Software Cost Estimation using Function Point: An Extended Approach

[1]Amrita Saxsena, [2]Saoud Sarwar
[1]M.Tech. Scholar (Corresponding author)
Department of Computer Science and Engineering, School of Engineering and Technology, Al-Falah University, Dhauj, Faridabad, Haryana, India

Abstract: Software cost estimation is a key research area of software engineering which is used to estimate the cost of the software. During software development process, it has been observed the requirements, which are captured by the software requirements analyst, are implemented so that the need of the stakeholders can be satisfied. Based on our literature review of software cost estimation, we identify that less attention is given to estimate the cost of the requirements of software and the computation of the general system characteristics (GSC). GSC can be evaluated under two different environment, i.e., crisp environment and fuzzy environment. Therefore, to address this research issue we proposed an extended approach for the estimation of the cost of the software. Finally, we present an example to show the applicability of the proposed method.

Keywords: Software requirements elicitation, functional requirements, non-functional requirements, library management system.

1. Introduction

Software engineering cost models and estimation techniques are used for a number of purposes. These include: *Budgeting*, Trade-off and risk analysis, Project planning and control, Software improvement investment analysis[1]. Software Cost estimation is an important research area in which we estimate the cost of software with different methods like COSMIC, FiSMA, IFPUG, MARK-II, NESMA. Among these methods IFPUG, i.e., International Function Point User Group has received much attention by the software industry [1,2]. IFPUG Function point (FP) was proposed by Allan Albrecht at IBM in 1979 to express the amount of business functionality of an information system. The objective of the FP is to describe the functionality of the software on the basis of different measuring parameters, i.e., external inputs; external outputs, external query, internal logical file, and external interface file [2].

More than 25 methods based on FP have been proposed by the research community; and among these methods, only five methods, i.e., IFPUG FP, Mark II, NESMA FP method, COSMIC FP, and FiSMA FP method, have received the recognition by the International Standardization Organization (ISO) and International Elecrotechnical commission (IEC) because it comply with the rules contained in the ISO/IEC 4143 norm. Table 1 presents the list of five FP methods along with ISO/IEC standardization [3].

Table 1: List of five FP methods along with ISO/IEC standardization [4]

S. No.	FP Methods	ISO/IEC standardization
1	IFPUG Function Point	ISO/IEC 20926
2	Mark II	ISO/IEC 20968
3	Netherland Software Metrics Association (NESMA) FP method	ISO/IEC 24570
4	Common Software Measurement International Consortium (COSMIC) FP method	ISO/IEC 19761
5	Finnish Software Measurement Association (FiSMA) FP method	ISO/IEC 29881

There are different applications of FPA in the area of software engineering, i.e., software size measurement, effort computation, productivity, and cost estimation [5, 6]. FPA is a two- step process, i.e., (i) computation of unadjusted function point (UFP) and (ii) computation of complexity adjustment factors (CAF) which is based on fourteen general system characteristics (GSC).

Software requirements elicitation is the first sub-process of software requirements engineering whose objective is to identify the need of the stakeholders so that a successful software system can be developed [7]. Stakeholder identification (SI) is an important activity of requirements elicitation techniques. Pacheco and Garcio [7] performed a systematic literature review of stakeholder identification methods on the basis of the following research questions: (a) "*methods and techniques to carry out the stakeholder identification in requirements elicitation*", (b) "*effective practices recommended for performing SI*", and(c) "*what are the consequences of incorrect SI on the quality of Software Requirements?*". If any stakeholder will participate without any real need then the requirement of the system may be incorrect and it may lead to failure of the software because it will rise to inconsistent specifications.

Interview is a part of traditional method which is used to elicit the information for the system-to-be. Before starting the elicitation interviews it is important to select the topic of the discussion so that the probability of missing the important requirements would be minimized. In order to select the topic of the interest for the interview, elicitation topic map (ETM) may be used. ETM is a kind of diagram which shows the topic of interest that would be discussed during interviews; and it also shows that how the different types of the stakeholders discuss different topics spontaneously. If a topic is not discusses spontaneously, than the results of the ETM suggest that the requirements analyst should prepare the questions [8]. Gathering of stakeholders requirements is a critical stage of the software development process. One of the problems of the requirements elicitation is the communication problem. In real life application, several stakeholders are involved and they have different communications because stakeholders may belong to different places and countries [9].

