

Enhancing software development cost control by forecasting the cost of rework: preliminary study

Tarig Ahmed Khalid, Eng-Thiam Yeoh

Faculty of Computing and Informatics, Multimedia University, Cyberjaya, Malaysia

Article Info

Article history:

Received May 12, 2020

Revised Jul 13, 2020

Accepted Jul 27, 2020

Keywords:

Earned value management

EVM

Software cost control

Software cost estimation

Software rework

ABSTRACT

Industrial reports show massive cost overruns associated with software projects. The cost of software reworks constitutes a large portion of the overall cost, reflecting a substantial challenge in cost control. Earned value management (EVM) is the most recognized model for project cost control. However, it shows many limitations in forecasting the software project cost, leading to a considerable challenge in cost control. Nevertheless, the major EVM limitation found is its inability to forecast the cost of software rework. This research investigated the factors affecting this limitation and suggests an enhanced EVM model. The significant contribution of this research is its incorporation of software-related factors into the EVM model. We introduced the software rework index (SRI), which is incorporated into the traditional EVM model to enhance its predictability of the software project cost at completion, including the rework cost. We defined the SRI in terms of two factors: product functional complexity and the team competency. Finally, we evaluated the proposed model using a dataset drawn from five actual projects. The results showed a significant enhancement in forecasting accuracy.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Tarig Ahmed Khalid

Faculty of Computing and Informatics

Multimedia University

Cyberjaya 63100, Malaysia

Email: t.a.khalid@ieee.org

1. INTRODUCTION

Delivering a quality product within the budget is a core objective for project managers. Meeting this aim in the software industry is challenging. Organizations waste billions of dollars on failed and challenging software projects [1]. The Chaos Report stated that 56% of software projects experienced cost-overrun throughout the year 2011 to 2015 and increased to 65% by the end of 2017 [2, 3], showing the complexity of the software development process. This complexity is due to the impacts of many factors, such as requirements complexity and volatility, software measurement difficulties, unplanned rework, and developers' competency levels [4-10]. In practice, the development and testing teams, mutually, conduct multiple cycles of reworks and tests until the software is compliant with the customers' requirements [6, 11]. Early detection of defects is of less cost compared to late detection. The delayed repairs may make the cost higher as 300 to 1000 times the cost of early corrections [12, 13]. In general, the cost of software reworks may constitute 30% to 50% of the total cost [10, 14-16]. Technical complexity makes early identification of issues challenging [12]. The rise of the Internet-of-Things and system of systems has increased the complexity of software development in terms of attributes such as code complexity, requirement complexity and volatility, and architectural complexity [6, 11, 17]. Recent studies stated that 78% to 82% of the reasons

behind software reworks are due to requirement-related problems [12, 14]. Since people develop the software, the human side cannot be ignored in favor of technical attributes. Many of the software development challenges are due to human factors [4, 9].

To control software project cost, project management and software engineering best practices recommend the earned value management (EVM) model as the most recognized cost control tool. The institute of electrical and electronics engineers (IEEE), project management institute (PMI), and international organization for standardization (ISO) have all included the EVM as part of their project management standards [11, 18-21]. The EVM model provides an integrated scope, schedule, and cost baseline [18]. It estimates the total project cost in terms of a variable called the estimate at completion (EAC). The EAC is derived in terms of two indices called the cost performance index (CPI) and Schedule Performance Index (SPI). The CPI and SPI are periodically calculated in terms of many variables among them is the percent completion (%C), which represents the percent of work done since the project kick-off.

However, Jurison [22] argued that there are difficulties in managing software projects due to software intangibility, complexity, and volatility. The EVM model is not designed to deal with such dimensions. It is incapable of accurately forecasting the EAC of software projects due to four factors. First, the EAC does not include the cost of software rework. Second, the calculation of its indices depends on the progress measurement. However, measuring intangible products such as software is challenging. Although %C could be subjective and uncertain in software projects, the EVM parameters are deterministic [23, 24]. Third, the EVM does not measure the quality as it only accepts the deliverables upon completion [25]. However, poor quality will lead to further rework and cost overrun. Fourth, many software projects failed or exceeded their baselines due to the human side, which is overlooked by the EVM model [26]. During software development, it is challenging to isolate human factors from technical ones [9].

This study suggests enhancing the EVM cost predictability by including the cost of rework and incorporating indigenous software factors such as product characteristics and team competency. Moreover, it reduces the subjectivity in software progress measurement by adopting the function point analysis (FPA). We hypothesized that the error in estimating the EAC using the proposed EVM is less than the error obtained when using the traditional EVM. We evaluated the proposed model using data collected from actual projects. The rest of the paper is organized as follows. Section 2 covers the related research. Section 3 provides a detailed description of the research method. Section 4 illustrates the results while Section 5 shows the discussion. Conclusions and future work are discussed in Section 6.

2. RELATED WORKS

Langsari and Sarno [27] suggested hybrid models by using the COCOMO II and fuzzy logic to estimate the project cost. Esteki et al. [28] suggested the augmentation of the cost control methods with a risk management framework. De Souza et al. suggested statistical approaches to enhance EAC accuracy by using historical data [29]. Acebes et al. [30] used the Monte Carlo simulation combined with the EVM model to conduct a project risk analysis. They argued that the traditional EVM model results in optimistic indices leading to an inaccurate EAC. Other researchers proposed quality based EVM models by incorporating quality performance indices based on historical data [31-33]. Although the introduction of these indices is a significant enhancement, the proposed models are difficult to generalize due to their dependency on historical data, which defies the fundamental attribute of projects, i.e., uniqueness. Also, the data unavailability may make statistical techniques challenging to apply.

The subjective nature of software progress measurements inspired researchers to integrate the FPA with the EVM model to provide quantitative measurements [34-36]. Other studies suggested the Agile EVM, which expresses the EVM variables in terms of Agile variables such as story points and the number of sprints [37-40]. While the Agile EVM makes it easy to apply the traditional EVM, it does not address limitations such as the lack of quality indicators and the human factors. Efe and Demiros [10] developed a change management model incorporating the cost of rework and changes into the EVM model. The suggested model obtained promising results in forecasting the software cost. Pracharasniyom et al. [41] and Vargas [42] proposed modified EVM models that assess individual project members' performance. However, both studies did not incorporate real human attributes into their proposed models as they used traditional EVM data. Our literature review indicates a considerable knowledge gap in relating the human factors to the EVM model. We found only limited studies investigating the importance of the human side of software development [8, 43-47].

