RESEARCH ARTICLE OPEN ACCESS

Cost Estimation Measurement in Software Development Based on Enhanced Layer Model

*Sanjida Nazneen, Saoud Sarwar
Department of Computer Science and Engineering, Faculty of Engineering and Technology
Al-Falah University, Dhauj, Faridabad, Haryana, INDIA
Corresponding Author: Sanjida Nazneen

ABSTRACT
Cost estimation in the early phase of the software development helps the developers to find out that how many software requirements would be implemented during the different releases of the software. Cost serves as an important criterion for the selection and prioritization of the software requirements. Different methods have been developed for the estimation of the software cost like lines of code (LOC), function point (FP), and constructive cost model (COCOMO) model. Based on our review, we identify that in literature less attention is given to the estimation of the cost of the requirements; and how these requirements are employed to find out the different parameters which are used for the estimation of the cost of the software, if it is developed by the Indian Software Industries. Therefore, in order to deal this issue, we proposed an enhanced layer based model for the estimation of the cost of the software. Finally, the proposed method is explained by the case study.

Keywords: Software Requirements, Cost Estimation, LOC, Function Point, COCOMO, and Goal Oriented Requirements Elicitation Methods

Date of Submission: 22-07-2017 Date of acceptance: 31-08-2017

I. INTRODUCTION

In software development companies, prediction of the cost of software estimation in early phase of software development helps the software companies to decide which requirements should be implemented [1].Practically, we cannot implement the entire requirements due to the budget problem. Therefore, it is important to estimate the cost of each software requirements so that the client can decide that from the given set of the requirements which requirements should be implemented [2, 3]. There are different methods for the estimation of the cost of the software like lines of code (LOC), function point (FP), and COCOMO model. Lines of code (LOC), also known as "*source lines of code*", is software metric which is used to measure the size of the computer program by counting the number of lines available in a program [4]. This metric is also used to estimate the effort required for the development of a software project. In real life applications, LOC received less attention by the software engineering community because it varies from one programming language to another program for the same application. For example, if you have developed a project "S" in C language and also in C++, then it has been observed that in both the cases the LOC would be different but the functionality of the S would be same. Therefore, to address this problem different types of the software matrices were developed like "*function point*" and "*COCOMO model*", etc. [4].

Function Point (FP), proposed by Allan Albrecht in 1970, is a software matric which is used to describe the functionality of the software. In 1986, "*The International Function Point User Group*" (IFPUG) was developed to promote and circulate the effective management of software development by using FP analysis (FPA) [5]. In literature, FP is also known as IFPUG FPA. FPA is standardized by the ISO/IES 20926: 2010. This metric is also used to compute the "*development effort*", "*productivity*", and "*cost*" [5]. In 1981, Constructive Cost Model (COCOMO) model was proposed by Barry Boehm for the estimation of the cost of the software.

Layer based methods in the area of software engineering have been popular for the development of the software product. For example, Lin *et al.* [6] proposed a "layer based method for the rapid development of the software product. In their work authors have employed the guidelines suggested by the Extreme Programming (XP). XP requires highly expressive programming language, i.e., JAVA and CASE Tools. Different methods have been developed on the basis of layer based concepts [6, 7, 8, 9]. Based on our review, we identify that in literature less attention is given to the estimation of the cost of the requirements; and how these requirements are employed to find out the different parameters which are used for the estimation of the cost of the software, if it is developed by the Indian Software Industries. Therefore, in order to deal this issue, we proposed an enhanced layer based model for the estimation of the cost of the software. Finally, the proposed method is explained by the case study.

The remaining part of this paper is organized as follows: In section 2, related work is given. We explain the proposed method in Section 3. In section 4, case study is used to explain the proposed method. Finally, conclusions and the future work are given in section 5.