In literature, we have identified different methods to elicit the software requirements, i.e., traditional methods, group elicitation method, package oriented method, goal oriented method, cognitive method, and contextual method. Among these methods, traditional methods are used to know the background of the system- to- be with the help of the following techniques, i.e., survey, questionnaire, analysis of existing documents.

Goal oriented method like "*knowledge acquisition for automated specification*" (KAOS), "*non-functional requirements*" (NFR) framework, "*attributed goal oriented requirements analysis*" (AGORA) method, "fuzzy attributed goal oriented software requirements analysis method" (FAGOSRA) etc. have received much attention by software requirements elicitation community for the elicitation and analysis of the software requirements using goal concepts [10]. Goal oriented methods are used to visualize the goal of the stakeholders by constructing the AND/OR graph.

The remaining part of the paper is structured as follows: In section 2, we present the related work. Proposed method is given in section 3. Application of the proposed method as case study is given in section 4. Finally, the conclusions and the future work are given in section 5.

2. Related work

In this section we present the related work in the area of software cost estimation models. Matson *et al.* [11] present an "assessment of several published statistical regression models that relate software development effort to software size measured in function points. The principal concern with published models has to do with the number of observations upon which the models were based and inattention to the assumptions inherent in regression analysis. The research describes appropriate statistical procedures in the context of a case study based on function point data for 104 software development projects and discusses limitations of the resulting model in estimating development effort".

Kemerer [12] evaluated the four of the most popular algorithmic models used to estimate software costs, i.e., Software Life Cycle Management (SLIM), Constructive Cost Models (COCOMO), Function Points, and ESTIMACS, i.e., Estimation by Management and Computer Services. Boehm et al [1] conducted a survey of different software cost estimation models and classify the methods into the following types: parametric models, expertise-based techniques, learning-oriented techniques, dynamics based models, regression-based models, and composite-Bayesian techniques for integrating, expertise-based and regression-based models. In 2007, Jorgensen and Shepperd [13] conducted a systematic literature review on software development cost estimation models. Heemstra and Kusters [14] discuss the merits of function point analysis. For the analysis point of view, they have used the large survey of Dutch Organization. On the basis of their analysis, authors found that FPA is reasonable for product size but adjustment part of the model is less useful. Symons [15] discusses the difficulties and improvements in the area of function point. Fetcke *et al.* [16] mapped the object oriented Jacobson approach into function point analysis. Kemerer and Porter focused on improving the reliability of the function point analysis. On the basis of our literature review, we identify that in the literature crisp data is used during the computation of the function point. In real life application, software developers may use the fuzzy data during the evaluation of the 14 general system characteristics. Several researchers have advocated adding other general system characteristic. For example, Junior et al. [17] presented a systematic literature review on the improvements of the function point analysis; and as a result they discuss the need to add new GSC according to the need of the software project.

In software engineering (SE) research, case studies play an important role to validate the results of the proposed systems. In SE research, we have identified that Institute Examination System, Institute Library Management System (LMS), Time Table problems are most studied problems [10, 18]. Therefore, in our work, we focus on the Institute LMS.

3. Proposed Methods

In this section, we present an extended approach for the estimation of the cost of the software.
Proposed method includes the following steps:

- Step 1: Selection of the software requirements elicitation method
- Step 2: Identify the functional and non-functional requirements
- Step 3: Compute FP of the software requirements
- Step 4: Compute the Cost of each requirement
- Step 5: Add some general system characteristics, if required
- Step 6: Apply fuzzy based method for the evaluation of GSC if developers use linguistic variables