3. RESEARCH METHOD

3.1. Conceptual modelling

We constructed a conceptual model that monitors the project performance via three variables: the cost, schedule, and predicted rework. We suggested that the software rework mainly depends on two major factors: the product complexity and developers' competency. Moreover, we used fuzzy logic to represent the software rework as the amount of software reworks depends on three distinct probabilities:

- a) No reworks if the deliverables are wholly accepted
- b) High level of unplanned reworks if the deliverables are entirely rejected
- c) Reworks ranging from low to high levels if the deliverables are partially accepted

Accordingly, if a low-competency team develops sophisticated software, the volume of defects will likely be significant, leading to a high level of software reworks. Conversely, the reworks level may be a minimum if a remarkably high competency team is assigned to develop software of low complexity. We also suggested the fuzzy logic to mitigate the subjectivity in the assessment of developers' competencies. The conceptual model is shown in Figure 1.

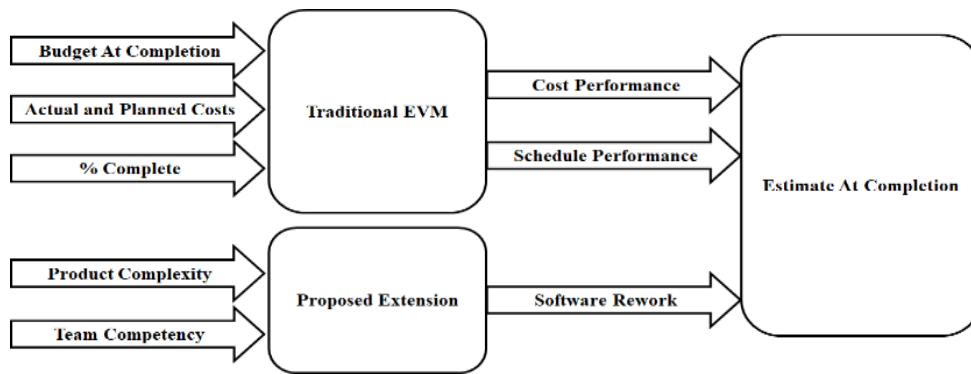


Figure 1. Proposed EVM conceptual model

3.2. Mathematical modelling

We constructed a mathematical model consisting of two parts: the traditional EVM and the proposed extension. In the traditional part, we adopted the FPA in measuring the software size, requirement volatility, and the software developed so far. Then we used these sizes to calculate the %C. We determined the software rework in terms of the functional complexity and team competency, as described by the conceptual model. We introduced the functional complexity index (FCI), which defines the software product's functional complexity. The FCI value was derived from the system functions using the FPA. Moreover, we introduced the team competency index (TCI) to model the team competency. The TCI was derived from the software development competency framework (SDCF), as suggested by Khalid and Yeoh [8]. The two indices were related in terms of fuzzy logic to calculate the software rework index (SRI). Finally, we expressed the EAC in terms of the CPI, SPI, and SRI, as described in Figure 2. The techniques used to address the limitations of the traditional model are described in Table 1.

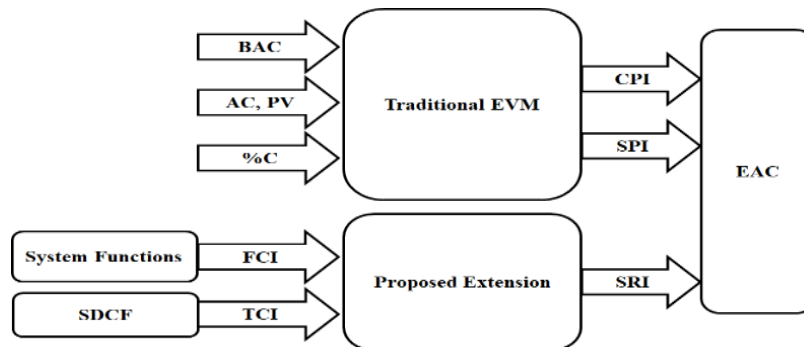


Figure 2. Proposed mathematical model

Table 1. Solution approaches to the proposed mathematical model

Limitation	Addressed By	Adopted Technique
Uncertainty in measuring %C	Quantifying the software sizing	Function Point Analysis
EAC forecast does not include the software reworks	Introducing the Software Rework Index (SRI)	Fuzzy Logic
Lack of addressing the human factors	Introducing the Team Performance Index (TCI)	Fuzzy Logic Function Point Analysis
Lack of addressing the technical software characteristics	Introducing the Functional Complexity Index (FCI)	Fuzzy Logic Function Point Analysis

We calculated the traditional EVM indices according to the following equations.

$$C\% = \frac{\text{Developed Size}}{\text{Requirement Volatility Size} + \text{Total Planned Size}} \times 100 \tag{1}$$

Where c% is the percent completion of the project work.

$$EV = \%C \times BAC \tag{2}$$

Where EV is the earned value and BAC is the budget at completion.

$$CPI = \frac{EV}{AC} \tag{3}$$

$$SPI = \frac{EV}{PV} \tag{4}$$

Where CPI and SPI are the cost performance index and schedule performance index, respectively. EV, AC, and PV are the earned value, actual cost, and planned value, respectively.

$$EAC_{T1} = \frac{BAC}{CPI} \tag{5}$$

Where EAC_{T1} is the cost estimate at completion considering only the cost performance.

$$EAC_{T2} = AC + \frac{BAC - EV}{CPI \times SPI} \tag{6}$$

Where EAC_{T2} is the cost estimate at completion considering bot the cost performance and schedule performance. Both AC and PV were calculated in person-days. On the other hand, we used fuzzy logic to model the relationship between the extended model variables. We used Mamdani’s fuzzy inference system (FIS) as it is the most used FIS due to its intuitive nature and its suiting to the human input [48]. The fuzzy model is shown in Figure 3.

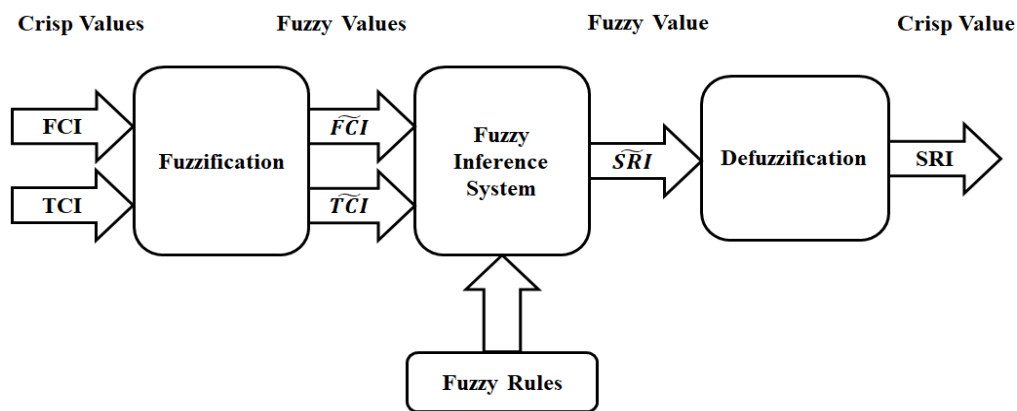


Figure 3. Proposed fuzzy model

We implemented the proposed fuzzy model according to the following steps.

