



# Integrating software effort estimation with risk management

Prerna Singal<sup>1</sup> · Prabha Sharma<sup>1</sup> · A. Charan Kumari<sup>2</sup>

Received: 9 September 2021 / Revised: 6 January 2022 / Accepted: 7 March 2022 / Published online: 6 April 2022

© The Author(s) under exclusive licence to The Society for Reliability Engineering, Quality and Operations Management (SREQOM), India and The Division of Operation and Maintenance, Lulea University of Technology, Sweden 2022

## Abstract

**Context** Risks associated with software projects play a significant role on delivery of software projects within a given budget. These risks are due to volatility in project requirements, availability of experienced personnel, ever-changing technology and many more project cost factors. Effort spent on managing the risks is termed as the risk exposure of the project. In this research, this risk exposure has been added to effort estimate of a software project. This total effort is termed as the integrated effort estimate. **Objective** To improve the accuracy of software effort estimates by integrating the risk exposure with the initial effort estimate of the project.

**Method** A formula to calculate integrated effort estimate of a software project has been proposed in the paper. This proposed formula has been tested on two datasets collected from industry, one for waterfall projects and another for agile projects. Initial effort estimates for waterfall projects are calculated using CoCoMo II and for agile projects are calculated using story point approach by Ziauddin.

**Results** The integrated effort estimates were more accurate than their corresponding initial effort estimates on all the four parameters: MMRE, SA, effect size and  $R^2$ .

**Conclusion** Integrated effort estimates are more comprehensive, reliable, and accurate than the initial effort estimates for the project.

**Keywords** CoCoMo II · Agile · Software effort estimation · Waterfall · Risk exposure · Risk management

## 1 Introduction

The uncertainty in the effort estimate of software development projects must be analysed in order to bridge the gap between the actual and predicted effort estimate of software projects (JøRgensen et al. 2004). This uncertainty is not necessarily due to inefficient effort estimation techniques, but due to the lack of knowledge in properly handling the uncertainties and associated risks in the effort estimation process itself (Kitchenham 1998). These uncertainties arise due to complex and non-linear adaptive nature of the software development process. So, for a more comprehensive and accurate effort estimation technique, these uncertainties should be addressed in the effort estimation technique itself. These uncertainties arise due to the risks associated with projects, which if not handled on time may lead to project failure, delay or budget overruns (Morgenshtern et al. 2007).

The risks associated with a software project are due to factors like volatility in project requirements, availability of experienced personnel or ever-changing technology. These factors play a significant role in the effort estimation process as well as the risk identification and management process (Capretz 2013; Kansala 1997). Most often, risk planning and its management is treated as a separate activity from the software effort estimation task (Boehm 1989). Risks are identified, analysed, mitigated, and controlled but their impact on the effort estimate of a project is not considered. This might result in over-optimism and over-confidence in the software effort estimate (JøRgensen

---

✉ Prerna Singal  
prernasingal@yahoo.com

<sup>1</sup> The NorthCap University, Gurugram, India

<sup>2</sup> Dayalbagh Educational Institute, Dayalbagh, Agra, India

2010). Thus, there is a need to integrate the risk management process in the effort estimation process for more accurate, comprehensive, and fair effort estimation.

Risks associated with a project can be either at the organization level or can be specific to a project (Higuera and Haimes 1996). For successful completion of a project, these risks need to be addressed at the project level. Risk management at the project level, therefore, will require extra effort which must be a part of the software effort estimation process. This implies that while estimating the effort for a software project, effort involved in management and control of risk should also be accounted for in the effort estimate of the project.

Risk management at the project level would involve estimating the exposure of the project to various risks. Risk exposure is potential loss to the project in the event the risk occurs (Barki et al. 2001). This loss can be determined by estimating the extra effort that would be spent in management of the risks at the project level. The various project cost factors determine the risks associated with the project and these project cost factor values will determine the risk exposure of the project (Madachy 1997). The proposed approach gives a method for determining this extra effort that goes into risk management based upon the project cost factor values. The risk exposure of the project is determined as the extra effort that will go into the risk mitigation and control when the assumed level of the project cost factor value is not met during the execution of the project (Kitchenham and Linkman 1997). This is essentially the error in assumption of the project cost factor values in the software effort estimation process which leads to inaccuracies in the software effort estimate.

In 1997, Madachy proposed a heuristic to calculate the risk exposure of the project using the project cost factors. He identified risk rules based on the CoCoMo cost factors and then assigned levels to various risks based on these risk rules. The risk exposure of the project was the sum of the product of the risk level and the contributing project cost factor values (Madachy 1997). This approach considers that the project cost factor values are assumed correctly by the experts, and the risks arise due to certain risk rules when applied to project cost factors.

In 1997, another researcher Kansala integrated the risk assessment process with software effort estimation and termed it as “RiskMethod”. He tested the proposed method on the project cost factors identified in the CoCoMo II model. His work dealt with assessing the impact of risk on the project from the project cost factor values, defining the total risk exposure of the project as the sum of risk exposure due to each project cost factor (product of probability of the risk due the factor and potential loss in case the risk occurs). This total risk exposure was used to prioritize, assess and mitigate the risks associated with the project

(Kansala 1997). According to the author, although Risk-Method worked well to calculate the risk exposure, it is difficult to understand for novice users and needs a supervisor. Besides, the effort spent on the risk exposure was not considered part of the total effort estimate.

In 1998, Briand et al. introduced a hybrid model for project effort estimation, risk assessment and benchmarking without relying on the historical data for project effort estimate. The proposed model based its effort estimate on productivity of the project and calculated the project cost overheads based on a questionnaire, to be filled by the experts associated with the project. This approach also appreciated the integration of risk assessment along with software project effort estimation (Briand et al. 1998). Their work relied solely on expert judgement, which allows for lot of scope for bringing in biases of the experts.

In 2006, Jantzen estimated the impact of project risks on project effort estimate, project duration and project quality. Their work emphasized on re-estimating and re-planning the software project during its execution, based on the various risks, their levels and status at various stages of the project (Jantzen 2006). The approach suggested considering risks with high probability first and then removing them from subsequent estimates once they have been mitigated. The work was not tested on any of the established effort estimation techniques.

Huang et al. in 2006, based on the fuzzy and uncertain nature of the project cost factors proposed an effort estimation technique based on fuzzy decision tree where along with the effort estimate of the project, the estimation error was used for risk analysis and its management (Huang et al. 2006). Their work suggested integrating the estimation error from the fuzzy decision tree into the effort estimate of the project but gave no information on the source of the error and risks to be controlled.

Manalif in 2013, proposed a fuzzy expert—CoCoMo model which added a contingency to the estimated effort based on the project cost factors of the CoCoMo model. The model integrated the risk management and project effort estimation steps, provided insights into estimated effort and project risks, and effort contingency to accommodate the identified risks (Capretz 2013; Manalif 2013). The contingency effort was based on risk rules which considered very few CoCoMo project cost factors.

In 2017, Aslam et al. considered the risks associated with rich mobile application development projects developed using agile methodology (Aslam et al. 2017). Along with risk factors, they also included the quality aspect of the project in the effort estimate of the project which enabled the development effort estimation at multiple quality levels. Their work was limited to project on rich mobile application development.

In 2018, Koutbi & Idri proposed inclusion of the cost of risk management in the effort estimation process at the organization level instead of handling it at a specific project level (Koutbi and Idri 2018). They argued that risk is better handled and mitigated over a portfolio of projects which improves the effort estimation process of the organization. But organizations can have projects of varied nature, for which the project cost factors contributing to risks may vary.

Dalal et al. (2018) applied generalized reduced gradient nonlinear optimization with best fit analysis to CoCoMo project cost factors to improve software effort estimation accuracy. They have tuned the CoCoMo cost drivers using optimization technique. This research proposes to gather alternative values for the project cost factors to improve the estimation accuracy.

In 2019, Ramakrishnan et al. built a multilayer perceptron model to estimate the software development effort. The model included project risk score in the effort estimation process (Ramakrishnan et al. 2019). They used an enhanced gradient boosting technique which decreased the standard deviation of the residuals indicating better effort estimation results.