Step 1: Selection of the software requirements elicitation method

Requirements elicitation or identification is an important sub-process of software requirements engineering which is employed to identify the need of the different types of the stakeholders so that a successful system can be developed. In literature, we have identified different methods to elicit the software requirements, i.e., traditional methods, group elicitation method, package oriented method, goal oriented method, cognitive method, and contextual method. Among these methods, traditional methods are used to know the background of the system- to- be with the help of the following techniques, i.e., survey, questionnaire, analysis of existing documents. Goal oriented methods are used to visualize the goals of the stakeholders by constructing the AND/OR graph. In large software development projects, different stakeholders participate and they produce a huge number of unstructured text documents and these documents are transformed into structured requirements. The conversion from unstructured to structure is a time consuming and error prone. Different tools have been developed to automate the requirements elicitation techniques. Therefore, Meth *et al.* [19] present a systematic literature review in the area of automated requirements elicitation. Carrizo *et al.*[20] presented a method for the selection of software requirements elicitation techniques. Hickey and Davis [21] focused on requirements identification; and the methods for the selection of requirements elicitation techniques. Lloyd *et al.* [22] discussed the effectiveness of the software requirements elicitation techniques in the area of distributed requirements engineering. Apart from this, different methods have been developed to elicit the software requirements using goal oriented methods. In goal analysis, goals are used for the requirements identification and refinements to guide the software development process. Different types of the goals may be expressed in different levels of abstraction; and then it is decomposed into sub-goals. Different methods have been developed to decomposed and refined the goals into sub-goals in the literature of goal oriented requirements engineering like "*knowledge acquisition for automated specification*", (KAOS), i* framework, NFR framework, "*attributed goal oriented requirements analysis*" (AGORA), and "*goal oriented idea generation*" (GOIG) methods, etc. [10, 18, 23, 24, 25, 26]. In 2018, Mohammad *et al.* [10] proposed an extended version of AGORA, i.e., "*fuzzy attributed goal oriented software requirements analysis (FAGOSRA) with multiple stakeholders*" for the analysis of the software requirements in which two different values were used in AND/OR graph, i.e., (i) fuzzy preference matrix and (ii) fuzzy contribution values. A case study based on Institute Examination System was used to demonstrate the proposed method. Therefore, in our work, we focus on the traditional methods and goal oriented methods for the elicitation of the software requirements.

Step 2: Identify the functional and non-functional requirements
In the proposed method, we apply the traditional method and goal oriented method to identify the different types of the software requirements, i.e., "*functional requirements*" (FR) and "*non-functional requirements*" (NFR). FR is employed to describe "*what the system is supposed to do*" while NFR is used to describe the non –behavioural aspects of the system. It means "*how the system is supposed to be*".

Step 3: Compute FP of the software requirements
Computation of UFP depends on the weighting factors of the following functional units (FU), i.e. External Inputs (EI), External Outputs (EO), External Inquiries (EI), Internal Logical File (ILF), and External Interface File (EIF); and also on the following general system characteristics, represented by F_i , where I = 1 to 14 [2].

1. "Does the system require reliable backup and recovery?
2. Is data communication required?
3. Are there distributed processing functions?
4. Is performance critical?
5. (v)Will the system run in an existing heavily utilized operational environment?
6. Does the system require on line data entry?
7. Does the online data entry require the input transaction to be built over multiple screens or operations?
8. Are the master files updated online?
9. Is the inputs, outputs, files, or inquiries complex?
10. Is the internal processing complex?
11. Is the code designed to be reusable?
12. Are the conversion and installation included in the design?
13. Is the system designed for multiple installations in different organization?
14. Is the application designed to facilitate change and ease of use by the user?"

Now CAF is calculated by using the following formula:

$$\mathrm{CAF}=0.65+0.01\mathrm{X}\sum_{i}^{14} F_i \qquad (1)$$

whereF_i (i=1 to 14) are the ratings of each factor on a scale of 0 to 5. FP is calculated by using the following formula:

$$FP = UFPXCAF \qquad (2)$$

Step 4: Compute the cost of each requirement
Once we have identified the values of the FP for each requirement then the next step is to compute the cost of the requirements. The cost of the software requirements depends on the value of the FP. The cost of the implementation of one FP varies from one country to another country. For example, the cost of the implementation of the one FP in India is USD 125 while the cost of the same requirement in USA, Japan, and Sweden are USD 1000, USD 1600, and USD 1500, respectively [27]. If in our project, there are P number of FP then the cost of the implementation of the project would be USD 125*P.

Step 5: Add some general system characteristics, if required
In traditional FP method, 14 GSC are used to understand the non-behavioural aspect of the system (NBAS). In today's environment, these GSC are not sufficient to understand the NBAS. Keeping in view the need of the new NBAS, different authors have shown their concern to add some new GSC according to the demand of the software project. For example, Ahn *et al.* [6] proposed the inclusion of ten new GSCs focused on software maintenance, i.e., (a) "*people perspective—engineer's skills*", (b) "*product perspective—technical characteristics*"(c) "*process perspective—environment characteristics*".