Step 1: SRI definition and fuzzification

We represented the software rework level as a ratio of the cost of rework (COR) to the cost of development (COD). This ratio is called the software rework index (SRI). Hence,

$$SRI = \frac{COR}{COD} \quad (7)$$

The value of SRI is fuzzy, depending on the quality status of deliverables after testing. If the deliverable is entirely accepted, then SRI equals zero. Conversely, if the deliverable is entirely rejected, then SRI is expected to be a sizable number. If the deliverable is partially accepted, then $SRI > 0$ ranging from low to high values. The fuzzy membership of SRI is shown in terms of trapezoidal shapes, as in Figure 4. The fuzzy numbers are selected according to the percentages of reworks stated by Micro Focus [14].

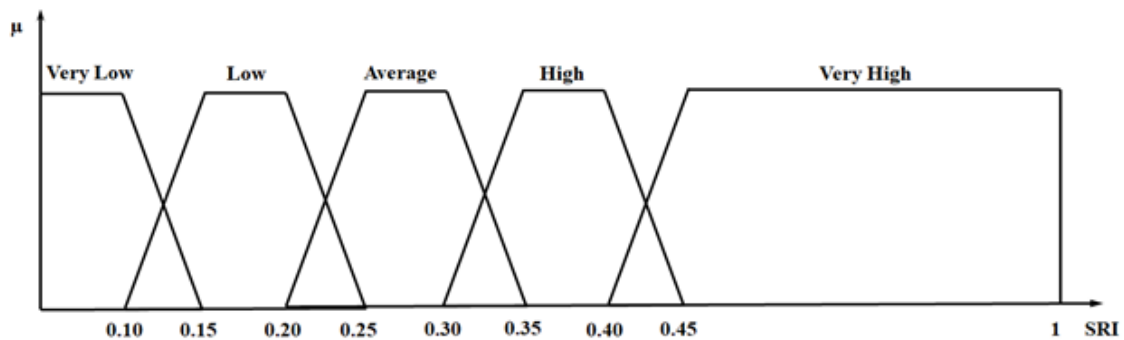


Figure 4. SRI fuzzy membership

Step 2: FCI definition and fuzzification

According to the FPA, each software, regardless of its function type, is assigned a specific weight representing its complexity. This weight could be low, average, or high. The FCI of each software function is given by:

$$FCI = \frac{FP}{FP_{max}} \times 100 \quad (8)$$

Where FP is the weight assigned to the software function (low, average, or high), FP_{max} corresponds to the “High” weight. The system function complexity is defined by calculating the number of data element types (DET), the number of record element types (RET), and the number of file types referenced (FTR), as described by IFPUG [49]. At a specific checkpoint in the project lifecycle, the overall FCI is calculated as:

$$FCI = \frac{\sum_{i=1}^N FP_i}{\sum_{i=1}^N FP_{max_i}} \times 100 \quad (9)$$

Where FP_i is the value of the function points developed by the i^{th} task, including the size of requirement changes, FP_{max} is the sum of the function points when assigned the highest values, and N is the number of tasks performed at the time of measurement. Then, FCI is fuzzified and expressed in linguistic terms such that:

$$\widetilde{FCI} = [\text{Very Low, Low, Average, High, Very High}] \quad (10)$$

The fuzzy membership of the FCI is shown in terms of trapezoidal shapes, as in Figure 5.

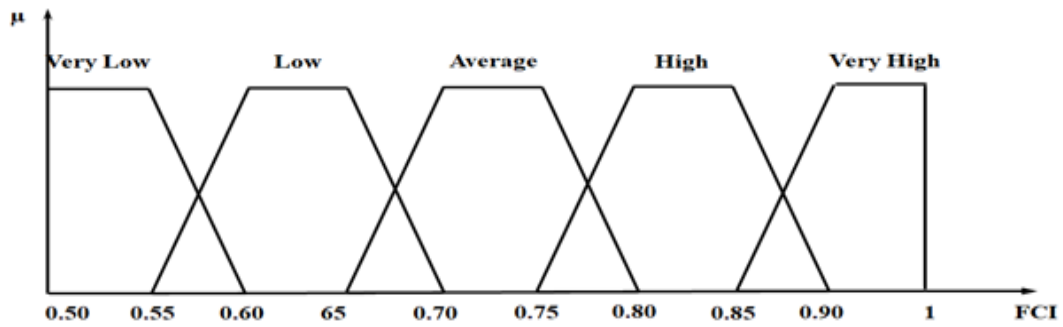


Figure 5. FCI fuzzy membership

Step 3: TCI calculation and fuzzification

We calculated the TCI in terms of the individual developer competency index (DCI), as described by Khalid and Yeoh [8]. The DCI, which quantifies the individual developer's competency, is calculated by giving weights to each competency. Then, the overall competency of each developer is calculated. The TCI resembles the centroid of the whole developers. TCI is fuzzified and expressed in linguistic terms as follows:

$$\widetilde{TCI} = [\text{Very Low, Low, Average, High, Very High}] \tag{11}$$

The fuzzy membership of TCI is shown in terms of trapezoidal shapes, as in Figure 6.

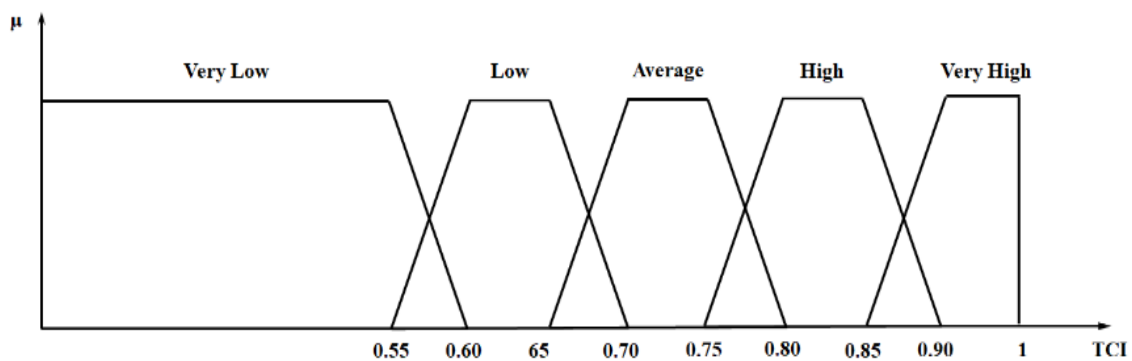


Figure 6. TCI fuzzy membership

Step 4: Fuzzy rules' definition

The fuzzy values of SRI are determined by establishing a relationship among SRI, FCI, and TCI according to the following fuzzy rule: **IF TCI IS [Value] AND FCI IS [Value] THEN SRI IS [VALUE]**. The set of fuzzy rules is shown in Table 2.

