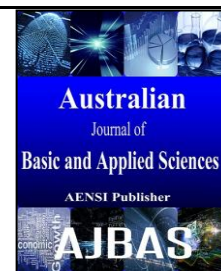




ISSN:1991-8178

Australian Journal of Basic and Applied Sciences

Journal home page: www.ajbasweb.com

Simplified Software Effort Estimation Using Fuzzy Set Theory

¹Ms. B. Arthi and ²Dr. A. Grace Selvarani¹Department of Information Technology, Easwari Engineering College, Chennai, TamilNadu, India²Department of Computer Science and Engineering (PG), Sri Ramakrishna Engineering College, Coimbatore, TamilNadu, India

ARTICLE INFO

Article history:

Received 12 March 2015

Accepted 28 June 2015

Available online 22 July 2015

Keywords:

Software Effort Estimation, Use Case Points, Fuzzy Set Theory, Fuzzy Size, Categorization.

ABSTRACT

Software effort and cost estimation is a very substantial activity that comprises very ambiguous elements. In the framework of object oriented software, uncertain approach and metrics were prolonged to assist project managers in this action. The metric Use Case Points (UCP) is an instance of metric that can be used to contemplate practical condition of the UseCase (UC) in the initial state of the progression. However, the metric UCP proposes some circumspection mainly associated to the relevance of the UC. To eradicate these constraints, this paper proposes a metric called Fuzzy Size of Use Case Points (FSUCP) which contemplates the logic of the Fuzzy Set Theory to establish progressive categorization that better deal with ambiguity. With this we also incorporate notion to elucidate the software estimation process by reducing the number of adjustment factors on the accuracy of the FSUCP which has same adjustment factors as that of UCP. The results of the factor analysis indicated that the number of adjustment factors could be reduced from 21 to 6 (2 environmental factors and 4 technical complexity factors). Results from an empirical evaluation show the relevance and some benefit of the recommended metric and confirmed that software effort estimation process has been more uncomplicated correlated to already current metrics.

© 2015 AENSI Publisher All rights reserved.

To Cite This Article: Ms. B. Arthi and Dr. A. Grace Selvarani, Simplified Software Effort Estimation Using Fuzzy Set Theory. *Aust. J. Basic & Appl. Sci.*, 9(23): 347-353, 2015

INTRODUCTION

Software effort estimation is one of the main specifications of fruitful project management. If an improbable inference about the progress cost is made, the project is in risk. Both underestimated and overestimated effort is dangerous. Underestimation leads to a condition where a project's obligations cannot be achieved because of a lack of time and/or repository. Overestimation can outcome in the elimination of a project recommendation, which otherwise would be acknowledged and would set up advanced favourable circumstances for the organization. Regrettably, effort estimation at the initial phase of software development is a dispute. Initially, very limited is recognized about the project. Secondly, there is a warning that the project will not be acknowledged for additional progress, so little assets can be used on effort estimation. Thus, there is a concession between the stage of estimation error and the resources entrust to the assessment activities (typically, the smaller the estimation error the bigger the estimation cost correlate with obtaining insight about the project within reach).

In this context two kinds of research could be useful:

- simplifying effort estimation methods without compromising their accuracy;
- Making effort estimation more accurate without increasing the time and money spent on effort estimation.

For precise estimation we include Fuzzy Size of Use Case Points (FSUCP), contemplates logics of the Fuzzy Set Theory to scale down the human domination on the categorization of the UC factors. This hypothesis was successfully enforced in many fields of acquaintance and recently it has been applied to create an enlargement of the FPA (Celar, S., 2013). Hence, the application of these logics in metric based on UCs is also attainable and seductive to avoid brusque and non continuous classifications. To disentangle the effort estimation process we incorporate the notion of reducing number of adjustment factors. In order to acquire UCP for the entity one has to start with the assessment of the complexity of actors and use cases; and then accustom it with two category of factors distinguish the progressive ambiance and the technical complexity of the entity beneath progress. The UCP method incorporates 21 adjustment factors, which deals with the technical complexity of the advanced system (13 technical complexity factors), and the