In 2020, Bhavsar et al. proposed a hybrid of scrum, Kanban and waterfall for management and estimation of software development projects (Bhavsar et al. 2020). In their research they highlighted the areas where the proposed approach will perform better than the standalone approaches.

In 2021, Tawosi et al. conducted a replication study for validation of multi objective effort estimation technique CoGEE (Tawosi et al. 2021). The JAVA version of the algorithm decreased the running time of CoGEE by 99.8% as compared to the earlier R version.

In 1997, Kitchenham and Linkman suggested that the uncertainties in the software development process cause inaccuracies in the software effort estimate irrespective of the effort estimation technique being used. The effort estimation is done in the beginning of the project when not many details are available regarding the various project cost factors impacting the effort estimate of the project. They have categorized sources of estimate uncertainties into three types: measurement error, model error and assumption error. Assumption errors occur in the evaluations of the project cost factor values due to the inherent uncertainties associated with these parameters. This assumption error is the risk which creeps into the project when project cost factor values do not meet the assumed level. Thus, the effort which goes into controlling such risks can be calculated by taking the difference in effort at the assumed level and the alternative level for each project cost factor and then multiplying it with the corresponding probability of attaining that alternative level. Sum of the

risk exposure for each project cost factor was termed as risk exposure of the project, which needs to be controlled and mitigated for successful project delivery (Kitchenham and Linkman 1997).

Since the first two errors relate to the uncertainties at the organisational level, this research paper considers only assumption error at the project level. This research uses the formula for calculation of risk exposure due to uncertainties given in Kitchenham and Linkman (1997). The proposed approach calculates the integrated effort estimate (*IE*) of software projects by adding the risk exposure to the initial effort estimate of the projects. Project experts can use this integrated effort for risk planning and mitigation along with software project planning. This research investigates the accuracy of the proposed model on the two most popular delivery approaches: waterfall model and Agile delivery model (Sureshchandra and Shrinivasavadhani 2008).

### 1.1 The main contributions of this research are as follows:

1. This research integrates the effort spent on risk management and planning with the software effort estimate of a project. The proposed approach gives a more comprehensive and realistic effort estimate of a project. Thus, it will help the project managers in delivering the software projects on time and within budget.
2. Two industrial datasets have been collected, one for the waterfall delivery projects based on CoCoMo II project cost factors (Boehm et al. 1997). Another for the agile delivery projects based on the project cost factors suggested by Ziauddin (Ziauddin and Zia 2012).
3. Two questionnaires for data collection have been prepared, one each for CoCoMo II model and Ziauddin approach.
4. Integrated effort estimates have been compared with the benchmark estimation techniques: CoCoMo II for waterfall projects and story point approach by Ziauddin for agile projects.
5. The integrated effort estimates for both the datasets are more accurate than the initial effort estimates.

Rest of the paper is organised as follows: Sect. 2 presents the proposed approach of integrating the impact of risk on the software effort estimate. Section 3 outlines research questions, datasets and evaluation criteria used for comparing the proposed approach with the baseline effort estimation technique. Section 4 presents experimental settings or the datasets. Section 5 presents the evaluation results of the proposed approach on the waterfall delivery model using CoCoMo II estimation technique and evaluation results on the agile delivery model using the story

size approach by Ziauddin and Zia (2012). Section 6 details some threats to validity of the proposed approach. Section 7 draws the conclusion and scope for future work.

## 2 Proposed approach

The proposed approach suggests including the risk exposure of a project in its effort estimate. This risk exposure is the effort dispensed in mitigating the risks involved in the completion of the project when the initial assumptions regarding the project cost factors are not met. All software projects are planned and estimated based on certain assumptions regarding the project cost factors or environment like associate experience, training and expertise level, requirements volatility, similarity with previous projects, language and tool experience, inter-personnel communication, and project architecture. These assumptions are made at a very nascent stage in the project lifecycle when much information about the project is not available. There is always an uncertainty associated with these assumptions, thus leading to various risks in a successful project delivery. The proposed approach investigates the improvement in effort estimation accuracy when considering the effort required to mitigate and control the project risk exposure in its initial effort estimate as shown in Fig. 1. This research proposes the following formula for calculating the integrated effort estimate ( $IE_i$ ) of  $i$ th software project:

$$IE_i = E_{i,initial} + E_{i,risk} \tag{1}$$

where  $E_{i,initial}$  is the initial effort estimate of the project, and  $E_{i,risk}$  is the risk exposure of the project. The risk exposure of the project is the sum of the individual risk

exposure due to each project cost factor. Each project cost factor value is assumed to be at a certain level while estimating the initial effort of the software project. These initial project cost factor values are determined from people who are experts and have long experience in handling the projects. But this initial level ( $EM_k$ ) of  $k$ th project cost factor might change due to the varying project conditions. Thus, the proposed approach considers an alternative level ( $EM_{k,alter}$ ) of  $k$ th project cost factor along with the probability ( $p_{k,alter}$ ) of not meeting the initial level. These alternative values along with their probabilities of occurrence have also been obtained from the same experts who did the initial effort estimation of the project. It is possible that some biases may get built into these probabilities. However, by considering more than one alternative for each project cost factor or by computing optimum values of the probabilities, the biases can be reduced. For both waterfall and agile delivery models, the benchmark estimation methods used (CoCoMo II for waterfall and Ziauddin for Agile) consider the project cost factors to be independent of each other. So, the probabilities for alternative project cost factor values will not add up to 1. For each project cost factor, the risk exposure of the  $i$ th project is calculated using the risk exposure formula given by Kitchenham and Linkman (1997):

$$E_{i,risk} = \sum_{k=1}^n (E_{k,alter} - E_{i,initial}) \times p_{k,alter} \tag{2}$$

where  $n$ , is the number of project cost factors considered in the initial effort estimate of the project.  $E_{k,alter}$  is the estimated effort at the alternative level of the  $k$ th project cost factor.  $E_{k,alter}$  is calculated using the alternative level  $EM_{k,alter}$  of the  $k$ th cost factor and initial level  $EM$  of rest of

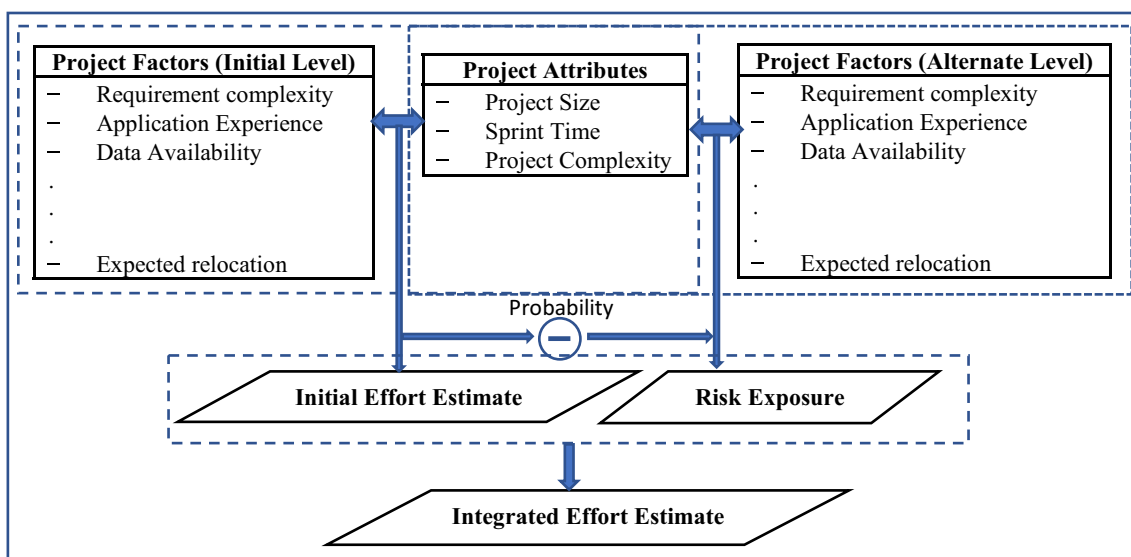


Fig. 1 Proposed approach for integrated effort estimation

the cost factors.  $E_{i,initial}$  is the initial estimated effort of the  $i$ th project at the initial assumed level of the project cost factors. There may be multiple alternative values to a project cost factor depending upon the varying project conditions, each having a probability of occurrence. For this research only one alternative value has been considered. Risk exposure for each project cost factor might increase or decrease the integrated estimated effort ( $IE_i$ ). In the above formula for risk exposure in Eq. (2), negative risk exposure is considered, since the proposed formula only computes the difference between the alternative effort estimate and the original estimate without the absolute sign. Cases where the project cost factor value improves from initial to the alternative level, will lower the  $IE$ . Cases where the initial project cost factor value is greater than the alternative level, will increase the  $IE$ .