Step 6: Apply fuzzy based method for the evaluation of GSC if developers use linguistic variables
In most of the studies, we found that crisp data is used to calculate the value of the GSC on the scale of 0 to 5. Sometimes the developers may use the linguistic variables to evaluate the GSC, for example,"*Does the system require reliable backup and recovery"* should be very high. Here, the term very high is a fuzzy term. Therefore, to deal with this kind of linguistic variables, in the proposed method, we use the fuzzy based method to evaluate the GSC under fuzzy environment [4].

4. Case study

The objective of this step is to apply the steps of the proposed method for estimation of the cost of Institute Library Management System (ILMS).ILMS is used to issue the existing books to the authentic students of a University or Institution. ILMS will provide the facility to search the existing books in some particular subjects, for example, Data Structures, Software Project Management, Brain Computing, etc. Students will search the books using ILMS; and as a result the system will give you the information about the availability of the books and when the issued book would be returned by the students. On the basis of the ISSN and ISBN numbers, the students can also search the books. For M. Tech. and Ph.D. research scholars, ILMS will provide you the facility to download the research papers of conference proceedings and journals.

Step 1: Selection of the software requirements elicitation method
Traditional method is an important requirements elicitation method which is used to identify the limitations in the existing system. Broadly speaking, traditional methods are divided into four parts, i.e., survey, questionnaire, analysis of existing documents, and interview. In this step, we mainly focus on the analysis of the existing documents because it is used to identify the limitations of the existing system, if any. The limitations in the existing system would be used as the requirements for the system-to-be. On the basis of our analysis, we identify the limitations in the existing systems; and these limitations are given below:

1. There is no support of the messages on the students mobile numbers or E-mail ID at the time the students return the book
2. There should be token number for the allotment of the books. If student X has requested some book then he/she should get the token number. This token number should be used a priority for the allotment of the books.
3. If there is some delay in the submission of the books by the students the same should be informed to the Librarian in advance. So that this information can be given to that student who is waiting for that book.

Step 2: Identify the functional and non-functional requirements

We have identified the following requirements of ILMS:

1. Login module
2. Search the books
3. Search the conference proceedings and journals
4. Token number for the issuing the books

The corresponding AND/OR graph for ILMS is given in Fig. 1. The objective of the AND/OR graph is used to decompose and refine the goals into sub-goals so that the requirements of the software can be identified. In the terminology of the goal oriented, requirement is that sub-goal which can't be decomposed and refined.

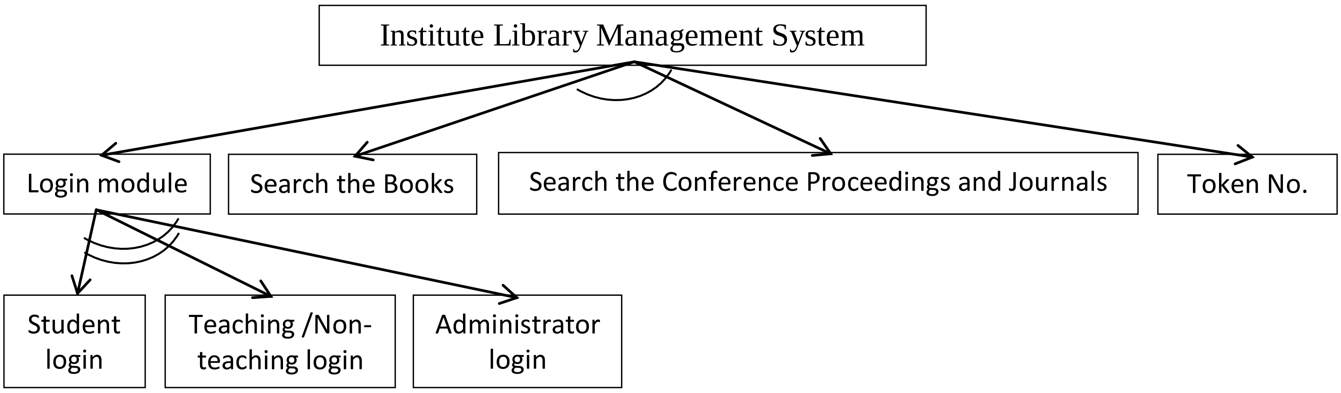
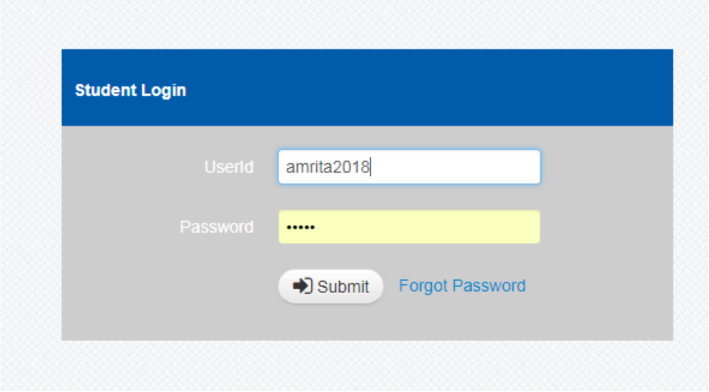