Corresponding Author: Ms.B.Arthi, Assistant Professor, Department of Information Technology, Easwari Engineering College, Chennai, TamilNadu, India.
E-mail:arthib98@gmail.com

ambiance in which it is progressed (8 environmental factors). The repercussions of technical complexity factors (TCF) are evaluate by empowering a value from 0 to 5 to each of them (the bigger the number is, the greater the range a given factor appears with). The forth coming paper is standardized as follows. Section 2 instants related activity. Sections 3 and 4 establish the metric FSUCP respectively. Section 5 shows the results of the employment of the imported metric. Section 6 concludes the paper.

Related work:

This section depicts the instants works that are the paltry for FSUCP. It contains a characterization of the metric UCP and characterizes works that incorporate the logics of the Fuzzy Set Theory in the framework of software metrics. It also illustrate the devaluation of number adjustment factors for elucidate the effort estimation activity.

Use Case Points:

The UCP (Use Case Points) (Ziauddin1, 2012) is an reworking from FP and MKII FP (Celar, 2013). Because of this, it includes the most FPA typical features, such as the utilization of adjustment factors to the Unadjusted UCP (UUCP). The UUCP is acquired by aggregating the Unadjusted Use Case Weights (UUCW) and the Unadjusted Actors Weights (UAW). The UUCW is attained by accustoming a weight to each UC of the system. The weight is given based on number of undertaking or entities of the UC. The aggregate of the weights of all UCs is the UUCW. Similarly the UAW is given by the aggregate of the weights of all actors of the system. Each actor is categorized based on its intricacy. After acquiring the UUCP, the next stride is the utilization of the adjustment factors, but our examination will target on the way UCP categorize the UCs. UCs in UCP could only be categorizing into one of the three categories: simple, average and complex, no matter how many portion it has or how big it is. For example, consider three UCs: UC1, UC2 and UC3 with respectively 70, 2 and 8 transactions. According to the UCP categorization tables, UC2 is considered simple and UC1 and UC3 are complex. However, UC1 is much more complex than UC3. This kind of situation can be quite common when applying the UCP metric. Many times, a UC have higher priority. This is due to its functionality, delivery deadline, simplicity, etc. The managers need to evaluate a UC or a set of UCs separately.

Fuzzy Theory and Metrics:

The Fuzzy Set Theory extends the standard set theory by granting the commencement of ambiguity

over the sets and a continuous modify through the logics of association. A fuzzy set is distinguished by a membership function ('Performance Evaluation of Software Effort Estimation using Fuzzy Analogy based on Complexity', 2012), which exemplifies the membership grade of a distinctive in a association. In this way, the individual can exist to more than one group in a given degree. The Fuzzy theory has been attainable applied in many segments such as engineering, psychology, artificial intelligence, medicine, sociology (Azzeh, 2011). Each type in a classification table resembles to a grammatical phrase that is not able to symbolize well all the original values of this classification. They also presents some benefit of the expansion introduces; the miniature becomes more common and the changeover from one grammatical term to another neighbour one becomes continuous, unlike the actual brusque way.

Fuzzy size of use case points (fsucp):

The metric UCP instantiates new component for calibrating the operation of the UCs. However, it uses a individual categorization of the operation of complexity. The use of the classification tables does not allow continuous change from one intricacy category to another. To grant such continuous change, we expand the metric UCP by adopting the FFPA (and establishes a metric named Fuzzy Size of Use Case Points(FSUCP). First we have to compute the values of SUCP (Size of Use Case Points), after this, the following steps are required: Fuzzyfication and Defuzzyfication of the grammatical terms. In FSUCP, firstly we are required to compute the value of SUCP. The following steps show how to compute SUCP.