For e.g.: consider a project where a new technology stack is to be used for developing the project. During project planning phase, it was assumed that the availability of trained resources in the new technology will be scarce, so the developer expertise level was considered very low while estimating the effort. At the same time, the manager was trying to find and recruit experienced personnel in the required technology area. The manager successfully recruited the required number of experts in the said area. Now when the project started the developer expertise level was very high, thus the project cost factor rating level for developer expertise improved, decreasing the effort estimate of the project.

Consider another situation, when at the beginning of the project the company had an agreement that all the required hardware and computing platform will be provided before the development of the project begins. So, the platform and hardware availability were assumed to be high while estimating the initial effort. But the hardware that arrived had some technical issues. This resulted in extra effort by the developers to build and develop the software, resulting in increase in the final effort estimate. So, risk exposure for a project cost factor might be negative or positive depending upon the change in project cost factor value from the initial to the alternative level. Figure 2 shows the flow of steps involved in calculating the integrated effort estimates.

In this research, the proposed approach has been applied to projects delivered using Waterfall as well as Agile delivery model. For waterfall delivery model, software project effort was estimated using COCOMO II (Boehm et al. 1997). For Agile delivery model, software project effort was estimated using the story point estimates as suggested by Ziauddin and Zia (2012). The steps have been described in Sect. 4 for both waterfall and agile delivery models.

### 3 Methodology

This section presents the research questions, datasets and evaluation criteria used to evaluate the accuracy of integrated effort estimates.

#### 3.1 Research question

This paper aims to provide the experimental evidence to answer the research question given below:

**RQ1** How good is the proposed function for calculating integrated effort estimates of software projects?

This research aims to integrate the effort spent on risk management into the effort estimate of a software project. This research proposes a function ( $IE$ ) to determine this integrated effort estimate. This research also compares the accuracy and reliability of the proposed approach with two other effort estimation techniques (CoCoMo II and story point approach by Ziauddin).

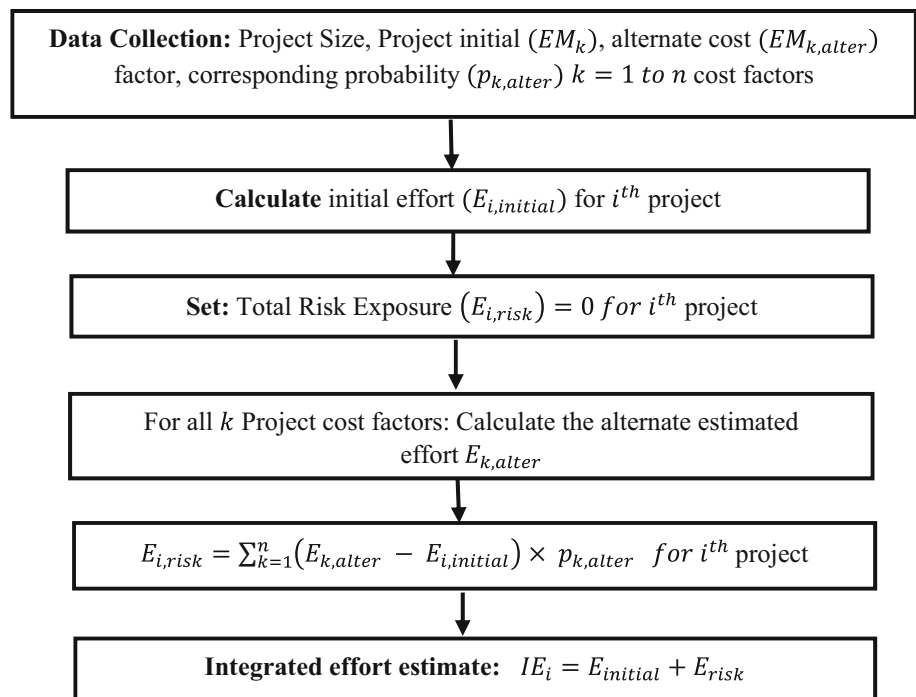
#### 3.2 Datasets

Datasets were collected from an Indian IT firm involved in software development, maintenance, and consultancy of software projects. Two types of projects were considered for the research—projects with Waterfall delivery model (Gilb 1985) and projects with Agile delivery model (Martin and Martin 2006). Experts from the projects were interviewed over a span of 1 year and data was collected based on two separate questionnaires—one for each delivery model, waterfall and agile. Experts included project managers, technical architects, analysts, and developers. These experts were directly involved in the project effort estimation process. Experts from over 75 different projects were interviewed, with 45 projects following the Waterfall Delivery Model and rest 30 were working on the agile principles. Projects were from varied domains covering banking, healthcare & pharmaceutical, and Insurance.

The questionnaires were designed using Microsoft Excel in a tabular format to make filling of the data convenient for the experts. Waterfall Model questionnaire had 69 fields to be filled while the Agile questionnaire had 45 fields. Table 1 shows a general format of the questionnaire.

Questionnaire for the Waterfall Model was based on CoCoMo II project cost factors (Boehm et al. 1997). The questionnaire focussed on lines of code in the project (measured in KLOC), actual effort spent (Man Months), and the project cost factors—their initial assumed level while estimating effort, probability of not meeting that assumed level and expected alternative level. The dataset thus collected is referred to as the “Waterfall model”

**Fig. 2** Steps in the proposed approach for integrated effort estimation



**Table 1** Questionnaire format

Questionnaire					
Project Id	KLOC/story points	Actual effort (man months)	Cost driver Assumed level	Probability of not meeting the assumed level	Expected alternative level

dataset. Similarly, questionnaire for the Agile model was based on the frictional and variable forces suggested by Ziauddin and Zia (2012). The questionnaire focused on the sprint time, story size, actual velocity, story complexity and the frictional & variable forces—their initial assumed level during effort estimation, probability of not meeting that assumed levels and the expected alternative level. The dataset thus collected is referred to as the “Agile model” dataset.

### 3.3 Evaluation criteria

The integrated effort estimated with the proposed model is compared with the initial estimated effort using the benchmark model based on four performance evaluation metrics: mean magnitude of relative error (MMRE), standardized accuracy (SA), effect size ( $\Delta$ ) and coefficient of determination ( $R^2$ ).

#### 3.3.1 Magnitude of relative error

Magnitude of relative error (MRE) is the ratio of the absolute difference between the integrated effort (IE) and the actual effort spent on a project, and the actual effort spent on a project. The formula for MRE will be:

$$MRE = \frac{|IE - actual\ effort|}{actual\ effort}$$

From the definition of MRE it follows that a high value will indicate that the estimated effort is far off from the actual effort of the project. Projects where MRE computed for the proposed approach is lower than the initial effort estimate of the project, indicate that by adding risk exposure the effort estimates of the project improved.

#### 3.3.2 Mean magnitude of relative error

MMRE is the mean of the magnitude of relative error (MRE) for all the projects considered in the dataset. Thus, the formula for MMRE will be:

$$MMRE = \frac{\sum_{i=1}^N MRE}{N} = \frac{\sum_{i=1}^N |E_i - actual\ effort_i|}{actual\ effort_i} / N \quad (3)$$

where  $N$  is the total number of projects in the dataset.