Institute Library Management System
Login module
Search the Books
Search the Conference Proceedings and Journals
Token No.
Student login
Teaching /Non-teaching login
Administrator login

Fig. 1: An abstract view of AND/OR graph for Institute Library Management System

In Fig.1, there are four sub-goals of ILMS, i.e., (i) Login module, (ii) Search the books, (iii) Search the conference proceedings and journals, and (iv) Token number for the issuing the books. There is an AND connection among these sub-goals. It means that until and unless all the sub-goals will not be executed, the goals of ILMS can't be executed. These goals can further be refined and decomposed into sub-goals; and after refinement there might be AND decomposition and OR decomposition. For example, login module is refined and decomposed into three sub-goals, i.e., student login, teaching/non-teaching login, and administrator login. These three sub-goals are connected to OR logical connective. It means that the achievement of any three sub-goals will lead to the achievement of the parent goals, i.e., login module.

Step 3: Compute FP of the software requirements

To identify the different measuring parameters used in the computation of the unadjusted function point (UFP), we visualize the software requirements in the same as it would come on the computer screen after the implementation. Here, we first compute the cost of the "***student login module***". The visual representation of the student login module is exhibited in Fig. 2. On the basis of the Fig. 2, we identify the following measuring parameters for the student login module: EI = 2, EO = 1, EQ = 1, ILF = 1, and EIF = 0. Now we compute the value of the UFP by considering average complexity of the software requirements; and the results are shown in Table 2.


Student Login

Userid: amrita2018

Password: •••••

Fig. 2: Visual representation of the student login module

Table 2: Computation of the UFP for student login module

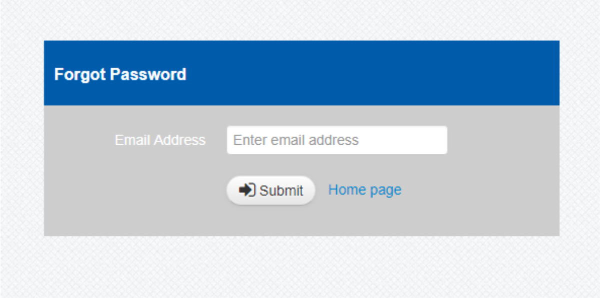
Measuring Parameters	Values	Average Complexity	Calculation
EI	2	4	8
EO	1	5	5
EQ	1	4	4
ILF	1	10	10
EIF	0	7	0
UFP			27

Before applying the equation 1 to compute the CAF, we first evaluate the value of the 14 GSC on the scale of 0 to 5. Here we assume that all the GSC are average. Therefore, “4” is used in the calculation.

CAF = 0.65 + 0.01 x (14 x 4)
= 0.65 + 0.01 x (56)
= 1.21

FP = UFP x CAF
= 27 x 1.21
= 32.67

Here, we are developing the software in India, so the cost of the implementation of one FPwould be USD 125. Finally the cost of the “*student login module*” is 32.67 FP = 32.67 x 125 = USD 4083.