Table 2. Proposed fuzzy rules

ID	TCI	FCI	SRI
1	Very Low	Very Low	Average
2	Very Low	Low	Average
3	Very Low	Average	High
4	Very Low	High	Very High
5	Very Low	Very High	Very High
6	Low	Very Low	Average
7	Low	Low	Average
8	Low	Average	High
9	Low	High	High
10	Low	Very High	Very High
11	Average	Very Low	Low
12	Average	Low	Low

ID	TCI	FCI	SRI
14	Average	High	High
15	Average	Very High	High
16	High	Very Low	Very Low
17	High	Low	Low
18	High	Average	Low
19	High	High	Average
20	High	Very High	Average
21	V. High	Very Low	Very Low
22	V. High	Low	Very Low
23	V. High	Average	Low
24	V. High	High	Low
25	V. High	Very High	Average

Step 5: EAC Calculation

The total cost of the development work is given by:

$$\text{Total Cost} = \text{COD} + \text{COR} \quad (12)$$

Where COD is the cost of developed and COR is the cost of reworks. From (7), it is given that $\text{COR} = \text{SRI} \times \text{COD}$. Hence, total cost = $\text{COD} \times (1 + \text{SRI})$. The COD is equal to the EAC_T according to the traditional EVM model, while the total cost equals the EAC_P according to the proposed model. Therefore:

$$\text{EAC}_P = \text{EAC}_T \times (1 + \text{SRI}) \quad (13)$$

Where SRI is the crisp value calculated according to the fuzzy rules described in Table 2. Substituting the value of EAC_T in (5) and (6), respectively, yields the proposed two scenarios of EAC:

$$\text{EAC}_{P1} = \frac{\text{BAC} \times (1 + \text{SRI})}{\text{CPI}} \quad (14)$$

$$\text{EAC}_{P2} = \left(\text{AC} + \frac{\text{BAC} - \text{EV}}{\text{CPI} \times \text{SPI}} \right) \times (1 + \text{SRI}). \quad (15)$$

Where EAC_{P1} is influenced by the value of CPI, and EAC_{P2} is influenced by the values of both CPI and SPI.

3.3. Hypotheses definition

We defined two hypotheses.

The first hypothesis

- **Null hypothesis:** When CPI is the only influencing factor, there is no difference in the errors obtained when estimating the EAC using the traditional EVM and proposed EVM.
- **Alternative hypothesis:** When CPI is the only influencing factor, the error in estimating the EAC using the proposed EVM is less than the error obtained when using the traditional EVM.

Hence,

$$H_{10}: \left| \Delta_{T1} \right| - \left| \Delta_{P1} \right| \leq 0$$

$$H_{1A}: \left| \Delta_{T1} \right| - \left| \Delta_{P1} \right| > 0$$

Where Δ_{T1} and Δ_{P1} are the errors in estimation, using the traditional and proposed models, respectively.

The second hypothesis

- **Null hypothesis:** When both CPI and SPI are influencing factors, there is no difference in the errors obtained when estimating the EAC using the traditional EVM and proposed EVM.
- **Alternative hypothesis:** When both CPI and SPI are influencing factors, the error in estimating the EAC using the proposed EVM is less than the error obtained when using the traditional EVM.

Hence,

$$H_{20}: \left| \Delta_{T2} \right| - \left| \Delta_{P2} \right| \leq 0$$

$$H_{2A}: \left| \Delta_{T2} \right| - \left| \Delta_{P2} \right| > 0$$

Where Δ_{T2} and Δ_{P2} are the errors in estimation, using the traditional and proposed models, respectively. As described in Figure 7, we used the paired t-test to compare the errors of the two models [50].

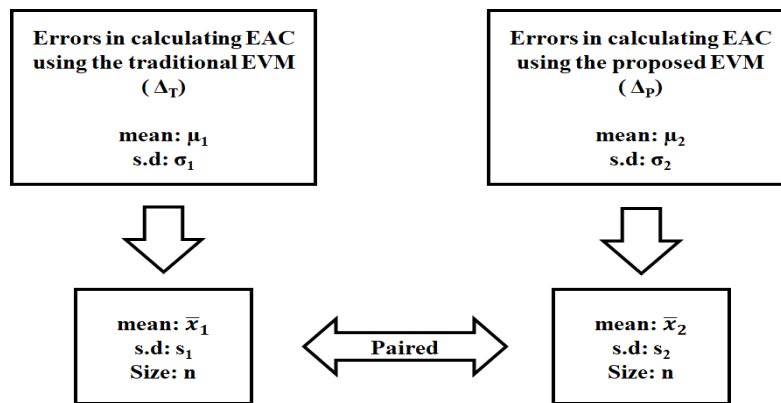


Figure 7. Paired t-test description

3.4. Experiment design

We designed two experiments. The first is to conduct the traditional EVM analysis. The output of each treatment consisted of the traditional EAC values; EAC_{T1} and EAC_{T2} . The second experiment conducted the fuzzy analysis as defined by the extended EVM model. The output of each treatment was an SRI value used to calculate the proposed EAC values; EAC_{P1} and EAC_{P2} . The traditional and proposed EAC values are used to define the elements of the first and second hypotheses, as shown in the following equations:

$$\Delta_{T1} = EAC_{T1} - ACAC \tag{16}$$

$$\Delta_{P1} = EAC_{P1} - ACAC \tag{17}$$

$$\Delta_{T2} = EAC_{T2} - ACAC \tag{18}$$

$$\Delta_{P2} = EAC_{P2} - ACAC \tag{19}$$

Where EAC_{T1} and EAC_{P1} are the estimates at completion using the traditional and proposed models, respectively, where the calculation is based on the influence of the CPI only, EAC_{T2} and EAC_{P2} are the estimates at completion using the traditional and proposed models, respectively, where the calculation is based on the influence of both the CPI and SPI. ACAC is the Actual Cost At Completion. The elements of the two experiments are shown in Table 3.

Table 3. Experiments' design elements

Element	Experiment # 1	Experiment # 2
Experimental Units	EVM checkpoints in a project	EVM checkpoints in a project
Dependent Variables	EAC_{T1} , EAC_{T1}	SRI
Independent Variables	%C, PV, AC, BAC	FCI, TCI
Number of Treatments	56	56

To facilitate the experiment execution, we developed a prototype using visual basic.net 2017 and fuzzy logic controller library called fuzzinator [51].