Actors Categorization:

Each actor has its complexity (CA) resolved based on the information given t or collected from the UC being categorized (Table 1). The complete complexity of actors in the UC (ACT) is computed by Equation 1.

$$ACT = \sum_{i=1}^n CA_i \quad (1)$$

n- Number of actors in UC

Precondition Categorization:

Each arrangement of the UC has the complexity (PrCC) analysed based on the number of reasonable interpretation certified by the condition (Table 2). The total complexity of the preconditions (PrCCT) is given by Equation 2.

$$PrCCT = \sum_{i=1}^n PrCC_i \quad (2)$$

n - Number of preconditions in the UC.

Table 1: SUCP Actor categorization.

Complexity	Data	SUCP
Simple	<=5	2
Average	6 to 10	4
Complex	>10	6

Table 2: SUCP Precondition categorization.

Complexity	Tested Expression	SUCP
Simplicity	1 logical expression	1
Average	2 or 3 logical expression	2
Complex	3 logical expression	3

Table 3: SUCP Scenario Categorization

Complexity	Entities+Steps	SUCP
Very Simple	<=5	4
Simple	6 to 10	6
Average	11 to 15	8
Complex	16 to 20	12
Very Complex	>20	16

Table 4: SUCP exception Categorization.

Complexity	Tested Expression	SUCP
Simple	1 logical expression	1
Average	2 or 3 logical expression	2
Complex	>3 logical expression	3

Table 5: SUCP Post condition Categorization

Complexity	Entities	SUCP
Simple	<=3	1
Average	4 to 6	2
Complex	>6	3

Main Scenario Categorization:

The main scenario must be categorized based on its number of elements and to the number of fundamental steps needed to the scheme consequence. The complexity of the scenario (PCP) is given by the quantification of both values (number of elements + number of steps), according to Table 3.

Surrogate Scenario Categorization:

All the substitute scenarios are categorized alike to the main scenario. Each surrogate scenario gets a number of points (NPCA) according to Table 3. The complete complexity of the surrogate scenarios (TPCA) is given by Equation 3.

$$TPCA = \sum_{i=1}^n NPCA_i \quad (3)$$

n - Number of alternative scenario in the UC.

Exception Categorization:

Each exception present in the UC must also be evaluated based on its complexity (CE), resolved by the number of logical expressions approved to deduce the exception happening. The total points added by exceptions (PET) are resolved by Equation 4. Table 4 helps in this categorization.

$$PET = \sum_{i=1}^n CE_i \quad (4)$$

n - Number exceptions in the UC.

Postcondition Classification:

The complexity of the post conditions (PoCC) is resolved according to the number of associated elements (Table 5). The total complexity of post

conditions (PoCCT) is given by Equation 5.

$$PoCCT = \sum_{i=1}^n POCC_i \quad (5)$$

n - Number of post conditions in the UC.

Calculation of Unadjusted Value:

The Unadjusted Use-case Size Point (USUCP) is given by the sum of the complexity values of all sections of the UC (Equation 7).

$$USUCP = TPCA + PrCCT + PCP + TPCA + PET + PoCCT \quad (6)$$

Application of the Adjustment Factor:

The way SUCP calculates the adjustment factor is derived from FP [1] and UCP [2, 10]. Here we need to lessen the number adjustment factors from 21 to 6 to simplify the effort estimation process.

Reducing the number of adjustment factors:**Technical Adjustment Factors (FAT):**

The technical factors resemble the domination that some technical attribute (contemporary in the software being progressed and instinctive to all UCs of the system) could have upon the software. Each adjustment factor in Table 6 domination of technical complexity factors (FAT) by assigning a value from 0 to 5 to each of them (the bigger the number is, the greater the range a given factor appears with). This value is multiplied by a weight of a factor and summed, see the following equation: The technical adjustment factor is calculated by Equation 7.

$$FAT = 0.65 + (0.01 * \sum_{i=1}^{13} TFWeight_i * value_i) \quad (7)$$

Where

- TFWeight_i is the weight of the *i*th technical complexity factor (see Table 6);

- value_i is the conclude degree of domination of the *i*th technical complexity factor on the project (value between 0 and 5).

