An Approach for Software Risk Estimation Using Fuzzy Function Point

Vineeta
Computer Science & Engineering Dept.
Modern Institute of Technology & Research Centre
Alwar, Rajasthan, India

Pranay Kumar Mishra
Department of Computer Application
Laxmi Devi Institute of Engineering& Technology
Alwar, Rajasthan, India

***Abstract*—This paper describes the estimation of risk of the software using fuzzy function point. Function point analysis is amongst commonly used techniques to estimate the software size. The function point approach is used in order to develop the architecture of the esrcTool [1] that estimate cost and risk of the software. In this work fuzzy function point approach is used for risk estimation. Function points calculatedusing Fuzzy Function Point Analysis (FFPA), can be used to determine cost of development.**

Keywords- FFPA, FPA, risk estimation.

I. INTRODUCTION

Software project management is an umbrella activity within software engineering. It begins before any technical activity is initiated and continues throughout the definition, development, and support of computer software. The project management activity encompasses measurement and metrics, estimation and scheduling, risk analysis, tracking, and control [12]. After determination of scope of a project, the manager must specifythe development period and the cost of the product. Such activities are crucialfor impractical estimation may lead to the failure of the project.

There have been various methods devised to compute the size of a software product like COCOMO (Boehm, 1981), PUTNAM-SLIM (Putnam, 1978) and FPA (Albrecht, 1979). FPA is most common approach used for size estimation of software. FPA is derived from measures of the information domain and a subjective assessment of problem complexity.FPA uses a quick mode of classifying its functions. For example, an external input that references two files and five data items is classified as "average"getting four points. Another external input that references two files and fifteen data items is also classified as "average", getting four points. In another case, when the function references two files and sixteen data items, this then is classified as "high", getting six points. Two serious problems are immediately evident in such a system of classification: (i) functions of different sizes receive the same point values, and (ii) similar functions are abruptly classified into different groups [2].

Fuzzy logic offers a solution to these drawbacks. Fuzzy logic may be considered as the multi-valued logic. It can describe the real-world expressions better than Boolean logic. In [8] the authors have used fuzzy logic for size estimation. Another reason for the use of fuzzy logic in this field, as given in [9], where it is used for effort estimation, is that these metrics don't have any exact mathematical model anyway and only produce estimation.

Taking into the accounts of benefits of fuzzy logic in software estimation, in this paper fuzzy function point is used for size estimation and then the risk is estimated.

The paper is organized as follows. In section 2 we present the background and related work. In section 3, we have explained the function point approach. Fuzzy function point approach is summarised out in section 4. Estimation of risk is described in section 5. Finally we conclude the paper in section 6.

II. BACKGROUND AND RELATED WORK

In the field of software engineering a lot of models/ tools have been developed according to need and requirements. In [3]authors have developed a SRAEM (Software Risk Assessment and Estimation Model). Using this model it is possible to estimate and prioritize the risk. This model uses two approaches. One approach is based on probability of risk and another is based on Mission Critical Requirements Stability Risk Metrics (MCRSRM). This software metrics is used when there are changes in requirements such as addition, subtraction, or deletion. In [4] the authors have developed a Software Estimation Tool Based on Software Engineering Metrics Model. In [6] the authors have developed a model and prototype tool to manage software risks. SoftRisk prototype is a tool prototype to manage software development risks. One another model Software Engineering Risk Model (SERIM) focuses on three risk elements: (i) technical risk, (ii) cost risk, and (iii) schedule risk. This model does not take into account of the software complexity issues, which plays an important role in determining the risk for the software projects. It also does not account for issues related to requirements. In [1] sources of estimate uncertainty, i.e. measurement error, model error and assumption errors are included. Also it uses Function

point approach to calculate the cost and risk included in the software. [2] introduces the fuzzy function point approach for software size estimation.

III. FUNCTION POINT APPROACH

Function Point is a measure of software size that uses logical functional terms. Function points are derived using an empirical relationship based on countable measures of software's information domain and assessments of software complexity. The function point calculation is done in three steps: (I) determine the unadjusted function point (UFP), (II) determine Value adjustment factor (VAF) and (III) determine Adjusted function points (AFP). Unadjusted Function points comprises of data and transactional functions. Data functions are Internal Logical File (ILF) and External Interface File (EIF) while transactional functions are External Inquiries (EQ), External Output (EO) and External Input (EI). Once identified, each function must be classified according to its relative functional complexity as low, average, or high. The data functions' relative functional complexity is based on the number of data element types (DETs) and the number of record element types (RETs). A RET is defined as a subset of logically related data within an ILF or an EIF. A DET is a singular, non-repeated field recognized by the user as having its own meaning. The transactional functions (EI, EO, EQ) are classified according to the number of file types referenced (FTRs) and the number of DETs. The number of FTRs is the sum of the number of ILFs and the number of EIFs updated or queried during an elementary process.The second stage, calculating the value adjustment factor (VAF), is an earmark of the general functionality provided to the user. The third and last stage is the final calculation of the function points. [10,11] gives the detailed calculations of the FP.