3.5. Data collection

This research adopted convenience sampling, where data was drawn from that available part of the population. We requested the research data from 55 sources. However, only three firms responded positively by sharing data belongs to five projects. The dataset facilitated 56 treatments for each experiment. We found the number of treatments enough to test the hypotheses since each experiment's sample size exceeded the minimum limit of 30 samples [52]. The experimental treatments provided by projects are shown in Table 4.

Table 4. Experimental treatments

Project ID	Experiment ₁ treatments	Experiment ₂ treatments
P01	11	11

P02	9	9
P03	10	10
P04	8	8
P05	18	18
Total	56	56

4. RESULTS AND DISCUSSION

The results of the first and second experiments are shown in Table 5 and Table 6, respectively.

Table 5. The first experiment results

Treatment ID	Project ID	PC	EV	CPI	SPI	EAC _{T1}	EAC _{T2}
1	P1	0.06	23.52	0.78	0.78	502.56	635.65
2	P1	0.23	90.16	0.86	0.86	455.81	513.11
3	P1	0.31	121.52	0.86	0.96	455.81	468.62
4	P1	0.45	176.4	0.9	0.9	435.56	461.17
5	P1	0.54	211.68	0.9	0.86	435.56	466.97
6	P1	0.63	246.96	0.88	0.84	445.45	478.21
7	P1	0.81	317.52	0.83	0.81	472.29	492.78
8	P1	0.9	352.8	0.81	0.9	483.95	491.77
9	P1	0.92	360.64	0.77	0.92	509.09	512.27
10	P1	0.96	376.32	0.76	0.96	515.79	519.49
11	P1	1	392	0.74	1	529.73	528
12	P2	0.16	41.92	0.93	0.93	281.72	299.46
13	P2	0.31	81.22	0.9	0.9	291.11	313.19
14	P2	0.46	120.52	0.93	0.94	281.72	291.84
15	P2	0.64	167.68	0.88	0.89	297.73	310.43
16	P2	0.74	193.88	0.89	1.03	294.38	291.31
17	P2	0.79	206.98	0.89	0.93	294.38	299.47
18	P2	0.86	225.32	0.89	0.93	294.38	297.32
19	P2	0.94	246.28	0.86	0.96	304.65	304.04
20	P2	1	262	0.88	1.02	297.73	297
21	P3	0.07	31.15	1.04	1.04	427.88	412.63
22	P3	0.11	48.95	0.96	0.96	463.54	480.74
23	P3	0.19	84.55	0.93	0.94	478.49	503.32
24	P3	0.29	129.05	0.93	0.95	478.49	496.61
25	P3	0.32	142.4	0.96	0.96	463.54	477.34
26	P3	0.44	195.8	0.94	0.94	473.4	491.03
27	P3	0.56	249.2	0.93	0.93	478.49	495.38
28	P3	0.74	329.3	0.95	0.95	468.42	475.2
29	P3	0.89	396.05	0.94	1.02	473.4	473.05
30	P3	0.94	418.3	0.99	1.08	449.49	446.97
31	P4	0.1	18	0.9	0.9	200	220
32	P4	0.3	54	0.9	0.9	200	215.56
33	P4	0.37	66.6	0.98	0.98	183.67	186.08
34	P4	0.49	88.2	0.9	0.9	200	211.33
35	P4	0.71	127.8	0.9	0.9	200	206.44
36	P4	0.8	144	0.89	0.89	202.25	207.45
37	P4	0.9	162	0.89	0.9	202.25	204.47
38	P4	1	180	0.87	1	206.9	206
39	P5	0.06	17.64	0.88	0.88	334.09	376.87
40	P5	0.12	35.28	0.88	0.88	334.09	374.09
41	P5	0.18	52.92	0.88	0.88	334.09	371.31
42	P5	0.2	58.8	0.77	0.77	381.82	472.69
43	P5	0.28	82.32	0.86	0.86	341.86	382.21
44	P5	0.34	99.96	0.86	0.86	341.86	378.36
45	P5	0.4	117.6	0.84	0.84	350	390
46	P5	0.45	132.3	0.83	0.83	354.22	394.72
47	P5	0.49	144.06	0.83	0.83	354.22	391.65
48	P5	0.56	164.64	0.85	0.85	345.88	373.04
49	P5	0.62	182.28	0.85	0.85	345.88	368.63
50	P5	0.67	196.98	0.84	0.84	350	371.5
51	P5	0.72	211.68	0.83	0.83	354.22	373.49
52	P5	0.79	232.26	0.85	0.85	345.88	359.45
53	P5	0.86	252.84	0.86	0.88	341.86	348.39
54	P5	0.92	270.48	0.86	0.92	341.86	343.73
55	P5	0.98	288.12	0.86	0.98	341.86	340.98
56	P5	1	294	0.82	1	358.54	357

Table 6. The second experiment results

Treatment ID	Project ID	TCI	FCI	SRI	EAC _{P1}	EAC _{P2}
1	P1	0.82	0.48	0.13	567.89	718.28