Table 6: Technical Factors adapted from (Dian Pratiwi, 2013)

Factor	Requirement	Influence	Weight
F1	Data communication	I1	2
F2	Distributed processing	I2	1
F3	Performance	I3	1
F4	Equipment utilization	I4	1
F5	Transaction Capacity	I5	1
F6	On-line input of data	I6	0.5
F7	User efficiency	I7	0.5
F8	On-line update	I8	2
F9	Code reuse	I9	1
F10	Complex processing	I10	1
F11	Easiness of deploy	I11	1
F12	Easiness operation	I12	1
F13	Many places	I13	1

Table 7: Environmental factor adapted from [6]

Factors	Description	Influence	Weight
E1	Formal development process existence	I1	1.5
E2	Experience with the application being Developed	I2	0.5
E3	Experience of the team with the used Technologies	I3	1
E4	Presence on an experienced analyst	I4	0.5
E5	Stable requirements	I5	1
E6	Part time workers	I6	2
E7	Motivation	I7	-1
E8	Difficulty in programming language	I8	-1

Environment Adjustment Factor (FAE):

The environmental factors resemble some attribute contemporary at the progress of environment that could dominate the software cost. Each factor from Table 7 receives a value and the Environmental Adjustment Factor (FAE) is given by Equation 8.

$$FAE = 1.4 + (-0.03 * \sum_{i=1}^8 EFweight_i * value_i) \quad (8)$$

Where,

- EFWeight_i is the weight of the *i*th environmental factor (see Table 7);
- value_i is the conclude the degree of dominance of the *i*th environmental factor on the project (value between 0 and 5).

Calculating Use Case Points:

By adding UAW to UUCW, according to Eq. (5), one obtains Unadjusted Use Case Points (UUCP).

$$UUCP = UAW + UUCW \quad (9)$$

To obtain Use Case Points (UCP) one has to multiply UUCP by FAT and FAE, see the following equation:

$$UCP = UUCP * FAT * FAE \quad (10)$$

Concluding the calculation:

The final value for a UC is given by Equation 9

$$SUCP = UUCP * (FAT - FAE) \quad (11)$$

Productivity factor and effort estimation:

To acquire effort estimation in man-hours one has to multiply UCP by the productivity factor (FP). The delinquency value for FP recommended by Karner is 20 h per UCP. Schneider and Winters proposed a method for determining the initial value of FP. Based on their involvement, they recommend to count the number of environmental factors F1–F6 which domination is concluded to be less than 3 and factors F7–F8 which dominance is concluded to be greater than 3. If the estimated total is equal to 2 or less, the delinquency value of 20 h/UCP should be used. If the total is between 3 and 4, they recommended using FP equal to 28 h/UCP. If the computed number is greater than 4, the value of 36 h/UCP should be used. (However, in this case the project is respect as a largely dangerous one.)

Calculating ucp with historical data:

Using default values for FP is a necessity if an organization does not have historical data concerning productivity. However, if historical data is available, it is reasonable to use such data to determine FP for the project being estimated.

After the completion of a single project, a post-productivity factor (PostFP) might be calculated as presented in the following equation:

$$PostFP = \text{Actual effort} / SUCP \quad (12)$$

Fuzzy size of use case points (fsucp):

The metric FSUCP presents new elements for measuring the functionality of the UCs. However, it also uses a discrete classification of the functionalities complexity, like UCP and FPA. The use of the classification tables does not allow a gradual change from one complexity category to another. To allow such gradual change, we extended the metric using FFPA steps (Jonathan Lee, 2011), the functionalities present in the UC receive a continuous classification. After this, the following steps are necessary: Fuzzyfication and Defuzzyfication of the linguistic terms.