### 3.3.3 Standardized accuracy

The performance evaluation measures MRE and MMRE have been criticized for being biased towards effort estimation techniques resulting in underestimates (Foss et al. 2003; Kitchenham et al. 2001; Korte and Port 2008; Port and Korte 2008; Shepperd and MacDonell 2012; Stensrud et al. 2003). Therefore, integrated effort estimates from the proposed approach are compared with the estimated effort of benchmark models using standardized accuracy (SA) also. Standardized accuracy is calculated based on the formula given below:

$$SA = 1 - \frac{MAR}{MAR_{P0}} \times 100 \quad (4)$$

where  $MAR$  is the mean absolute error i.e., the mean of the absolute difference between the estimated and actual effort estimates of all the projects.

$MAR_{P0}$  is the MAR of the proposed effort estimation method as described in Idri et al. (2018). For performance evaluation, a lower MMRE value or a higher SA value implies a better effort estimation approach.

### 3.3.4 Effect size

Effect size ( $\Delta$ ) is used to determine the reliability of the proposed approach (Idri et al. 2018; Nassif et al. 2019). It can be calculated based on the formula given below:

$$\Delta = \frac{MAR - MAR_{P0}}{\sigma_{P0}} \quad (5)$$

where  $\sigma_{P0}$  refers to the standard deviation of the randomly guessed effort values. Higher value of effect size indicates that the results obtained are more reliable for most of the cases.

### 3.3.5 Coefficient of determination

Coefficient of determination ( $R^2$ ) is used to determine the correlation between the dependent and the independent variables (Nagelkerke 1991). It varies from 0 to 1. A value closer to 1 indicates a strong correlation between the variables. For this research, independent variables are the project cost factor values and the size of the project. Estimated effort will be the dependent variable.

## 4 Experimental settings

The proposed approach was tested on two benchmark effort estimation techniques. CoCoMo II was used for waterfall model dataset and story sizing in story points approach by Ziauddin and Zia (2012) was used for agile model dataset. The calculations were done using MATLAB on a 64-bit personal computer running on Windows 10 operating system. This section will further elaborate on the steps executed in both the cases.

### 4.1 Case I: waterfall delivery model using CoCoMo II

In waterfall delivery model the development of software project is carried out in phases starting with requirement analysis, design, development, testing and then the final product is ready to be put in production. CoCoMo II Model is one of the established models used for effort estimation of projects using waterfall delivery model. This section elaborates the steps carried out using CoCoMo II as the base effort estimation model for initial effort calculation.

- (1) *Data Collection* Experts associated with 45 different software development projects in an Indian IT firm were interviewed over a span of 1 year. A questionnaire based on CoCoMo II model was used to collect the data. Size of the project was measured in terms of kilo lines of code (KLOC). Actual effort spent in the development of the respective projects was measured in terms of Man Months (MM). Man Months is the amount of time (hours) a person spends working on a software project for a month. The values of the project cost factors were taken from the project cost factor values in CoCoMo II Model. There were 5 scale factors and 17 project cost factors identified in the CoCoMo II Model (Dillibabu et al. 2000). All the scale and cost factors have been calibrated at six levels: very low, low, nominal, high, very high and extra high. So, experts filled in the initial level, probability of not meeting the initial level and the alternative level for the scale and project cost factors in the questionnaire.
- (2) *Initial Effort* For the waterfall model dataset, initial effort value ( $E_{i,initial}$ ) for the  $i$ th project was calculated using the CoCoMo II effort estimation formula (Boehm et al. 1997) given below:

$$E_{i,initial} = A \times Size^E \times \prod_{k=1}^n EM_k \quad (6)$$

where  $E = B + 0.01 \times \sum_{j=1}^5 SF_j$ , and  $A$  is a constant whose value can be calibrated according to the

project’s local environment. It has been established that CoCoMo II estimates the software development effort more accurately (Boehm et al. 1997) when the constant  $A$  is calibrated according to the Organisation’s productivity and activity distributions. Since the waterfall model dataset is a small dataset of 45 projects only, this research uses the standard values of  $A$  and  $B$  proposed in the CoCoMo II model. For this research,  $A$  is set to the standard value of 2.94 proposed in the CoCoMo II model.  $B$  is also a constant set at the standard value of 0.90 proposed in the CoCoMo II model.  $EM_k$  denotes the project effort multiplier for the  $k$ th Project cost factor which impacts the estimated effort of the project. There are 17 cost factors ( $n = 17$ ) in the CoCoMo II Model. Size of the project is determined in KLOC.  $SF_j$  are the five scale factors. From the expression for  $E$ , it can be observed that that  $SF_j$ ’s make the effort grow exponentially.  $E_{i,initial}$  is estimated in Man Months. Table 2 lists all the scale factors and project cost factors with their values at different levels in the CoCoMo II Model.

The initial effort ( $E_{i,initial}$ ) of the  $i$ th project is

calculated by substituting the values of project size,  $EM_k, SF_j$  from the waterfall model dataset.

- (3) *Alternative Effort*: The alternative estimated effort  $E_{i,alter}$  of  $i$ th project at the alternative level  $EM_{k,alter}$  of  $k$ th project cost factor is calculated using Eq. (6). The alternative effort is calculated by using the alternative level of the  $k$ th project cost factor while all the other project cost factors remain the same as the initial level.
- (4) *Risk Exposure* Risk exposure of the  $i$ th project  $E_{i,risk}$  is calculated by substituting the values of  $E_{i,initial}, E_{i,alter}$  and  $p_{i,alter}$  in Eq. (2) for all the projects in waterfall model dataset.
- (5) *Integrated Effort Estimate* The integrated effort estimate  $IE_i$  of the  $i$ th project in the waterfall model dataset is calculated using Eq. (1).

### 4.2 Case II: Agile delivery model using story size

Agile delivery model emphasizes on iterative product development, where the software project is developed and delivered continuously in sprints, taking customer feedback after each sprint. This section will elaborate the steps for

**Table 2** CoCoMo II Scale factors and project cost factors (Boehm et al. 1997)

Cost factors	Very low	Low	Nominal	High	Very high	Extra high
RELY	0.82	0.92	1.00	1.10	1.26	
DATA		0.90	1.00	1.14	1.28	
CPLX	0.73	0.87	1.00	1.17	1.34	1.74
RUSE		0.95	1.00	1.07	1.15	1.24
DOCU	0.81	0.91	1.00	1.11	1.23	
TIME			1.00	1.11	1.29	1.63
STOR			1.00	1.05	1.17	1.46
PVOL		0.87	1.00	1.15	1.30	
ACAP	1.42	1.19	1.00	0.85	0.71	
PCAP	1.34	1.15	1.00	0.88	0.76	
PCON	1.29	1.12	1.00	0.90	0.81	
APEX	1.22	1.10	1.00	0.88	0.81	
PLEX	1.19	1.09	1.00	0.91	0.85	
LTEX	1.20	1.09	1.00	0.91	0.84	
TOOL	1.17	1.09	1.00	0.90	0.78	
SITE	1.22	1.09	1.00	0.93	0.86	0.80
SCED	1.43	1.14	1.00	1.00	1.00	
Scale factors	Very low	Low	Nominal	High	Very high	Extra high
PREC	6.2	4.96	3.72	2.48	1.24	0.0
FLEX	5.07	4.05	3.04	2.03	1.01	0.0
RESL	7.07	5.65	4.24	2.83	1.41	0.0
TEAM	5.48	4.38	3.29	2.19	1.1	0.0
PMAT	7.8	6.24	4.68	3.12	1.56	0.0



effort estimation where the initial effort is estimated using story points.