2	P1	0.82	0.57	0.13	515.07	579.81
3	P1	0.81	0.57	0.14	519.62	534.23
4	P1	0.81	0.59	0.15	500.89	530.35
5	P1	0.81	0.68	0.23	535.74	574.37
6	P1	0.81	0.61	0.16	516.72	554.72
7	P1	0.8	0.64	0.17	552.58	576.55
8	P1	0.8	0.64	0.17	566.22	575.37
9	P1	0.8	0.65	0.17	595.64	599.36
10	P1	0.8	0.67	0.21	624.11	628.58
11	P1	0.8	0.68	0.23	651.57	649.44
12	P2	0.78	0.53	0.17	329.61	350.37
13	P2	0.78	0.51	0.17	340.6	366.43
14	P2	0.78	0.55	0.17	329.61	341.45
15	P2	0.78	0.57	0.16	345.37	360.1
16	P2	0.78	0.57	0.16	341.48	337.92
17	P2	0.78	0.61	0.17	344.42	350.38
18	P2	0.78	0.62	0.17	344.42	347.86
19	P2	0.78	0.63	0.17	356.44	355.73
20	P2	0.78	0.63	0.17	348.34	347.49
21	P3	0.79	0.55	0.17	500.62	482.78
22	P3	0.78	0.56	0.16	537.71	557.66
23	P3	0.78	0.55	0.17	559.83	588.88
24	P3	0.77	0.57	0.16	555.05	576.07
25	P3	0.78	0.56	0.16	537.71	553.71
26	P3	0.78	0.58	0.16	549.14	569.59
27	P3	0.78	0.58	0.16	555.05	574.64
28	P3	0.78	0.58	0.16	543.37	551.23
29	P3	0.79	0.58	0.16	549.14	548.74
30	P3	0.79	0.58	0.16	521.41	518.49
31	P4	0.82	0.56	0.13	226	248.6
32	P4	0.82	0.54	0.13	226	243.58
33	P4	0.82	0.56	0.13	207.55	210.27
34	P4	0.82	0.59	0.14	228	240.92
35	P4	0.82	0.58	0.13	226	233.28
36	P4	0.82	0.59	0.14	230.57	236.49
37	P4	0.82	0.58	0.13	228.54	231.05
38	P4	0.82	0.58	0.13	233.8	232.78
39	P5	0.72	0.48	0.13	377.52	425.86
40	P5	0.72	0.49	0.14	380.86	426.46
41	P5	0.72	0.5	0.16	387.54	430.72
42	P5	0.71	0.5	0.16	442.91	548.32
43	P5	0.72	0.52	0.16	396.56	443.36
44	P5	0.72	0.59	0.23	420.49	465.38
45	P5	0.72	0.58	0.23	430.5	479.7
46	P5	0.72	0.56	0.19	421.52	469.72
47	P5	0.72	0.57	0.21	428.61	473.9
48	P5	0.72	0.56	0.19	411.6	443.92
49	P5	0.72	0.57	0.21	418.51	446.04
50	P5	0.72	0.58	0.23	430.5	456.94
51	P5	0.72	0.57	0.21	428.61	451.92
52	P5	0.72	0.56	0.19	411.6	427.75
53	P5	0.72	0.56	0.19	406.81	414.58
54	P5	0.72	0.56	0.19	406.81	409.04
55	P5	0.72	0.56	0.19	406.81	405.77
56	P5	0.72	0.54	0.16	415.91	414.12

To test the two hypotheses, we applied the procedures described by Xu et al. [50] to derive the results shown in Table 7.

Table 7. Hypotheses testing results

	1 st Hypothesis	2 nd Hypothesis
Mean of the differences, $\bar{\Delta}$	0.131	0.089
Standard deviation of the differences, $STDev(\Delta)$	0.180	0.528
Standard error of the differences, $SE(\Delta)$	0.024	0.071
T value	40.372	9.348
Degree of Freedom (n)	55	55
Significance Level, α	0.05	0.05
t-distribution of n degree of freedom, t_n	1.673	1.673

4.1. Results significance

Regarding Table 7, we observed that the T value is higher than the t_n value for the two tests. Therefore, we may conclude that the null hypothesis is rejected at a confidence level of 95% for the two tests.

Hence, we may conclude that $|\Delta_{P1}| < |\Delta_{T1}|$ for the first hypothesis and $|\Delta_{P2}| < |\Delta_{T2}|$ for the second hypothesis. Moreover, when we investigated the accuracy of the traditional EVM model, assuming no software reworks, we found that the mean error in estimating EAC is less than 6%, which is quite tolerable. Upon considering the reworks, we found that the mean errors approaching values between 14% and 18% that can be seen in Table 8. These results strongly support Efe and Demiros [15] that when ignoring the reworks, the overall picture was acceptable. Therefore, we can conclude that the inaccuracy in estimating the software projects' EAC is due to the incapability of the traditional EVM to incorporate the cost of software rework.

Table 8. Accuracy of the traditional and proposed models

	Mean Error	
	Considering the CPI	Considering the CPI and SPI
Traditional EVM	17.47 %	14.08 %
Proposed EVM	04.48 %	05.18 %
Error Reduction	12.99%	8.90%

The results of the two experiments show that the proposed EVM model is more precise than the traditional one by around 13% in estimating the EAC when considering the impact of the CPI. On the other hand, when considering the impact of both the CPI and SPI, the proposed model is more precise than the traditional one by around 9%.

4.2. Scientific and practical implications

This research brings the recommendations made by many scholars such as Efe and Demiros [15] and Bhardwaj and Rana [53] into reality. It customizes the traditional EVM, to suit the very needs of software projects. This research's main scientific achievement is identifying the main reason behind the inaccuracy of the traditional EVM, i.e., its incapability to estimate the cost of software rework. Moreover, we establish a new way of thinking by investigating the relationship between human factors, software complexity, and the cost of rework. We believe this study can assist the software project managers in many aspects, such as forecasting the software rework and overall cost more accurately, reducing the subjectivity in progress measurement and software sizing by adopting the FPA, and considering and quantifying human factors.

4.3. Threats to validity

The threats to the research validity may include threats to internal validity, construct validity, and external validity. The analysis of these threats follows the guidelines suggested by Kitchenham et al. [54] and Perry et al. [55]. Internal validity. The study suggests that software rework depends mainly on team competency and software functional complexity. Other factors, such as the competency of the testing team, are not studied. For example, if the testers are not competent, the product might be signed off with uncovered defects resulting in a high maintenance cost. Moreover, the TCI is calculated as an average of the individual developers' DCIs while ignoring the team's collective attributes. Therefore, there is a need to investigate the influence of collective attributes on software project performance. Moreover, the collected data did not include agile projects, which may have more focus on the human side by embodying values such as commitment, focus, and respect [56].

Construct validity. The developers' experience may influence the accuracy of the FPA measurements [57]. Also, the subjectivity, skills, and experiences of individuals who filled the developers' assessment forms may affect the accuracy of both the DCI and TCI calculations. External validity. Although we successfully evaluate the proposed EVM model, we suggest that this model is neither comprehensive nor generic due to the following threats:

- a) Due to the weak response of the software firms, the dataset was drawn from only three firms. This situation may increase the possibility of inadvertent sampling bias.
- b) The model evaluation did not include projects with large sizes and large team members. These projects may have more complexity-related factors not investigated by this study.
- c) We did not test projects adopting methodologies other than the waterfall projects.