Fuzzyfication of Linguistic Terms:

The classification tables are transformed into a continuous classification, this process is called fuzzyfication (a more formal definition to fuzzyfication could be found in (Azzeh, 2011)). This can be made through the generation of a trapezoidal fuzzy number to each complexity category found on the classification tables. Then, each classification table

for a UC (actors, preconditions, exceptions, etc) is represented by a graph, to generate the graph, that is the trapezoidal number, the following variables are calculated, for each category in the classification tables ($1 \leq i \leq n$, and n is the number of linguistic terms in the classification table being analysed).

Shortly

- p_i = lower value of the linguistic term T_i in the classification Table
- $n_i = (p_i + p_{i+1})/2$
- $a_i = n_{i-1}$
- $b_i = p_{i+1}$

Table 8 shows the values of the above variables for the FSUCP classification tables. For example, the table for Actors Classification has three linguistic terms: Simple (T1), Average (T2) and Complex (T3). The table for Scenarios Classification has five terms. Some adjustment is necessary when transforming the first or last linguistic terms. The graphs obtained for each FSUCP classification table are present in Figure 1.

Table	p1	n1	a1	b1	p2	n2	a2	b2	p3	n3	a3
1	1	3.6		7	7	8.6	3.6	12	12		8.6
2	1	1.6		3	3	4	1.6	5	5		4
3	1	3.6		7	7	8.6	3.6	12	12	13.6	8.6
4	1	1.6		3	3	4	1.6	5	5		4
5	1	2.6		5	5	5.6	2.6	8	8		5.6

Table 8: Values for the Fuzzyfication of the Terms.

Table	b3	p4	n4	a4	b4	p5	n5	a4	b4
1									
2									
3	17	17	19	14	22	22		19	
4									
5									

Defuzzyfication of the Linguistic Terms:

The process of translating back fuzzy numbers into single “real world” values is named Defuzzyfication (a formal definition to defuzzyfication can be found in (Azzeh, 2011)). The defuzzyfication was made through the utilization of two simple rules (Jonathan Lee, 2011), explained below. After this, the UFSUCP (Unadjusted FSUCP) can be computed. The process involves the computation of the membership function (x), that resembles the membership degree of the element x (a number) in the set in question, in this case, the membership grade of the value in a category of complexity. Each rule is applied in a special situation, the first one is applied when the acquired number belongs to only one fuzzy number and the last rule is applied when the value is between two fuzzy numbers (in a conversion region).

1. If the number to be categorized (logical expressions, entities, entities plus steps) is between the values of p_i and n_i of the contributor Fuzzy number, the value will be the same of the category to which this number belongs. This occurs because it is in the upper base of the trapezoidal number, in this

case the value of the membership function (x) is equal to one (1), generating the same value that SUCP generates.

2. When the number to be classified (logical expressions, entities, entities plus steps) is between the values of n_i and b_i of the correspondent Fuzzy number, in other words, it is located in a common range for two Fuzzy numbers (because it is also between a_{i+1} and p_{i+1}), it is necessary to compute the degree of membership the number has to each of the corresponding fuzzy numbers (or the complexity categories it represents), then Equation 10 needs to be applied.

$$dFSUCP(x) = (x).SUCP_i + (x).SUCP_{i+1} \quad (10)$$

where $dFSUCP(x)$ is the value obtained after the defuzzyfication of the number x , $SUCP_i$ is the value of SUCP for the category i and $SUCP_{i+1}$ is the value of SUCP for the category $i + 1$.