- (1) *Data Collection* Experts associated with 30 different software development projects in an Indian IT firm were interviewed over a span of 1 year. A questionnaire based on effort estimation approach for Agile projects proposed by Ziauddin and Zia (2012) was used to collect the data. The projects followed the Agile delivery model with stories being delivered in sprints. Questionnaire collected responses for one sprint in each project covering story size, story complexity, actual velocity, sprint time, dynamic factors, and frictional factors in the project. Size of the story was rated on a scale of 1–5 based on the effort required for the development of the story. Table 3 provides the guidelines given by Ziauddin and Zia (2012) for determining the story size. Complexity was also rated on a scale of 1–5 depending upon the nature of the work and complexity of technical and non-technical requirements. The complexity of the story is a key factor to the underlying uncertainties in the story effort estimation. Ziauddin has laid down guidelines to determine the complexity of the story on a scale of 1–5 as listed in Table 4. Guidelines provided by Ziauddin in Tables 3 and 4 are generally followed by the effort estimation experts. Actual velocity of the sprint is the actual number of stories delivered during the sprint time. The variable factors which impact the effort estimation of the agile project were categorised into dynamic factors and frictional factors. The impact of these variable factors on the effort estimation of the project is like the impact of cost and scale factors on the effort estimate of the project in the CoCoMo II Model. These variable factors can be thought of as project cost factors which impact the project’s effort

estimate. Dynamic factors were calibrated at 4 levels: normal, high, very high and extra high, by Ziauddin. Friction factors were also calibrated at 4 levels: stable, volatile, highly volatile, and very highly volatile, by Ziauddin. The values of these variable factors at different levels are listed in Tables 5 and 6.

The questionnaire collected data for these dynamic and frictional factors for each story: their initial level, the alternative level, and the probability of not meeting the initial level. The collected data is referred as “The Agile model” dataset.

- (2) *Initial Effort* For the agile model dataset, initial effort value  $E_{i,initial}$  for  $i$ th project was calculated using the model proposed by Ziauddin and Zia (2012). The model estimates the effort for a sprint using the story size, complexity, dynamic and frictional factors. The product is described in the form of user stories creating a product backlog owned by the product owner, usually a representative of the customer for whom the product is being developed (Ambler and Lines 2012). The team delivers the selected user stories at completion of each sprint. As opposed to waterfall model where the manager is responsible for estimating the effort in the planning phase, in agile approach the team members decide on the effort that will go in the delivery of the user story at the beginning of each sprint. Team members estimate the required effort based on their experience, story size, complexity, and project cost factors. The effort is expressed in terms of story points, where one story point corresponds to a day’s work for the team member. The project cost factors might change during the sprint execution leading to the uncertainty in effort estimate by the team member. These project cost factors account for the risks associated with the project which impact the effort estimate of the sprint.

**Table 3** Guidelines to determine story size (Ziauddin and Zia 2012)

Scale value	Guidelines		
	Story size	Broken to smaller stories	Work
5	Extremely large, too large to estimate correctly	Must	Can be taken as a separate project
4	Very large	May be	Dedicated effort for a longer period (more than a week)
3	Moderately large	No	2–5 days
2	Small	No	1–2 days
1	Very small	No	Few hours

**Table 4** Guidelines to determine story complexity (Ziauddin and Zia 2012)

Scale value	Guidelines								
	Complexity	Dependencies	Required skillset	Story description	Story Unknowns	Refactoring	Research	Judgement Calls	Impact outside story
5	Extreme	Multiple	No	Difficult	Many	Large, complex	Extensive	Very difficult	Large
4	Very	Multiple	Not strong	Somewhat difficult	Many	Large, Medium	Yes	Difficult	Moderate
3	Moderate	Moderate	Strong	Somewhat difficult	Very few	Medium	Little	Few difficult	Minimal
2	Moderate	Few	Very strong	Medium	No	Low	Very little	Easy	No
1	Easy	No	Very strong	Easy	No	No	No	No	No

**Table 5** Friction factor values (Ziauddin and Zia 2012)

Friction factor	Stable	Volatile	Highly volatile	Very highly volatile
Team composition	1	0.98	0.95	0.91
Process	1	0.98	0.94	0.89
Environmental factors	1	0.98	0.98	0.96
Team dynamics	1	0.98	0.91	0.85

**Table 6** Dynamic factor values (Ziauddin and Zia 2012)

Variable factor	Normal	High	Very high	Extra high
Expected team changes	1	0.98	0.95	0.91
Introduction of new tools	1	0.99	0.97	0.96
Vendor's defect	1	0.98	0.94	0.90
Team member's responsibilities outside the project	1	0.99	0.98	0.98
Personal issues	1	0.99	0.99	0.98
Expected delay in stakeholder response	1	0.99	0.98	0.96
Expected ambiguity in details	1	0.98	0.97	0.95
Expected changes in environment	1	0.99	0.98	0.97
Expected relocation	1	0.99	0.99	0.98

Steps given below were followed to calculate the initial effort estimate.

- (a) *Effort for a story* For each story, the effort dispensed towards the development of the story was calculated using the formula given below:

$$ES(\text{Effort for a story}) = \text{story size} \times \text{story complexity} \quad (7)$$

This effort estimate of the story is expressed in story points.

- (b) *Effort for the whole sprint* The estimated effort for all the stories in the sprint is added to get

the effort estimate of the sprint, using the equation given below:

$$E(\text{Effort for whole sprint}) = \sum_{i=1}^n ES_i \quad (8)$$

where n is the number of stories being delivered in the sprint. Now, the effort for the whole sprint is available in story points.

- (c) *Variable Factors* From the agile project dataset, initial values of the Frictional and dynamic factors were used to calculate the impact of variable factors on the initial effort estimate. The impact was calculated using the formula given below:

$$D(\text{Variable Forces}) = \prod_{k=1}^4 \text{Friction factors}_i \times \prod_{m=1}^9 \text{Dynamic factors}_j \tag{9}$$

- (d) *Agile Velocity* In this step, the velocity for each sprint in the project was determined based on the estimated sprint effort (E), sprint time (T) and variable forces (D) in the sprint using the formula given below:

$$V(\text{Velocity}) = \left(\frac{E}{T}\right)^D \tag{10}$$

In Agile delivery, the focus is to improve and stabilize the velocity of the project over various sprints. This stability in velocity will depend on the project cost factors, in this case dynamic and friction factors. These factors change often during the execution of the sprint thus leading to uncertainties in the estimated effort (Parvez 2013). These uncertainties are the risks associated with the project which need to be addressed during the project execution. The effort that goes into the control and mitigation of these risks will be accounted in the estimated effort in the proposed approach.

- (e) *Initial Effort Estimate* Now, using the velocity of the sprint, the proposed approach calculates the initial effort estimate for the sprint using the formula:

$$E_{i,\text{initial}} = E = (V)^{\frac{1}{D}} \times T \tag{11}$$

The estimated effort will be in Days, which is the number of days estimated for delivering the stories in the sprint.

- (3) *Alternative Effort* The alternative effort  $E_{i,\text{alter}}$  required for the alternative values of dynamic and frictional factors is calculated by repeating the step 2 and substituting the alternative value of the  $m$ th dynamic factor and  $k$ th frictional factor in Eq. (11) while keeping all other factors at the initial level.
- (4) *Risk Exposure* Risk exposure  $E_{i,\text{risk}}$  of the  $i$ th project is calculated by substituting the values of  $E_{i,\text{initial}}$ ,  $E_{i,\text{alter}}$  and  $p_{i,\text{alter}}$  in Eq. (2) for all the projects in agile model dataset.
- (5) *Integrated Effort Estimate* The integrated effort estimate  $IE_i$  of the  $i$ th project in the agile model dataset is calculated using Eq. (1).

## 5 Results and analysis

In this section the experimental results obtained are discussed and analysed for both the models.

### 5.1 Case I: waterfall delivery model using CoCoMo II

Variations in the MRE values of both the approaches are depicted project wise in Fig. 3. Out of the 45 projects, the proposed approach gave more accurate effort estimates for 62% (28) projects. These 28 projects (Project Ids: 1, 2, 5, 6, 8, 9, 10, 12, 13, 14, 15, 18, 19, 21, 22, 26, 27, 29, 30, 31, 33, 35, 38, 40, 41, 42, 43 and 44) had lower MRE values for the proposed approach as compared to the MRE values of the CoCoMo II model. CoCoMo II model estimated the effort more accurately for 35% (16) projects. Projects with ids 4, 7, 11, 16, 17, 20, 23, 24, 25, 28, 32, 34, 36, 37, 39 and 45 had lower MRE values for CoCoMo II model as compared to the proposed approach.