Forgot Password

Email Address: Enter email address

Fig. 3: Visual representation of the forgot password option of student login module

On the basis of the visual representation of the forgot password option of student login module, as shown in Fig. 3, we identify the five different values of the measuring parameters which are used to compute the UFP, i.e., EI = 1, EO = 0, EQ = 1, ILF = 1, and EIF = 0. Here, we also assume the average complexity of all the five measuring parameters; and the results are given in Table 3.

Table 3: Computation of the UFP for forgot password of student login module

Measuring Parameters	Values	Average Complexity	Calculation
EI	1	4	4
EO	0	5	0
EQ	1	4	4
ILF	1	10	10
EIF	0	7	0
UFP			18

Similarly, we compute the value of the FP, as we have already computed for the student login module. As a result, we found that the value of the FP = 21.78. The cost of the implementation of the "*forgot password*"option of student login module in India is USD 2722.

Step 5: Add some general system characteristics, if required

It has been observed that security cost is always excluded in most of the parametric cost estimation models because it's assumed that security aspect is normally considered during late phase of software development process [28]. To overcome this problem, in the proposed method we include the cost of the security during the FP calculation. Instead of using the 14 GSC, we include 15 GSC by considering the security for those requirements which needs more security attentions, for example, in the Login module. Suppose the cost of including the security is T. Then the cost of the login module after adding the 15th GSC would be USD (4083+T).

Step 6: Apply fuzzy based method for the evaluation of GSC if developers use linguistic variables

In our case study, we assume that developers have used crisp data to evaluate the GSC. Therefore, we have not considered the fuzzy based methods for the estimation of the values of the GSC. In future work, we shall apply the fuzzy based methods to calculate the values of the GSC.

5. Conclusions and future work

In this paper, we present a method to compute the cost of the software requirements. After constructing the AND/OR graph, we have identified four sub-goals for ILMS, i.e., (i) login module, (ii) search the books, (iii) search the conference proceedings and journals, (iv) token number for the issuing the books. Proposed method is a six step process, i.e., (a) selection of the software requirements elicitation method, (b) identify the functional and non-functional requirements, (c) compute FP of the software requirements, (d) compute the Cost of each requirement, (e) add some general system characteristics, if required, and (f) apply fuzzy based method for the evaluation of GSC if developers use linguistic variables. In our work, we have elicited the software requirements of ILMS using traditional methods and goal oriented methods; and as a result, we have identified different requirements of ILMS. We compute the cost of the student login module and as a result we identify that the cost of the implementation of the student login module is USD 4083, if it is implemented in India. Future work includes the following:

1. To apply fuzzy based approach with real life example like Railway Management System, ATM System, etc.
2. To add some new GSC to improve the usability, reliability, etc.