5. CONCLUSION

This research aims to enhance the software cost control by customizing the traditional EVM to satisfy the needs of the software projects. Accordingly, we design a modified EVM to reduce the identified

limitations. The model has been evaluated using data drawn from five actual projects. The evaluation indicates that the proposed EVM model is more accurate than the traditional model. However, our main finding is that the EVM cost forecast inaccuracy is due to its inability to incorporate the cost of software rework. The research contributions include the introduction of the software rework index (SRI), enabling the EVM model to forecast the cost of rework during the project lifecycle. The direction of the future work may include evaluating the proposed model using data drawn from projects with larger sizes and higher functional complexity. Future studies should also investigate the impact of different software development methodologies such as spiral and Scrum. Moreover, we plan to study the team's collective characteristics, as the overall team competency is not a simple averaging of the individual attributes

REFERENCES

- [1] R. N. Charette, "Why software fails [software failure]," in *IEEE Spectrum*, vol. 42, no. 9, pp. 42-49, Sept. 2005. doi: 10.1109/MSPEC.2005.1502528.
- [2] The Standish Group, "The chaos report," standishgroup.com, 2015. Available: https://www.standishgroup.com/sample_research_files/CHAOSReport2015-Final.pdf. [Accessed: November 30, 2017]
- [3] The Standish Group, "Project resolution benchmark report," standishgroup.com, 2018. Available: https://www.standishgroup.com/sample_research_files/DemoPRBR.pdf. [Accessed December 7, 2018].
- [4] R. Colomo-Palacios, C. Casado-Lumbreras, P. Soto-Acosta, F.J. García-PeñaAlvo, E. Tovar-Caro, "Competence gaps in software personnel: A multi-organizational study," *Computers in Human Behavior*, vol. 29, no. 2, pp. 456-461, 2013.
- [5] Blueprint, "The rework tax: reducing software development rework by improving requirements," Blueprint Software Systems Inc., 2015. Available: <http://www.blueprintsys.com/content/the-rework-tax-reducing-software-development-rework-by-improving-requirements/>. [Accessed: July 2, 2016].
- [6] I. Sommerville, "Software testing" in software engineering, 9th ed. Boston: Edison Wesley, 2010.
- [7] T.A. Khalid, ET. Yeoh, "Controlling software cost using fuzzy quality based EVM," In *conference on computing, control, networking, electronics, and embedded systems engineering, -ICCCCEE. IEEE*, pp 275-280, 2015.
- [8] T.A. Khalid, ET. Yeoh, "Towards incorporating human factors in the software project cost control models–Preliminary study," *The Journal of Modern Project Management*, vol. 6, no. 2, 2018.
- [9] P. Waychal, L.F. Capretz, "Need for a soft dimension," *3rd International conference on software engineering*, Geneva, Switzerland. doi: 10.5121/csit.2017.70414, 2017.
- [10] P. Efe, O. Demirors, "A change management model and its application in software development projects," *Computer Standards & Interfaces*, 2019.
- [11] Project Management Institute, "Software Extension to the PMBOK Guide, 5th Edition," *Project Management Institute, IEEE Computer Society*. Newtown Square, PA, 2013.
- [12] S. Sheard, C.B Weinstock, M.D. Konrad, D. Firesmith, "FAA research project on system complexity effects on aircraft safety: identifying the impact of complexity on safety," Carnegie-Mellon University, Software Engineering Institute. Pittsburgh, PA, 2016.
- [13] F. Feiler, "An incremental life-cycle assurance strategy for critical system certification," Carnegie-Mellon University, Software Engineering Institute. Pittsburgh, PA, 2014
- [14] Micro Focus, "Successful projects start with high quality requirements," *Micro Focus International*, Rockville, Maryland, 2016. Available: <https://www.microfocus.com/media/white-paper/WP-Successful-projects-start-with.pdf>. [Accessed: December 10, 2016].
- [15] P. Efe, P. Demirors, "Applying EVM in a software company: benefits and difficulties," In *39th Euromicro Conference on Software Engineering and Advanced Applications. IEEE*, pp. 333-340, 2013.
- [16] G. A. Cass, M. Stanley, Sutton Jr., L. J. Osterweil, "Formalizing rework in software processes," In *Software Process Technology. Springer*, Berlin, pp. 16-31, 2013.
- [17] Nguyen-Duc, "The impact of software complexity on cost and quality-a comparative analysis between Open source and proprietary software," *International Journal on Software Engineering and Application*, vol. 8, no. 2, pp. 17-31, 2017.
- [18] Project Management Institute, "A guide to the project management body of knowledge," sixth edition. PMI. Newton Square, PA, 2017
- [19] Naval Air Forces Organization, NAVAIR, "Using software metrics and measurements for earned value toolkit," 2004. Available: <https://acc.dau.mil/adl/en-US/19591/file/1043/NAVAIR%20Software%20EVM%20Toolkit%20%20Dec%202004.pdf>. [Accessed: September 27, 2013].
- [20] P. Bourque, R.E. Fairley (eds.), "Guide to the software engineering body of knowledge," Version 3.0. IEEE Computer Society, www.swebok.org, 2014.
- [21] ISO, "ISO 21508:2018 Earned value management in project and program management," Edition 1. Switzerland, 2018.
- [22] J. Jurison, "Software project management: the manager's view," *Communications of the association for information Systems*, vol. 2, no. 1, pp. 17, 1999.
- [23] D. Zowghi, M. Haghghi, B. Zohouri, "Cost and schedule control approach in fuzzy environment," *International Journal of Research and Reviews in Information Sciences (IJRRIS)*, vol. 1, no. 2, pp. 67-73, 2011.