II. RELATED WORK

A layer based development method proposed by Lin et al. [6] includes three steps, i.e., (i) the use case identification (ii) the architectural specification, and (iii) architectural component construction. In their method, use case was used to represent the customer requirements. Architectural framework was used to identify the view layer components, control layer components, and model layer components. In the third step, the JAVA code was used for architectural component construction. In 2013, Pauline et al. [7] developed an enhanced model for the estimation of the effort, cost, and performance of the software projects. In their work, authors have considered three types of the complexities, i.e., system complexity, input/output complexity, and application complexity. Waheed *et al.* [8] apply the enhanced model concept in the web services in which the authors deal the issue of the security. Authors have divided the layers into low level to high level. The objective of the low level is to provide more interaction; and high level is used to provide the less attention among the services. In their work, they have included a layered model between service and service consumers. In similar studies, Rijwani and Jain [9] apply the multi-layered feed forward artificial neural network technique for the enhancement of software effort estimation. Based on our studies, we identify that in the literature of the software engineering. In 2008, Gupta *et al.* [10] proposed a "*Software Estimation Tool Based on Three-Layer Model for Software Engineering*". In the first layer, i.e., basic metrics, FP was used for object oriented system. General software metrics was used as second layer. In this layer authors have computed LOC, effort, time, and cost using COCOMO II. In the third layer, maintenance and quality were used as advanced software metrics. Based on our literature review, we find out that in these studies less attention is given to consider the relationship between LOC and FP. Therefore, in-order to address this issue, in this paper we proposed a four layered approach for the estimation of the software cost.

III. PROPOSED METHOD

In this section, we present the layered approach for the estimation of the software cost. The block diagram of the proposed approach is given in Fig. 1.

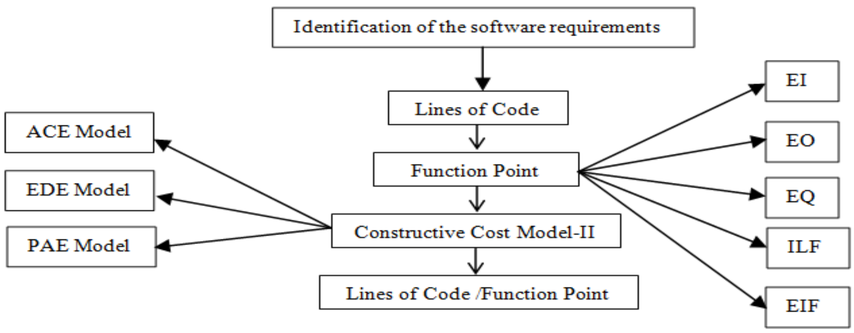

Fig. 1: Proposed layered approach for the estimation of the software

Layer 0: The objective of this layer is to identify the requirements of software. There are different methods to identify the requirements like "*traditional methods*", "*group elicitation method*", "*cognitive method*", "*contextual method*", and "*goal oriented requirements elicitation method (GOREM)",* etc. In our work, we employed GOREM to identify the requirements of the software. In GOREM, the high level objective of an organization are refined and decomposed into sub-goals. These sub-goals ate further refined and decomposed into sub-goals; and this process continues till the responsibility of the last sub-goals are assigned to some agents or some system [2, 3]. In GOREM, we find out the requirements by constructing the AND/OR graph of all the goals and sub-goals [2].

Layer 1: In Fig.1, the Lines of Code (LOC) is at the first layer. The objective of this layer is to compute the cost of the software based on the values of the LOC.

Layer 2: In this layer, the function point (FP) is used to describe the functionality of the systems form the user's point of view. Albrecht's proposed the concepts of FP in which the system is decomposed into the following units: (i) "*External Inputs*" (EI), (ii) "*External Outputs"* (EO), (iii) "*External Queries*" (EQ), (iv) "*Internal Logical Files*" (ILF), and (v) "*External Interface Files*" (EIF). After identifying the above five parameters, the unadjusted FP (UFP) are calculated by using the weights factors, as shown in Table 1.

