclusion of Functional and Non-Functional Parameters for the Prediction of Overall Efforts of Mobile Applications.,” Comput. Stand. Interfaces, vol. 71, p. 103404, 2020.

[42] R. Francese, C. Gravino, M. Risi, G. Scanniello, and G. Tortora, “On the use of requirements measures to predict software project and product measures in the context of Android mobile apps: A preliminary study,” in 2015 41st Euromicro Conference on Software Engineering and Advanced Applications, IEEE, 2015, pp. 357–364.

[43] N. R. Darwish and Y. M. Abdelmohsen, “Toward a Proposed Model for EffortEstimation of Developing Mobile Applications”.

[44] M. Pandey, R. Litoriya, and P. Pandey, “Validation of existing software effort estimation techniques in context with mobile software applications,” Wirel. Pers. Commun., vol. 110, no. 4, pp. 1659–1677, 2020.

[45] M. Pandey, R. Litoriya, and P. Pandey, “Applicability of machine learning methods on mobile app effort estimation: Validation and performance evaluation,” Int. J. Softw. Eng. Knowl. Eng., vol. 30, no. 01, pp. 23–41, 2020.

[46] Z. Mushtaq, S. Alshmrany, F. Alturise, and T. Alkhalifah, “Early Size and Effort Estimation of Mobile Application Development,” EAI Endorsed Trans. Energy Web, vol. 9, no. 37, pp. e3–e3, 2022.

[47] M. Pandey, R. Litoriya, and P. Pandey, “An integrated MCDM approach for mobile app cost predictor based on DEMA extended with choquet integral,” Multimed. Tools Appl., vol. 83, no. 12, pp. 34943–34962, 2024.

[48] A. Kaur and K. Kaur, “Analyzing Adaption of Size and Effort Estimation Approaches in Mobile Software”.

[49] R. Sutoyo, H. L. H. S. Warnars, F. L. Gaol, E. Abdurachman, and B. Soewito, “Measurement of QuestDone mobile application using 7 steps use case points method,” in 2017 IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom), IEEE, 2017, pp. 90–95.

[50] K. Qi and B. W. Boehm, “Detailed use case points (DUCPs) a size metric automatically countable from sequence and class diagrams,” in Proceedings of the 10th International Workshop on Modelling in Software Engineering, 2018, pp. 17–24.

[51] E. Markevich, V. Kukartsev, A. Strokan, E. Nozdrenko, N. Lysyannikova, and S. C. Mongush, “Comparative analysis of UCP and SLIM methods to estimate the development of software products,” in Journal of Physics: Conference Series, IOP Publishing, 2021, p. 012121.

[52] M. Haoues, A. Sellami, and H. Ben-Abdallah, “A rapid measurement procedure for sizing web and mobile applications based on COSMIC FSM method,” in Proceedings of the 27th International Workshop on Software Measurement and 12th International Conference on Software Process and Product Measurement, 2017, pp. 129–137.

[53] T. Arnuphaptrairong and W. Suksawasd, “An empirical validation of mobile application effort estimation models,” in Proceedings of the International MultiConference of Engineers and Computer Scientists, 2017, pp. 15–17.

[54] G. Catolino, “Effort-oriented methods and tools for software development and maintenance for mobile apps,” in Proceedings of the 40th International Conference on Software Engineering: Companion Proceeedings, 2018, pp. 450–451.

[55] A. Altaleb, M. Altherwi, and A. Gravell, “An industrial investigation into effort estimation predictors for mobile app development in agile processes,” in 2020 9th International Conference on Industrial Technology and Management (ICITM), IEEE, 2020, pp. 291–296.

[56] K. Qi and B. Boehm, “Effort estimation of open source Android projects via transaction analysis,” J. Softw. Evol. Process, vol. 33, no. 1, p. e2253, 2021.

[57] S. A. Shahwaiz, A. A. Malik, and N. Sabahat, “A parametric effort estimation model for mobile apps,” in 2016 19th International Multi-Topic Conference (INMIC), IEEE, 2016, pp. 1–6.

[58] V. Tynchenko, A. Andreev, Y. F. Kaizer, S. Zinner, N. Lysyannikova, and N. Kuzmin, “Application of the COCOMO II method to estimate the labor costs of developing software products for various platforms,” in Journal

of Physics: Conference Series, IOP Publishing, 2021, p. 012124.

[59] M. Lusky, C. Powilat, and S. Böhm, “Software cost estimation for user-centered mobile app development in large enterprises,” in Advances in Human Factors, Software, and Systems Engineering: Proceedings of the AHFE 2017 International Conference on Human Factors, Software, and Systems Engineering, July 17-21, 2017, The Westin Bonaventure Hotel, Los Angeles, California, USA 8, Springer, 2018, pp. 51–62.

[60] A. Minkiewicz, “Measuring object oriented software with predictive object points,” PRICE Syst. LLC, pp. 609–866, 1997.

[61] G. Costagliola, F. Ferrucci, G. Tortora, and G. Vitiello, “Class point: an approach for the size estimation of object-oriented systems,” IEEE Trans. Softw. Eng., vol. 31, no. 1, pp. 52–74, 2005.

[62] S. Kim, W. M. Lively, and D. B. Simmons, “An Effort Estimation by UML Points in Early Stage of Software Development.,” in Software Engineering Research and Practice, Citeseer, 2006, pp. 415–421.

[63] M. Carbone, G. Santucci, and others, “Fast and Serious: a UML based metric for effort estimation,” in Proceedings of the 6th ECOOP workshop on quantitative approaches in object-oriented software engineering (QAOOSE’02), Citeseer, 2002, pp. 313–322.

[64] Y. Chen, B. W. Boehm, R. Madachy, and R. Valerdi, “An empirical study of eServices product UML sizing metrics,” in Proceedings. 2004 International Symposium on Empirical Software Engineering, 2004. ISESE’04., IEEE, 2004, pp. 199–206.

[65] J. Smith, “The estimation of effort based on use cases,” Ration. Softw. White Pap., 1999.

[66] J. Lee, W.-T. Lee, and J.-Y. Kuo, “Fuzzy logic as a basic for use case point estimation,” in 2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011), IEEE, 2011, pp. 2702–2707.

[67] S. Di Martino, F. Ferrucci, C. Gravino, and F. Sarro, “Web effort estimation: function point analysis vs. COSMIC,” Inf. Softw. Technol., vol. 72, pp. 90–109, 2016.

[68] S. Prykhodko, N. Prykhodko, K. Knyrik, and A. Pukhalevych, “Mathematical Modeling of Effort of Mobile Application Development in a Planning Phase.,” in ICTES, 2019, pp. 96–105.

[69] F. Valdés and A. Abran, “Comparing the estimation performance of the EPCU model with the expert judgment estimation approach using data from industry,” Softw. Eng. Res. Manag. Appl. 2010, pp. 227–240, 2010.

[70] S. S. Gautam and V. Singh, “The state-of-the-art in software development effort estimation,” J. Softw. Evol. Process, vol. 30, no. 12, p. e1983, 2018.

[71] M. Shepperd and C. Schofield, “Estimating software project effort using analogies,” IEEE Trans. Softw. Eng., vol. 23, no. 11, pp. 736–743, 1997.

[72] B. Barry and others, “Software engineering economics,” N. Y., vol. 197, no. 140, p. 6, 1981.