References

1. Boehm B., Abts C., Chulani S., "Software Development Cost Estimation Approaches-A Survey", Annals of Software Engineering, Vol. 10, pp. 177-205, 2000.
2. Pressman R. S., "Software Engineering-A Practitioner's Approach"5th Edition, McGraw-Hill Series in Computer Science, ISBN: 0-07-365578-3, 2001.
3. Al-Hajri M. A., Ghani A. A. A., Sulaiman M. N., and Selamat M. H., "Modification of Standard Function Point Complexity Weights System", The Journal of Systems and Software, Elsevier, Vol. 74, pp. 195-206, 2005.
4. Sadiq M., Afrin A., Tahir H., Nazneen S., "Fuzzy_FP: A Fuzzy Oriented Approach for the Computation of Function Point", 4th IEEE International Conference on Computing for Sustainable Development", pp. 1300-1303, 2017.
5. Abdullah N. A. S., Abdullah R., Selamat M. H., and Jaafar A, "Software Security Characteristics for Function Point Analysis", 16th IEEE International Conference on Industrial Engineering and Engineering Management, Hongkong, pp. 394-397, 2009.
6. Ahn Y, Suh J, Kim S, and Kim H, "The Software Maintenance Project Effort Estimation Model Based on Function Points", Journal of Software Maintenance Evolution, Vol. 15, No. 2, pp. 71-85, 2003.
7. Pacheco C. and Garcia I., "A systematic literature review of stakeholder identification methods in requirements elicitation" , The Journal of Systems and Software, Vol. 85, pp. 2171-2181, 2012.
8. Burnay C., Jureta I. J., and Faulkner S., "What stakeholders will or will not say: A theoretical and empirical study of topic importance in requirements engineering elicitation interviews", Information System, Vol. 46, pp. 61-81, 2014.
9. Coughlan J., Lyeett M., Macredie R. D., "Communications issues in requirements elicitation: A content analysis of stakeholders expressions", Information and Software Technology, Vol. 45, pp. 525-537, 2003.
10. Mohammad C. W., Shahid M., and Husain S.Z., "FAGOSRA: Fuzzy Attributed goal oriented software requirements analysis method", 9th IEEE International Conference on Contemporary Computing, pp. 384-389, 2016.
11. Matson J. E., Barrett B. E. and Mellichamp J. M., "Software Development Cost Estimation using Function Points", IEEE Transactions on Software Engineering, Vol. 20, No. 4, pp. 275-287, 1994.
12. Kemerer C. F. and Porter B. S., "Improving the Reliability of Function Points Measurement: An Empirical Study", IEEE Transactions on Software Engineering, Vol. 18, No. 11, pp. 1011-1024, 1992.
13. Jorgensen M. and Shepperd, "A Systematic Literature Review of Software Development Cost Estimation Studies", IEEE Transactions on Software Engineering, Vol. 33, No. 1, pp. 33-53, 2007.
14. Heemstra F. J. and Kusters R. J., "Function Point Analysis: Evaluation of Software Cost Estimation Model", European Journal of Information Systems, Vol. 1, No. 4, pp. 229-237, 1991.
15. Symons C. R., "Function Point Analysis: Difficulties and Improvements", IEEE Transactions on Software Engineering, Vol. 14, No. 1, 1988.
16. Fetcke T., Absan A., and Ngugen T. H., "Mapping the OO-Jacobson Approach into Function Point Analysis", Proc. of IEEE TOOLS-23'97, pp. 1-11, 1997.
17. Junior M. F., Fantinato M., and Sun V., "Improvements to Function Point Analysis Method: A Systematic Literature Review", IEEE Transactions on Software Engineering, Vol. 62, No. 4, pp. 495-506, 2015.
18. Mohammad C. W., Shahid M., and Husain S.Z., "Fuzzy attributed goal oriented software requirements analysis with multiple stakeholders", International Journal of Information Technology, Springer, 2018.
19. Meth H., Brhel M., Maedche A., "The state of the art in automated requirements elicitation", Information and Software Technology, Vol. 55, pp. 1695-1709, 2013.
20. Carrizo D., Dieste O., and Juristo N., "Systematizing requirements elicitation techniques selection", Information and Software Technology, Vol. 56, pp. 644-669, 2014.
21. Hickey A. M., and Davis A. M., "Requirements Elicitation and elicitation technique selection: A model for two knowledge intensive software development processes", 36th Hawii International Conference on System Sciences, pp. 1-10.
22. Lloyd W. J., Rosson M. B., and Arthur J. D., "Effectiveness of elicitation techniques in distributed requirements engineering", IEEE Joint International Conference on Requirements Engineering, pp. 1-8, 2002.
23. Mayela D. H., and Francisca L. A. M., "Goal oriented techniques and methods: goal refinement and levels of abstraction", 39th IEEE Latin American Conference, pp. 1-12.

24. Arif M. and Sarwar S., "Identification of requirements using goal oriented requirements elicitation process", Vol. 120, No. 15, International Journal of Computer Applications, 2015.
25. Kaiya H., Horai H., and Saeki, "AGORA: Attributed goal oriented requirements analysis method", IEEE International Conference on Requirements Engineering, pp. 13-22, 2002.
26. Shibaoka M., Kaiya H., and Saeki M., "GOORE: Goal oriented and ontology driven requirements elicitation method", International Conference on Advances in Conceptual Modeling-Foundations and Applications, LNCS, Springer, pp. 225-234, 2007.
27. Czarnacka-Chrobot B, "What Is the Cost of One IFPUG Method Function Point- Case Study", Technical Report.
28. Abdullah N. A. S., Abdullah R., Selamat M. H., Jaafar A., "Software Security Characteristics for function point analysis", IEEE International Conference on Industrial Engineering and Engineering Management, 2010.