Table 1: Functional Units with weighting factors

Functional Units	Weighting Factors		
	Low	Average	High
EI	3	4	6
EO	4	5	7
EQ	3	4	6
ILF	7	10	15
EIF	5	7	10

The FP is calculated by multiplying the UFP with "*complexity adjustment factor*" (CAF), as shown in following equation:

$$FP = UFP \times CAF \quad (1)$$

Where CAF is computed by using the following equation:

$$CAF = \left[0.65 + 0.01X \sum_{i=1}^{i=14} F_i\right] \quad (2)$$

In equation (2), F_i are the general system characteristics. The detailed descriptions about these characteristics are given below:"(F-1) Does the system require reliable backup and recovery? (F-2)Is the data communication required? (F-3) Are there distributed processing functions? (F-4) Is performance critical? (F-5) Will the system run in an existing heavily utilized operational environment? (F-6) Does the system require online data entry? (F-7) Does the online data entry require the input transaction to be built over multiple screens or operations? (F-8) Are the master files updated on line? (F-9) Are the inputs, outputs, files, ore inquiries complex? (F-10) Is the internal processing complex? (F-11) Is the code designed to be reusable? (F-12) Are the conversion and installation included in the design? (F-13) Is the system designed for multiple installations in different organizations? (F-14) Is the application designed to facilitate change and ease by the user?"

Layer 3: In third layer, the COCOMO-II is used for the estimation of the software. COCOMO II is the revised version of COCOMO, as developed by the Barry Boehm. There are three stages in the COCOMO II, i.e., "(1) Application Composition Estimation (ACE) model (2) Early Design Estimation (EDE) model (3) Post Architecture Estimation (PAE) model". In ACE model size is estimated by using object points. There are three objects in ACE model, i.e., screen, reports, and 3GL modules. This model is designed for quickly developed applications using interoperable components, i.e., "*graphic user interface*" (GUI) builders, database or object managers,hypermedia handlers, smart data finders, and domain specific components, etc.There are seven early design cost derivers in EDE model, i.e., "(1) Product reliability and complexity (2) Required reuse (3) Platform difficulty (4) Personal capability (5) Personal experience (6) Personal, and (7) Schedule". In PAE model is employed when the software life cycle architecture has been completed. There are 17 cost drivers of the PAE model, i.e., "(1) Reliability required (2) Database size (3) Product complexity (4) Reusability (5) Documentation (6) Execution time constraints (7) Main storage constraints (8) Platform volatility (9) Analyst capability (10) Programmers capability (11) Personal Continuity (12) Analyst experience (13) Programmer experience (14) Language and tool experience (15) Use of software tools (16) Site locations and communication technology between sites and (17) Schedule".

Layer 4: In the fourth layer, the ratio of LOC and the FP would be used for the estimation of the cost of the software. For example, the standard relationship for the LOC/FP = 128, if the software is written in C language; and LOC/FP = 64, if the software is written in C++ Language. Proposed method is explained in the next section by considering a case study based on Institute Examination System (IES).

IV. CASE STUDY

In this section, we explain the proposed method by considering the Institute Examination Systems (IES).The objective of IES is to provide all the facilities to the students related with the examination.

Layer 0: In this layer, we identify the requirements of IES with the help of GOREM. Therefore, to identify the requirements, we first refine and decompose the IES into sub-goals; and as a result we have identified the following sub-goals:

Sub-goal 1.0: Login module
Sub-goal 1.1: To download the list of the eligible students for the end-semester examination
Sub-goal 1.2: To generate the hall ticket for the end semester examination or backlog paper, if any
Sub-goal 1.3: To deposit the end semester examination fee or backlog paper fee, if any
Sub-goal 1.4: To generate the examination date sheet for all the papers including backlog papers
Sub-goal 1.5: To generate the seating arrangement by identifying the room number and the block number
Sub-goal 1.6: Mark-sheet generation
Sub-goal 1.7:Request for Re-Evaluation of the paper after depositing the required fee

Layer 1: The objective of this layer is to find out the LOC in the student module. As we know that LOC vary from one language to another language. Therefore, to find out the LOC in the module, we will have to first compute the FP of the student module. The computation of the FP of the student module is given in the next layer, i.e., layer 2.