.

IV. FUZZY FUNCTION POINT APPROACH

The fuzzy function point approach, as proposed in [2] aims to provide a more precise approach to the function points counting process while at the same time guaranteeing the validity of the final calculation of traditional function points.Fuzzy Function Point Analysis consists of four stages: The first stage involves generating trapezoid fuzzy number N (a, m, n, b) for each linguistic term T_i *(low, average, high).* The second stage introduces a new linguistic term *very high* in order to tackle situations explained in section 1. In the third stage, p_i function pointsare directly associated with the fuzzy number of the linguistic term, where $\mu_{\tilde{N}}(x) = 1$. The value of the function points for the new linguistic term of very high complexity is calculated by the extrapolation of the values already defined for the terms low, medium, and high of each function. The approximant function of Newton's formula is used for extrapolation. The final stage includes defuzzificationprocess to compute the function points [2].

TABLE I

FTR	DETs			
	1-4	**5-15**	**16-26**	**27 or more**
0-1	Low	Low	Average	High
2	Low	Average	High	Very High
3 or more	Average	High	High	Very High

Table I is the extended EI complexity matrix. Value to the linguistic term *very high* for EI is 27, for EO and EQ is 34 and for EIF and ILF is 82. The value of the function points for the term *very high*of each function data as well as transactional is obtained by substituting the progressive differences in the approximant function [2] and shown in table II.

TABLE II

MP	COUNT	LOW	AVERAGE	HIGH	VERY HIGH
EI	**X**	**3**	**4**	**6**	**9**
EO	**X**	**4**	**5**	**7**	**10**
EQ	**X**	**3**	**4**	**6**	**9**
ILF	**X**	**7**	**10**	**15**	**22**
EIF	**X**	**5**	**7**	**10**	**14**
TUFFP	**TOTAL**				

Defuzzification yields the Fuzzy Function Point which can be calculated as:

$$p_d = \mu_{\tilde{N}}(x) * \bar{p_i} + \mu_{\tilde{N}}(x) * p_{i+1} \quad (1)$$

Figure 1 illustrates an example of computing fuzzy function point of an EI with 2 FTR and 10 DETs.Membership function values are computed as$\mu_{\tilde{N}}(10)$=(16-10)/(16-5)=0.55 &$\mu_{\tilde{N}}(10)$=1-0.55=0.45 and putting the values in the above equation we get p_d= (0.55*4) + (0.45*6) =4.90 function points.

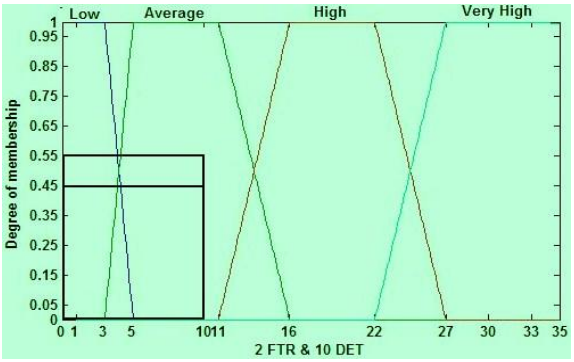

Figure 1

Figure 2, figure3 and figure 4 shows the membership function to the fuzzy number defined for EI for 0-1 FTR, 2 FTR and 3+ FTR respectively.