An empirical evaluation:

A experiment was done to evaluate the performance and growth of the proposed metrics. The study was completed with a real project database of a private company. The UCs, the FPs and

adjustment factors used and, the productivity of the team were provided. Four metrics were collected: FP, UCP and FSUCP. The FP for each UCs is not directly obtained from the base. It was important to compute them, considering the operation present in each UC. To compute the UCP, the actors and UCs were manually measured, after that, the adjustment process was done by considering the attributes presented in the actual FP analysis and the information provided by people involved in the project. To group the other metrics, a context was implemented. This context receives as input the description of the UCs in a XML (eXtended Markup

Language) pattern. This pattern elucidates the automatic calculation of the metrics. To make possible the use of the framework, the UCs was written with the proposed notation and transformed in XML documents. Five modules of the system, each one with respect to one month of work, were elicited to the analysis. The study was completed in two stages. The first one, named preparation stage, provides data about the productivity related to each metric. This stage incorporates only one module. Table 9 presents the productivity aligning found in this stage.

Table 9: Productivity Measures (Effort).

Measure	Value
FP/Hour	0.2235
UCP/Hour	0.0384
FSUCP/Hour	0.3979

By converting the productivity of 0.0384 UCP/h in terms of h/UCP, we acquired a productivity of 26 h/UCP. This value is very similar to that one recommended in (Mohammed Wajahat Kamal 2011). The productivity value acquired to FP is 0.2235 FP/h or 4.5 h/FP. The second stage, named analysis stage, includes the four other modules. In this phase, effort estimates were computed by using the productivity acquired in the preparation stage. The results of the analysis phase are in Table 10. To adjust the values of the metrics, the below values were used:

FP: 1.2

UCP: 0.64

USP e USFP: 1.05

The adjustment factor value for FP is the sum of the technical adjustment factors. This increments the

UFP. The adjustment factor value for UCP is the sum of the technical and environmental factors, which commonly decreases the UUCP. The adjustment factor value for SUCP and FSUCP also includes the aggregate of technical and environmental factors, but these last ones have the same weight as the first ones. This fact tends to increment the UUSP and UFUSP. With the productivity acquired in the preparation stage and with the values computed the next step is to outgrowth effort estimates and experience the metrics demean or, distinguishing the calculated and original effort. Each module effort is estimated in this way:

Effort = Module in size
(UCP,FSUCP)/productivity

Module	UCP	FSUCP	Actual
A2	281.21	407.11	1258
A3	351.52	801.69	1051
A4	676.25	931.29	1111
A5	351.52	891.58	1007

Module	UCP	FSUCP
A2	77.6	67
A3	66.5	22.7
A4	39.1	16
A5	65.1	11.4

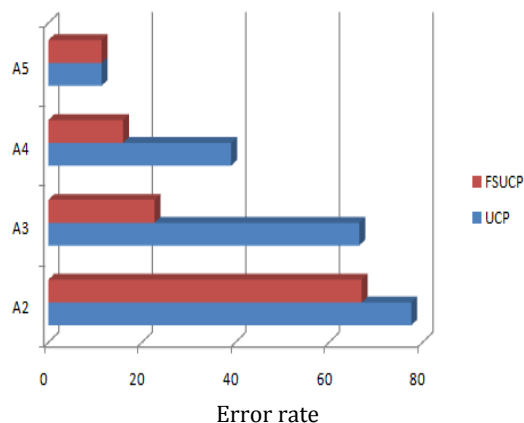
The outcome of the count is in Table 11. Figure 2 shows the error rates when distinguish the estimated values to the original ones. The percentage is in Table 12.