Layer 2: In our study, we compute the cost of the sub-goals 1.0, i.e., login module; and sub-goals 1.2, i.e., to download the list of the eligible students for the end-semester examination.To find out the value of the function point (FP), we identify the value of the EI, EO, EQ, ILF, and EIF. To find out the value of these parameters, we first visualize the sub-goal 1.0 and sub-goal 1.1 in the same way as it would be represented on the computer screen after the implementation, as shown in Fig. 2.

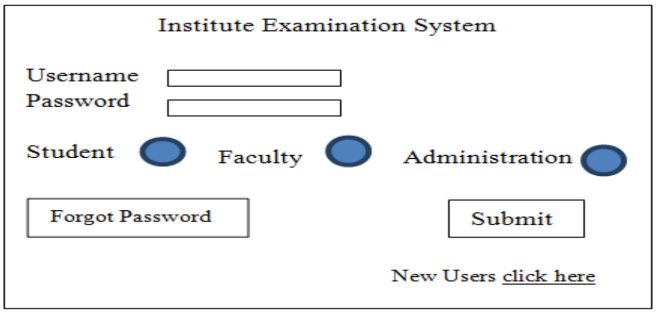

Fig. 2: Sub-goals 1.0: Login Module

After visualizing the sub-goal 1.0, the next step is to find out the value of the EI, EO, EQ, ILF, and EIF. As a result, we have identified the following values for the given two sub-goals:

For Sub-goal 1.0: There are seven inputs, i.e., Username, Password, Student, Faculty, Administration, Submit, and Forgot Password (if already a user). For new users eight new inputs would be used, i.e., Name, Fathers Name, Date of Birth, Name of the Department, Date of Joining, Permanent Address, Correspondence Address,User name. There would be three outputs depending on the type of users. There would be one query, i.e., are you a student or employee of this University?; There would be one database two store the record of the students. There is no external interface file. Finally, we have got the following values:

EI = 15; EO = 3; EQ = 1; ILF = 1; EIF = 0

Table 2: Calculations for the UFP for sub-goal 1.0 and sub-goal 1.1

Functional Units	Sub-goal 1.0		Sub-goal 1.1	
EI	15	4	25	4
EO	3	5	1	5
EQ	1	4	1	4
ILF	1	10	1	10
EIF	0	7	0	7
UFP	89		119	

For Sub-goal 1.1: In the similar way, we have computed the following values for sub-goal 1.1:

EI = 25; EO = 1; EQ = 1; ILF = 1; EIF = 0

In our study, we assume the average weighting factors; and the values of the general system characteristics are also average. The value of the unadjusted weighted factors for sub-goal 1.0 and sub-goal 1.1 are given in Table 2.

Finally, the values of the CAF for the sub-goal 1.0 and 1.1 are given below:

CAF = [0.65+0.01X56] = 0.65+0.56 = 1.21 (for sub-goals 1.0 and 1.1)

Finally, the value of the FP for sub-goal 1.0 = 89 X 1.21 = 108 (approximately); and the value of the FP for sub-goal 1.1 = 119X1.21 = 144 (approximately).

On the basis of the case study of the Chrobot [12], we identify that the cost of the implementation of the one FP in Indian software industries is $125. Therefore, on the basis of this result, we compute the cost of the sub-goals 1.0 and 1.1. As a result, we have the following values:

The cost of sub-goal 1.0 = 108 X125 = $ 13500 and the cost of sub-goal 1.1 = 144X 125 = $ 18000.

Layer 3: In the similar way, we can compute the cost of the sub-goals by applying the COCOMO-II; and would be discussed in our future work.