- [24] L. M. Naeni S. Shadrokh, A. Salehipour, "A fuzzy approach for the earned value management," *International Journal of Project Management*, vol. 32, pp. 709-716, 2014.
- [25] P. Solomon P, "Basing based earned value on technical performance," *The Journal of Defense Software Engineering*, vol. 1, no. 2, pp. 25-28, 2013.
- [26] M. Yilmaz, R. V. O'Connor, R. Colomo-Palacios, P. Clarke P, "An examination of personality traits and how they impact on software development teams," *Information and Software Technology*, vol. 86, pp. 101-22, 2017.
- [27] K. Langsari and R. Sarno, "Optimizing effort and time parameters of COCOMO II estimation using fuzzy multi-objective PSO," *2017 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, Yogyakarta, pp. 1-6, 2017. doi: 10.1109/EECSI.2017.8239157.
- [28] M. Esteki, T. Javdani Gandomani, & H. Khosravi Farsani, "A risk management framework for distributed scrum using PRINCE2 methodology," *Bulletin of Electrical Engineering And Informatics*, vol. 9 no. 3, pp. 1299-1310, 2020. doi:10.11591/eei.v9i3.1905.
- [29] D. De Souza, A. R. C. Rocha, "A proposal for the improvement the predictability of project cost using EVM and historical data of cost," In *35th international conference of software engineering-ICSE, ACM SRC, San Francisco, International Journal of Software Engineering and Knowledge Engineering*, vol. 25, no. 1, pp. 27-50, 2015. Available: <https://doi.org/10.1142/S0218194015400021>. [Accessed: October 30, 2015].
- [30] F. Acebes, J. Pajares, J. M. Galán, A. López-Paredes, "A new approach for project control under uncertainty. Going back to the basics," *International Journal of Project Management*, vol. 32, no. 3, pp. 423-434, 2014.
- [31] D. De Souza, A. R. C. Rocha, D. Cristina, B. A. Constantino, "A proposal for the improvement of project's cost predictability using earned value management and quality data-an empirical study," In *European conference on software process improvement. Springer*, Berlin, pp. 170-181, 2014.
- [32] X. Ma, B. Yang, "Optimization study of Earned Value Method in construction project management," In *2012 International conference on information management, innovation management and industrial engineering (ICIII). IEEE*, vol. 2, pp 201-204, 2012.
- [33] J. Xu, H. Zhang, Li F, "Project integrated management based on quality earned value," In *2nd international conference on information science and engineering (ICISE). IEEE*, pp 432-435, 2010.
- [34] H. K. Raju, Y. T Krishnegowda, "Software sizing and productivity with Function Points," *Lecture Notes on Software Engineering*, vol. 1, no. 2, pp. 204, 2013.
- [35] G. P. Jiang, L. Xie, "Associated Control Research for Software Project Schedule and Budget Based on Function Point Method," *DEStech Transactions on Computer Science and Engineering, (CSSE)*, 2018.
- [36] H. Huang, J. Zheng, "Quality earned value analysis based on IFPUG method in software project," In *Proceedings of 2018 International Conference on Big Data Technologies*, pp. 101-108. ACM, 2018.
- [37] T. Sulaiman, B. Barton, T. Blackburn, "Agile EVM-earned value management in Scrum Projects," *Agile Conference, IEEE*, pp. 10-16, 2006.
- [38] J. Rusk, "Earned value for agile development," *Software Tech News*, vol. 12. no. 1, pp. 20-27, 2009.
- [39] P. K. Roy, P. Goutam, "Agile with EVM," *Int. Journal of Core Engineering & Management*, vol. 1, no. 8, 2014.
- [40] C. J. Torrecilla-Salinas, J. Sedeño, M. J. Escalona, M. Mejías, "Estimating, planning and managing Agile Web development projects under a value-based perspective," *Information and Software Technology*, vol. 61, pp. 124-144, 2015.
- [41] K. Pracharnasniyom, S. Utsugi, Y. Koizumi, S. Hirose, H. Aso, T. Konosu, "Human resource management in small-scale projects," *Journal of Business Administration and Languages*, 2015. Available: <http://journal.tni.ac.th/upload/files/pdf/Human%20Resource%20Management%20in%20Smallscale%20Project.pdf>. [Accessed: October 2, 2017].
- [42] R. V. Vargas, "Using earned value management indexes as a team development factor and a compensation tool," In *Project management institute global congress EMEA*. Prague, 2004.
- [43] G. Broza, "The Human Side of Agile: How to help your team deliver," 3P Vantage Media, ISBN-13: 978-0988001626, 2012.
- [44] T. DeMarco, T. R. Lister, "Peopleware: productive projects and teams," Addison-Wesley. Upper Saddle River, NJ, 2015.
- [45] R. AlQaisi, E. Gray, B. Steves, "Software systems engineering: A journey to contemporary agile and beyond, do people matter?," *BCS*, pp. 159-173, 2017. Available: <http://ssudl.solent.ac.uk/id/eprint/3580> [Accessed: January 15, 2018].
- [46] S. Wagner, M. Ruhe, "A systematic review of productivity factors in software development," In *Proceedings of the 2nd international workshop on software productivity analysis and cost estimation. State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences*, arXiv:1801.06475v1, 2018.
- [47] F. Zainal Abidin, M. Darmawan, M. Osman, S. Anwar, S. Kasim, A. Yunianta, T. Sutikno, "Adaboost-multilayer perceptron to predict the student's performance in software engineering," *Bulletin of Electrical Engineering and Informatics*, vol. 8, no. 4, pp. 1556-1562, 2019.
- [48] S. N. Sivanandam, S. Sumathi, S. N. Deepa, "Introduction to fuzzy logic using MATLAB," *Springer*. Berlin, 2007.
- [49] International Function Points Users Groups (IFPUG), "Function Point Counting Practices Manual," Release 4.3.1. Netherland, 2010.
- [50] M. Xu, D. Fralick, J. Z. Zheng, B. Wang, X. M. Tu, C. Feng, "The differences and similarities between two-sample t-test and paired t-test," *Shanghai Archives of Psychiatry*, vol. 29, no. 3, pp. 184-188, 2017.
- [51] H. Omran, "Fuzzinator: A fuzzy logic controller," 2011. Available: <https://www.codeproject.com/KB/recipes/Fuzzinator/FuzzyLogicController.zip>. Accessed: 1 June 2016

- [52] Minitab, Inc. (2010). Statistical inference and t-tests. <http://www.minitab.com/uploadedFiles/Documents/sample-materials/TrainingTTest16EN.pdf>. [Accessed: November 20, 2018].
- [53] M. Bhardwaj, A. Rana, "Impact of Size and Productivity on Testing and Rework Efforts for Web-based Development Projects," In *ACM SIGSOFT software engineering notes*, vol. 40, no. 2, pp. 1-4, 2015.
- [54] B. A. Kitchenham, S. L. Pfleeger, L. M. Pickard, P. W. Jones, D. C. Hoaglin, K. El Emam, J. Rosenberg, "Preliminary guidelines for empirical research in software engineering," *IEEE Transactions on software engineering*, vol. 28, no. 8, pp. 721-734, 2002.
- [55] D. E. Perry, A. A. Porter, L. G. Votta, "Empirical studies of software engineering: a roadmap," In *Proceedings of the conference on the future of software engineering. ACM*, pp 345-355, 2000.
- [56] K. Schwaber, J. Sutherland, "The scrum guide. The definitive guide to scrum: The rules of the game," 2017. Available: <https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf#zoom=100>. [Accessed: May 10, 2018].
- [57] T.S.O.I Ho-Leung, "To evaluate the function point analysis: a case study," *International Journal of the Computer, the Internet and Management*, vol. 13, no. 1, pp. 31-40, 2005.

BIOGRAPHIES OF AUTHORS



Tarig Ahmed Khalid is a research scholar at Multimedia University, Malaysia. He obtained his B.Sc. and M.Sc. in electrical engineering from University of Khartoum, Sudan. He is a certified project management professional (PMP). His research interests include software project management, requirements engineering, and fuzzy systems. He is a senior member of the IEEE and member of the PMI.



Eng-Thiam Yeoh is a Senior Lecturer in Faculty of Computing & Informatics, Multimedia University, Malaysia. He obtained his PhD from Multimedia University in 2009, after obtaining his M.Phil. degree from University of Cambridge, England (1991) and his first degree in Computer Science from National University of Malaysia (1989). His research interests include e-learning, software engineering, multimedia education, fuzzy systems, natural language processing and big data. He is a member of the IEEE.