For one project (Id 3), both CoCoMo II model and the proposed approach had MRE value of 0.034. This implies that the high risk project cost factors were balanced by the low risk cost factors in the proposed model. From Fig. 3, it is evident that the project wise integrated effort estimates show large variations for both CoCoMo II and the proposed approach with respect to MRE. Therefore, the performance comparison of CoCoMo II and proposed approach is also done based on MMRE, SA, effect size and  $R^2$  values obtained on the waterfall model dataset, as shown in Fig. 4.

The MMRE value of the proposed approach (0.1837) is slightly lower than that of the CoCoMo II Model (0.2155). The proposed approach has not only a lower MMRE value than the CoCoMo II model, but also a higher value of SA (0.845) than the CoCoMo II model (0.829), indicating that the proposed approach estimates effort more accurately than the CoCoMo II model. Additionally, a higher value of effect size for the proposed model (0.596) as compared to the CoCoMo II model (0.521) indicates that the proposed approach will give better effort estimates for most of the cases implying, more reliability than the CoCoMo II model. Proposed approach has a higher  $R^2$  value (0.729) than the CoCoMo II model (0.581), indicating that the risk exposure has a considerable impact on the effort estimation process, thereby making the effort estimates less biased. Table 7 has the actual effort, estimated effort using CoCoMo II model and the integrated effort estimates for all the projects in the waterfall model dataset. Out of 45 projects, 11 projects (P5, P8, P16, P27, P28, P30, P31, P36, P38, P43 and P45) have integrated effort estimates lower than the corresponding CoCoMo II effort estimates in man months. Remaining 34 projects have higher integrated

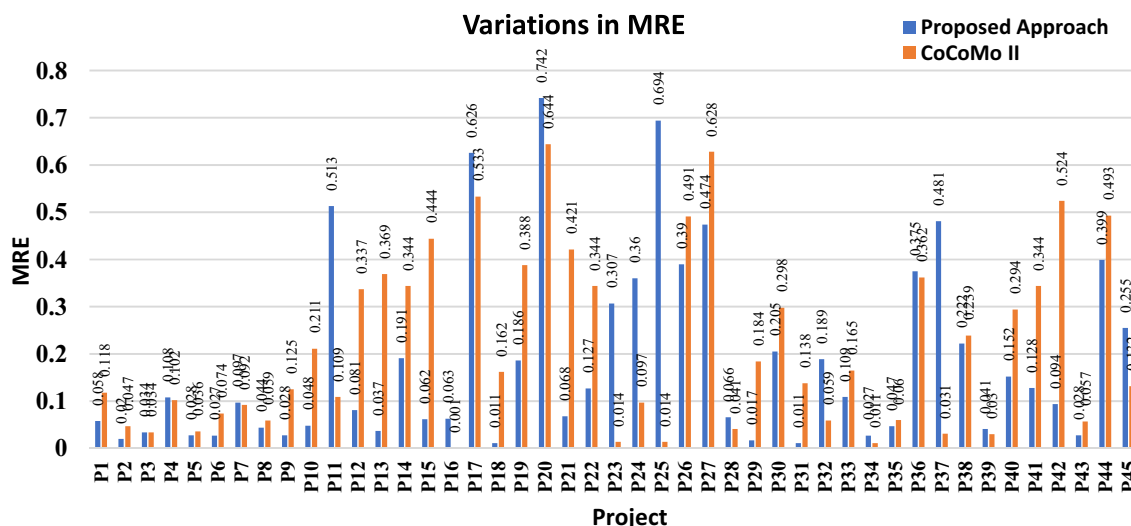
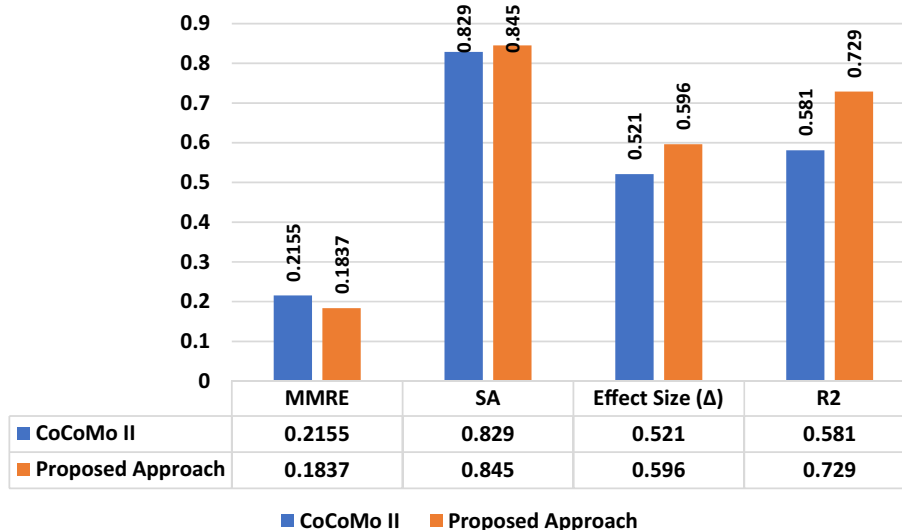


Fig. 3 MRE values of CoCoMo II and proposed approach on waterfall model dataset

Fig. 4 Comparison of proposed approach and CoCoMo II on the waterfall model dataset



effort estimates than the corresponding CoCoMo II effort estimates. This shows that risk exposure can either decrease or increase the effort estimate of a project based on the pessimistic or optimistic assumptions of project cost factor values.

**5.2 Case II: Agile delivery model using story size**

The effort estimates for the projects in agile model dataset obtained using Ziauddin approach and proposed approach are compared based on MMRE, SA, effect size and R<sup>2</sup>. Variations in the MRE values of both the approaches are depicted in Fig. 5. The MRE for the proposed approach varies from 0.04 to 0.63, whereas the MRE for the Ziauddin model lies between 0.0 and 0.8. Figure 5 shows a

variation of 59% in the MRE values of the proposed approach and a variation of 80% in the Ziauddin model.

Out of the 30 projects, the proposed approach gave more accurate effort estimates for 43% (13) projects. These 13 projects (Project Ids: 1, 2, 3, 4, 10, 11, 17, 18, 24, 25, 26, 29 and 30) had lower MRE values for the proposed approach as compared to the MRE values of the Ziauddin model. The difference in the MRE values was in the range of 4% to 17%. Ziauddin model estimated the effort more accurately for 50% (15) projects. Projects with ids 5, 6, 8, 9, 12, 13, 14, 16, 19, 20, 21, 22, 23, 27 and 28 had lower MRE values for Ziauddin model as compared to the proposed approach. The difference in MRE values for these projects was in the range of 1% to 25%.

It can be noted that the variability of MRE values for the proposed model is 13% as compared to 24% for Ziauddin

**Table 7** Waterfall model dataset effort estimates in man months

Id	Actual Effort	CoCoMo II	Integrated effort estimate
P1	1634	1440.80	1728.74
P2	700	666.95	713.99
P3	3987	3852.07	4123.87
P4	450	404.29	498.52
<b>P5</b>	<b>2608</b>	<b>2702.00</b>	<b>2535.76</b>
P6	3567	3304.39	3661.62
P7	2256	2048.08	2473.87
<b>P8</b>	<b>912</b>	<b>965.82</b>	<b>952.08</b>
P9	2879	2518.22	2959.18
P10	2435	1922.34	2318.95
P11	1456	1297.09	2202.69
P12	2234	1480.34	2414.97
P13	3200	2020.05	3080.99
P14	3567	2339.52	4248.97
P15	3678	2044.51	3905.35
<b>P16</b>	<b>2759</b>	<b>2756.71</b>	<b>2585.53</b>
P17	3015	4621.11	4901.98
P18	3459	2897.42	3497.34
P19	2435	1490.16	1981.09
P20	859	1412.51	1496.22
P21	3147	1821.52	2934.56
P22	1987	1304.26	2239.51
P23	4567	4632.83	5968.33
P24	3629	3275.44	4936.89
P25	2897	2937.63	4906.66
P26	2453	1247.94	1496.43
<b>P27</b>	<b>3786</b>	<b>6163.27</b>	<b>5581.17</b>
<b>P28</b>	<b>2687</b>	<b>2577.10</b>	<b>2510.67</b>
P29	2937	2395.47	2887.85
<b>P30</b>	<b>2874</b>	<b>3730.24</b>	<b>3463.45</b>
<b>P31</b>	<b>3384</b>	<b>3851.66</b>	<b>3346.51</b>
P32	3287	3480.28	3906.64
P33	2845	2376.82	2534.23
P34	3504	3542.16	3598.09
P35	2134	2005.14	2034.21
<b>P36</b>	<b>2739</b>	<b>1746.16</b>	<b>1710.69</b>
P37	2469	2391.86	3657.09
<b>P38</b>	<b>3200</b>	<b>3963.39</b>	<b>3911.64</b>
P39	2489	2562.53	2590.45
P40	1999	1411.74	1694.45
P41	2598	1704.90	2264.62
P42	2876	1369.66	2605.07
<b>P43</b>	<b>2309</b>	<b>2441.11</b>	<b>2374.31</b>
P44	1784	905.34	1072.13
<b>P45</b>	<b>2792</b>	<b>3159.18</b>	<b>3502.64</b>