Conclusion:

The stipulation with UCs is one of the most used approach to document specification in object oriented systems. The metrics generally used dictate adversity and some stipulation in the process of calibrating software through UCs. This paper recommended a metric named FSUCP that uses the data supplied by the sections and elements of a UC to

find its size. And also uses logics of the Fuzzy Set Theory to constitute continuous classifications of the sections of a UC, eradicating some issues when classifying elements through tables of categories. The metrics present benefits when distinguished with UCP, like the possibility of acquiring counts for each UC distinctively and, the results shown by the empirical evaluation, illustrate that FSUCP provide better count than UCP. The next experiment was that the adjustment factors used in UCP did not provide a notable betterment in the accuracy of effort estimation. In addition, the model of 21 adjustment factors seemed to be necessary. In most cases, a

single additional predict or summed to the relapse model was enough to provide counts with similar accuracy as when FAT and FAE were used. Finally, based on the outcome of the performed factor analysis the number of adjustment factors might be lessens to 2 environmental factors and 4 technical complexity factors. And by resolving the FSUCP Metric with shortens number of adjustment factor result in exact estimation and it also disentangle the effort estimation process.



REFERENCES

- Ali Bou Nassif, 2012. 'Estimating Software Effort Using an ANN Model Based on Use Case Points', 11th International Conference on Machine Learning and Applications.
- Celar, S., 2013. 'Software size estimating method based on MK II FPA 1.3 Mk II Function Point Analyses' EBSCO Industries.
- Dian Pratiwi, Jl. Kyai Tapa, 2013. 'Implementation of Function Point Analysis in Measuring the Volume Estimation of Software System in Object Oriented and Structural Model of Academic System', International Journal of Computer Applications (0975 – 8887), 70(10).
- Jonathan Lee, Wen-Tin Lee, Jong-Yih Kuo, 2011/ 'Fuzzy Logic as a Basic for Use Case Point Estimation' IEEE International Conference on Fuzzy Systems, June 27-30, Taipei, Taiwan.
- Marcio Rodrigo Braz, Silvia Regina Vergilio, 'Software Effort Estimation Based on Use Cases', Computer Science Department, Federal University of Parana - UFPR, CP: 19081, CEP: 81531-970, Curitiba, Brazil.
- Azzeh, 2011. 'Analogy-based software effort estimation using fuzzy numbers', Journal of Systems and Software, 84: 270–284.
- Mohammed Wajahat Kamal and A. Moataz Ahmed, 2011. 'A Proposed Framework for Use Case based Effort Estimation using Fuzzy Logic: Building upon the outcomes of a Systematic Literature Review', International Journal on New Computer Architectures and Their Applications (IJNCAA), 1(4): 953-976. The Society of Digital Information and Wireless Communications, (ISSN: 2220-9085).
- Ochodek M., J. Nawrocki, 2010. 'Enhancing use-case-based effort estimation with transaction types', Foundations of Computing and Decision Sciences, 35(2): 91–106.
- Ochodek, 2010. 'Simplifying effort estimation based on Use Case Points', Journal of Information and Software Technology, 0950-5849, Elsevier B.V.
- Pauline, M., 2012. 'Software Cost Estimation Model based on Proposed Function Point and Trimmed Cost Drivers Using COCOMO II', International Journal of Engineering Research & Technology, 1(5): 1 – 7.
- Pauline, M., 2013. 'Layered model to Estimate Effort, Performance and cost of the Software projects', International Journal of Computer Applications, 63(13): 17 – 23.
- Performance Evaluation of Software Effort Estimation using Fuzzy Analogy based on Complexity', 2012. International Journal of Computer Applications 40(3):32-37, February 2012, Published by Foundation of Computer Science, New York, USA.
- Tharwon Arnuphaptrairong, 2013. 'Early Stage Software Effort Estimation Using Function Point Analysis: Empirical Evidence', Proceedings of the International MultiConference of Engineers and Computer Scientists 2013 Vol II, IMECS 2013, March 13 - 15, Hong Kong .
- Vikas Kumar, Sweta Pandey, 2013. 'Accurate Software Size Estimation Using the Updated Function Point Analysis Model', International Journal of Advanced Research in Computer and Communication Engineering 2(6).
- Ziauddin1, 2012. 'Efforts Estimation by Combining the Use Case Point and COCOMO', International Journal of Computer Applications (0975 – 8887), 52(7).