Layer 4: On the basis of the results of FPs, we can compute the LOC in sub-goal 1.0 and 1.1 if these goals would be implemented using C and C++.
The LOC in sub-goal 1.0 would be 108 X 128 = 13, 824 LOC, if this goal would be implemented using C language. There would be 108 X 64 = 6912 LOC, if the same goal would be implemented using C++ language. In the similar way, we calculate the LOC for sub-goal 1.1. In this goal there would be 144X128 = 18, 432 LOC, if it is implemented using C language; and there would be 144X64 = 9216 LOC, if it is implemented using C++ Language.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a layer based approach for the estimation of the cost of the software. In the proposed method there are five layers, i.e., layer 0-identification of the software requirements, layer 1-lines of code, layer 2-function point, layer 3-constructive cost model- II, and layer 4- LOC/FP. In our work, we have identified 8 sub-goals. We have applied the proposed method to compute the cost of the two sub-goals, i.e., sub-goal 1.0 and sub-goal 1.1. After applying the FP, we found that the cost to implement the sub-goal 1.0 and 1.1 are $ 13500 and $18000, respectively. In future work, we will implement the cost of all the sub-goals; and we also apply the COCOMO-II for the computation of the cost of all the sub-goals.

REFERENCES

[1] Lavazza L, Morasca S, and Robiolo G. 2013. Towards a Simplified Definition of Function Points. Information and Software Technology, Vol. 55, pp. 1799-1809, 2013.

[2] Sadiq M and Jain S. K. 2015.A Fuzzy Based Approach for the Selection of Goals in Goal Oriented Requirements Elicitation Process. International Journal of Systems Assurance Engineering and Management, Springer, Vol. 6, Issue 2, pp. 157-164, June 2015.

[3] Sadiq and Neha. 2016. Elicitation of Testing Requirements from the Selected Set of Software's Functional Requirements using Fuzzy Based Approach. International Conference on Computational Intelligence in Data Mining, Springer, organized by KIIT University, Bhubaneswar, Odisha, India.

[4] Pressman R. S. 2001. Software Engineering-A Practitioners Approach. 5th Edition, McGraw-Hill Higher Education.

[5] Junior M. D. F., Fantinato M, and Sun V. 2015. Improvements to the Function Points Analysis Method: A Systematic Literature Review. IEEE Transactions on Engineering Management, Vol. 62, No. 4, 2015.

[6] Lin L., Yang W, and Lin J. 2012. A Layer Based Method for Rapid Software Development. Computers and Mathematics with Application, Vol. 64, pp. 1364-1375.

[7] Pauline M., Aruna P., and Shadaksharappa B. 2013. An Enhanced Model to Estimate Effort, Performance, and Cost of the Software Projects. ICTACT Journal on Soft Computing, Vol. 3, Issue 3, pp. 524-533.

[8] Waheed S. Z., Wadood A., and Azam F. 2014. Enhanced Layer Based Model in Support of Web Services Interoperability. VAWKUM Transactions on Computer Sciences, Vol. 5, No. 1, 2014.

[9] Rijwani P. and Jain S. 2016. Enhanced Software Effort Estimation using Multi-layered Feed Forward Artificial Neural Network Technique. Procedia Computer Science, Elsevier, Vol. 89,pp. 307-312, 2016.

[10] Gupta D., Kaushal S. J., and Sadiq M. 2008. Software Estimation Tool Based on Three-Layer Model for Software Engineering. 4th IEEE International Conference on Management of Innovation and Technology, pp. 623-628, Bangkok, Thailand.

[11] Sadiq M. 2017. A Fuzzy-Set Based Approach for the Prioritization of Stakeholders on the Basis of the Importance of Software Requirements. IETE Journal of Research, Taylor and Francis, 2017. DOI: http://dx.doi.org /10.1080/03772063.2017.1313140

[12] Chrobot B. C. What is the Cost of One IFPUG Method Function Point? Case Study. Proceedings in the World Congress in Computer Science, Computer Engineering, and Applied Computing 2012

International Journal of Engineering Research and Applications (IJERA) is **UGC approved** Journal with Sl. No. 4525, Journal no. 47088. Indexed in Cross Ref, Index Copernicus (ICV 80.82), NASA, Ads, Researcher Id Thomson Reuters, DOAJ.

Sanjida Nazneen. "Cost Estimation Measurement in Software Development Based on Enhanced Layer Model." International Journal of Engineering Research and Applications (IJERA), vol. 7, no. 8, 2017, pp. 35–39.