Bold values highlight negative risk exposure for the projects. Integrated effort estimate is less than the CoCoMo II effort estimate

model. For two projects with ids 7 and 15, both Ziauddin model and the proposed approach had the same MRE values, 0.35 and 0.14, respectively. This implies that in the proposed model, the corresponding risk exposure for the two projects was insignificant.

Both the approaches were also compared based on the MMRE, SA, effect size and  $R^2$  values obtained on the agile model dataset, as shown in Fig. 6. The MMRE value of the proposed approach (0.282) is slightly lower than that of the Ziauddin Model (0.288). The proposed approach has not only a lower MMRE value than the Ziauddin model, but also a higher value of SA (2.14) than the Ziauddin model (1.85), indicating that the proposed approach estimates effort more accurately than the Ziauddin model. Additionally, a higher value of effect size for the proposed model (0.713) as compared to the Ziauddin model (0.603) indicates that the proposed approach will give better effort estimates for most of the cases, implying more reliability than the Ziauddin model. Proposed approach has a higher  $R^2$  value (0.102) than the Ziauddin model (0.018), indicating that the risk exposure has a considerable impact on the effort estimation process, thereby making the effort estimates less biased. Table 8 has the actual effort, estimated effort using Ziauddin approach and the integrated effort estimates for all the projects in the agile model dataset.

### 5.3 Revisiting research questions

**RQ1** How good is the proposed function for calculating integrated effort estimates of software projects?

To answer RQ1, the proposed function for calculating integrated effort estimates has been tested on two datasets: waterfall and agile. The accuracy of the integrated effort estimates has been compared with the initial estimates based on MMRE, SA, Effect Size and  $R^2$ . The experimental results show that the proposed function gave better results for all the four parameters for both the datasets. There is strong evidence to claim that the proposed integrated effort estimates are more accurate and reliable than the initial effort estimates done without considering the effort spent on risk planning & mitigation.

## 6 Threats to validity

This section discusses threats to validity and limitations of the results presented. There are primarily two potential threats as discussed below:

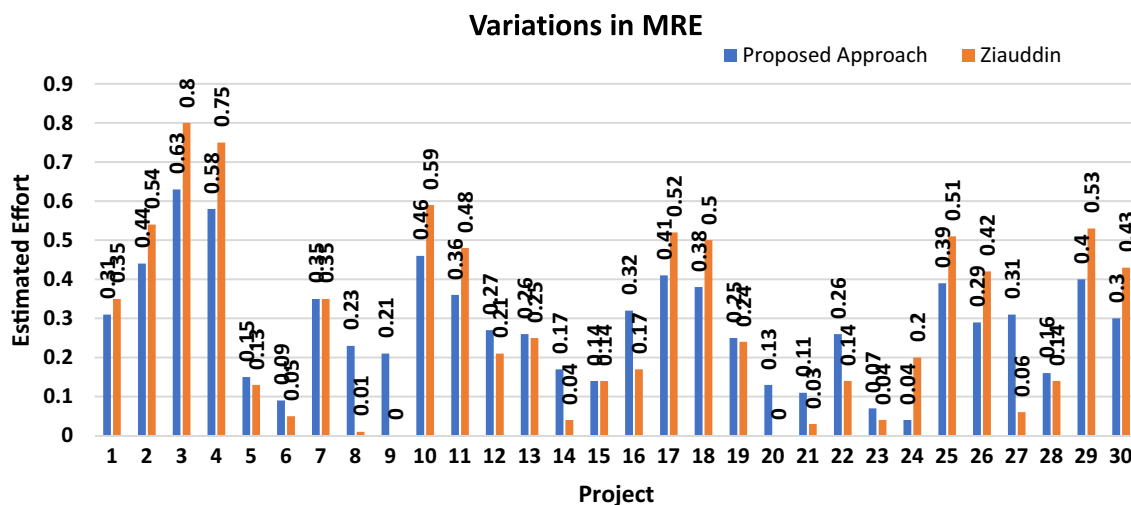
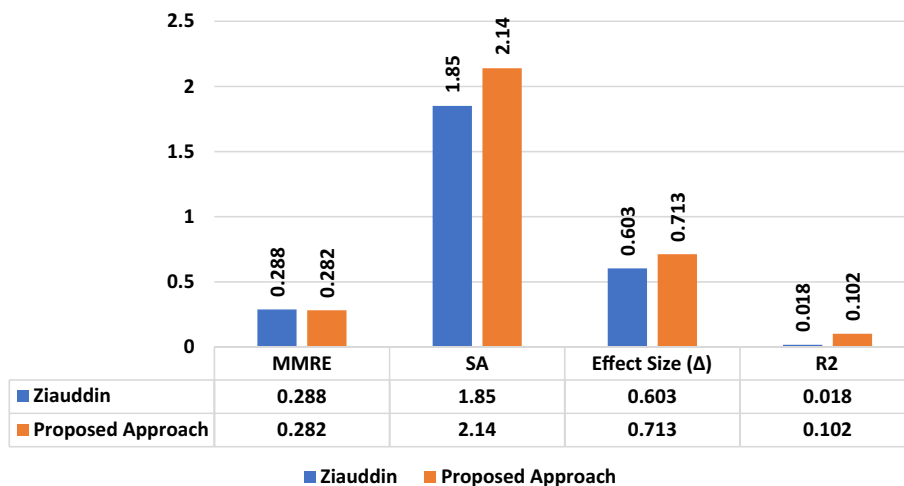


Fig. 5 MRE values of proposed approach and Ziauddin approach on Agile model dataset

Fig. 6 Comparison of proposed approach and Ziauddin model on the Agile model dataset



6.1 External validity

External validity is concerned with generalization of results obtained. Threats to external validity are conditions that limit the ability to generalize the results of the experimental research conducted in industrial practices (Wohlin et al. 2012).

The proposed model can be adapted for other well established effort estimation models such as analogy-based effort estimation, function points, playing poker, expert judgement and use case points, for both waterfall and agile delivery projects. Hence, the integrated effort estimates can also be obtained for the different estimation techniques mentioned above. However, the data available for these effort estimation techniques will need to be transformed according to the formula (Eq. 1) for calculating integrated effort estimates. The proposed approach needs the breakup of the total effort in terms of contribution of project cost factors that impact the effort required to develop the

project. This can be achieved in consultation with the experts involved in delivery of the project.

6.2 Internal validity

Threats to internal validity are influences that can affect the dependent variables with respect to causality, without the researcher’s knowledge (Wohlin et al. 2012).

The threat of biases that may get built into the probability estimates of the alternate project cost factor values is a matter of concern. However, by considering more than one alternative for each project cost factor or by computing optimum values of the probabilities, can get rid of the biases.

Data has been collected for already completed projects. The project cost factor values, their alternative values, and their alternative values may be biased. For new projects, the project conditions will be different and cannot be determined beforehand, the project cost factor values might

**Table 8** Agile model dataset effort estimates in story points

Project	Actual Effort	Ziauddin estimated effort	Integrated effort estimate
1	64	86.49230623	83.60254056
2	76	117.1164834	109.1452747
3	81	145.9146474	132.0753141
4	68	119.1527558	107.4204655
5	66	57.66670951	56.37771335
6	67	63.59278837	60.88423559
7	67	43.43547709	43.27209018
8	68	67.05171397	52.21070975
9	56	56.27626776	44.25048377
10	60	95.27370841	87.73995771
11	58	86.01565676	78.676606
12	62	48.89733619	45.28741067
13	65	48.8169556	47.96385472
14	63	60.6509433	52.03628395
15	65	56.06236739	55.76618453
16	66	77.10079698	45.07614048
17	65	98.66454163	91.55481726
18	62	93.10650035	85.71088812
19	76	57.59757584	57.1290063
20	75	75.05298685	65.01085258
21	73	70.55184923	64.72607561
22	72	61.62483617	53.08436635
23	73	70.14982105	67.56120822
24	74	89.07373303	76.97409167
25	60	90.82081777	83.18574934
26	65	92.41713	83.58270393
27	62	58.08273498	42.89177877
28	55	47.45795162	46.40748415
29	60	91.56107642	83.83622669
30	59	84.5248573	76.74826971

differ significantly from the initial level to the alternative level.

## 7 Conclusion and future work

Literature survey had highlighted the need to integrate the risk management process with software effort estimations, but none of the techniques had actually counted the effort that goes into risk management into the effort estimate of the project. The proposed approach shows that incorporating this effort for mitigating the risks associated with the project into the effort estimate gives a more accurate estimate of the actual effort that will be required to deliver the project successfully. Besides improving the accuracy of the effort estimate, the proposed approach also collects comprehensive data related to the project cost factors which impact the project. This data can be utilized by the

organizations to help them better understand their organization behaviour, the factors that impact the project delivery and the required steps to control those factors. The proposed approach gives more comprehensive effort estimates which provide advance insights for planning and execution of projects. The research can be extended to include the impact of more than one alternative project cost factor values on the risk exposure of the project. The proposed approach could also be tested on multi company data.

**Funding** Not applicable.

**Availability of data and material** The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

**Code availability** Not applicable.

## Declarations

**Conflict of interest** Purna Singal, Prabha Sharma and A. Charan Kumari declare that they have no conflict of interest.

## References

- Ambler SW, Lines M (2012) *Disciplined agile delivery: a practitioner's guide to agile software delivery in the enterprise*. IBM Press, Indianapolis
- Aslam W, Ijaz F, Lali MIU, Mehmood W (2017) Risk aware and quality enriched effort estimation for mobile applications in distributed agile software development. *J Inf Sci Eng* 33:1481–1500
- Barki H, Rivard S, Talbot J (2001) An integrative contingency model of software project risk management. *J Manag Inf Syst* 17:37–69
- Bhavsar K, Shah V, Gopalan S (2020) Scrumbanfall: an agile integration of scrum and kanban with waterfall in software engineering. *Int J Innov Technol Explor Eng* 9:2075–2084
- Boehm B (1989) Software risk management. In: *European software engineering conference*, pp 1–19
- Boehm B, Abts C, Clark B, Devnani-Chulani S (1997) *COCOMO II model definition manual*. The University of Southern California
- Briand LC, El Emam K, Bomarius F (1998) COBRA: a hybrid method for software cost estimation, benchmarking, and risk assessment. In: *Proceedings of the 20th international conference on Software engineering*, pp 390–399
- Capretz LF (2013) Software project risk assessment and effort contingency model based on cost factors. *J Comput Model* 3:113
- Dalal S, Dahiya N, Jaglan V (2018) Efficient tuning of COCOMO model cost drivers through generalized reduced gradient (GRG) nonlinear optimization with best-fit analysis. In: *Progress in advanced computing and intelligent engineering*. Springer, pp 347–354
- Dillibabu R, Krishnaiah K (2005) Cost estimation of a software product using COCOMO II. 2000 model—a case study. *Int J Project Manage* 23:297–307
- El Koutbi S, Idri A (2018) Software effort estimation risk management over projects portfolio. *Comput Inf Sci* 11:45–76
- Foss T, Stensrud E, Kitchenham B, Myrtveit I (2003) A simulation study of the model evaluation criterion MMRE. *IEEE Trans Softw Eng* 29:985–995
- Gilb T (1985) Evolutionary Delivery versus the “waterfall model”. *ACM Sigsoft Softw Eng Notes* 10:49–61
- Higuera RP, Haimes YY (1996) *Software Risk Management*. Carnegie-mellon univ pittsburgh pa software engineering Inst
- Huang S-J, Lin C-Y, Chiu N-H (2006) Fuzzy decision tree approach for embedding risk assessment information into software cost estimation model. *J Inf Sci Eng* 22:297–313
- Idri A, Abnane I, Abran A (2018) Evaluating Pred (p) and standardized accuracy criteria in software development effort estimation. *J Softw Evol Process* 30:e1925
- Jantzen K (2006) Estimating the effects of project risks in software development projects
- JøRgensen M, Teigen KH, MoløKken K (2004) Better sure than safe? Over-confidence in judgement based software development effort prediction intervals. *J Syst Softw* 70:79–93
- Jørgensen M (2010) Identification of more risks can lead to increased over-optimism of and over-confidence in software development effort estimates. *Inf Softw Technol* 52:506–516
- Kansala K (1997) Integrating risk assessment with cost estimation. *IEEE Softw* 14:61–67
- Kitchenham B (1998) The certainty of uncertainty. In: *European software measurement conference FEMSA*
- Kitchenham B, Linkman S (1997) Estimates, uncertainty, and risk. *IEEE Softw* 14:69–74
- Kitchenham BA, Pickard LM, MacDonell SG, Shepperd MJ (2001) What accuracy statistics really measure. *IEE Proc Softw* 148:81–85
- Korte M, Port D (2008) Confidence in software cost estimation results based on MMRE and PRED. In: *Proceedings of the 4th international workshop on Predictor models in software engineering*, pp 63–70
- Madachy RJ (1997) Heuristic risk assessment using cost factors. *IEEE Softw* 14:51–59
- Manalif E (2013) Fuzzy Expert-COCOMO risk assessment and effort contingency model in software project management
- Martin RC, Martin M (2006) *Agile principles, patterns, and practices in C#* (Robert C. Martin). Prentice Hall PTR.
- Morgenshtern O, Raz T, Dvir D (2007) Factors affecting duration and effort estimation errors in software development projects. *Inf Softw Technol* 49:827–837
- Nagelkerke NJ (1991) A note on a general definition of the coefficient of determination. *Biometrika* 78:691–692
- Nassif AB, Azzeh M, Idri A, Abran A (2019) Software development effort estimation using regression fuzzy models. *Comput Intell Neurosci* 2019
- Parvez AWMM (2013) Efficiency factor and risk factor based user case point test effort estimation model compatible with agile software development. In: *2013 international conference on information technology and electrical engineering (ICITEE)*, pp 113–118
- Port D, Korte M (2008) Comparative studies of the model evaluation criterions MMRE and PRED in software cost estimation research. In: *Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement*, pp 51–60
- Ramakrishnan N, Girijamma H, Balachandran K (2019) Enhanced process model and analysis of risk integration in software effort estimation. In: *2019 international conference on smart systems and inventive technology (ICSSIT)*, pp 419–422
- Shepperd M, MacDonell S (2012) Evaluating prediction systems in software project estimation. *Inf Softw Technol* 54:820–827
- Stensrud E, Foss T, Kitchenham B, Myrtveit I (2003) A further empirical investigation of the relationship between MRE and project size. *Empir Softw Eng* 8:139–161
- Sureshchandra K, Shrinivasavadhani J (2008) Moving from waterfall to agile. In: *Agile 2008 conference*, pp 97–101
- Tawosi V, Sarro F, Petrozziello A, Harman M (2021) Multi-objective software effort estimation: a replication study. *IEEE Transact Softw Eng*
- Wohlin C, Runeson P, Höst M, Ohlsson MC, Regnell B, Wesslén A (2012) *Experimentation in software engineering*. Springer, Berlin
- Ziauddin SKT, Zia S (2012) An effort estimation model for agile software development. *Adv Comput Sci Its Appl (ACSA)* 2:314–324

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.