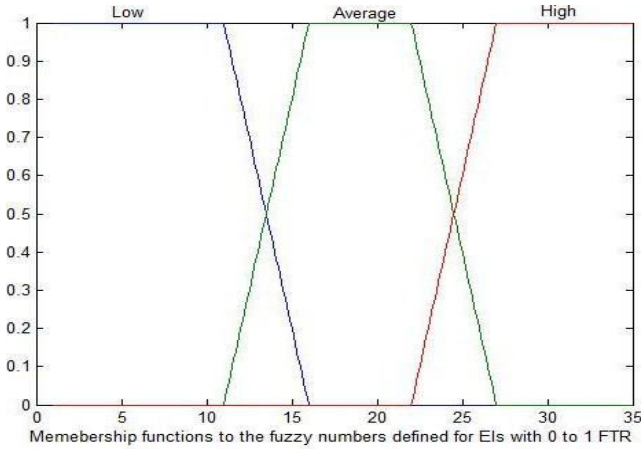

Figure 2

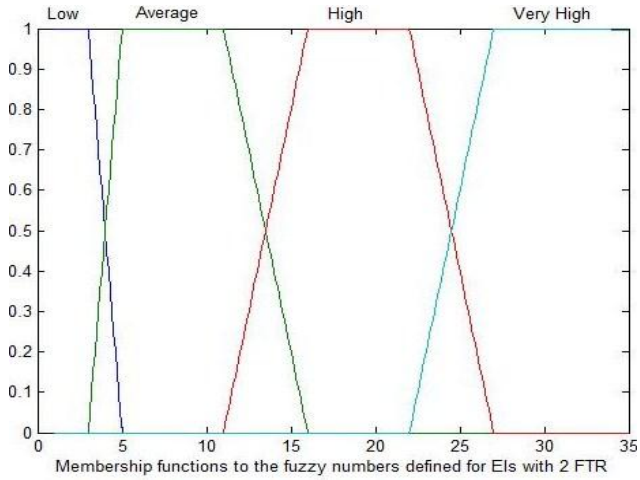

Figure 3

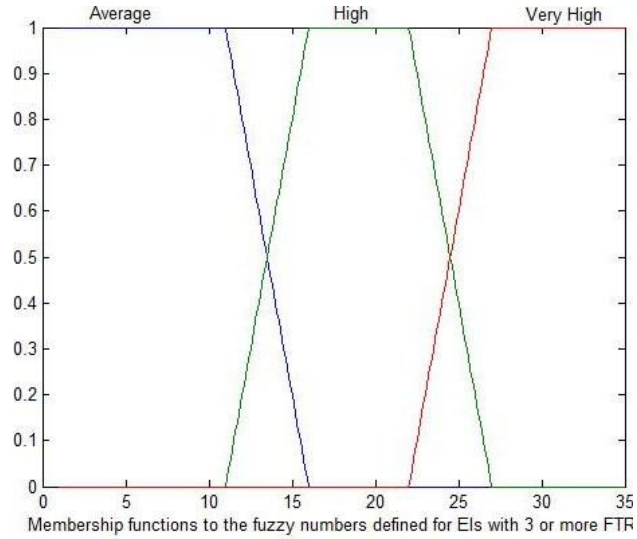

Figure 4

Summation of all fuzzy function points yields the Total Unadjusted Fuzzy Function Points (TUFFP). The Total Adjustment Factor (TAF)is derived from the sum of the degree of influence (DI) of the 14 general system characteristics (GSCs). The DI of each one of these characteristics ranges from 0 (no influence) to 5 (strong influence).The general characteristics of a system are : (i) datacommunications; (ii) distributed data processing; (iii)performance; (iv) heavily used configuration; (v)transaction rate;(vi) online data entry; (vii) end userefficiency (viii) online update(ix) complex processing;(x) reusability; (xi) installation ease; (xii) operationalease; (xiii) multiple sites; (xiv) facilitate change.

The TAF is calculated by the following equation:

$$TAF=0.65+0.01*\Sigma DI \quad (2)$$

whereΣDI is sum of the general system characteristics. With the help of the following equation we can get the total points of an application:

$$AFFP = TUFFP * TAF \quad (3)$$

where AFFP = Adjusted Fuzzy Function Points

V. RISK ESTIMATION

After calculating the function points for the application, now the risk exposure is estimated. For calculation of risk we used, assumption error as the source of uncertainty.

Assumption Error- This error occurs when we make incorrect assumptions about input parameters. For example, if we have correctly identified all the customer requirements then the product size is supposing 1300 fuzzy function points. Also Assuming 0.5 person-days per function point,we believe that there is a 0.3 probability that the requirement complexity has been underestimated and, thus, we estimate another 100 fuzzy function point. At this point the concept of risk exposure is used to calculate the effective current cost of a risk [3]. The risk is estimated using the following equation:

$$RE=P(risk) * Total\ loss \quad (4)$$

Where RE =Risk Exposure, P(risk)=Probability of risk occurring and Total Loss=E2-E1 where E1 is the effort if the original assumption is true and E2 is the effort if thealternative assumption is true. Suppose E1=400 person days and E2= (1300+100)*0.5=700 person days, then risk exposure= (700-400)*0.3= 90 person days.

VI. CONCLUSION

This paper proposes the fuzzy function point approach to estimate the software risk. The membership functions have been generated using MATLAB. The fuzzy trapezoidal numbers for each linguistic term and function points for each function type and complexity were computed analogous to the Fuzzy Function Point approach [2]. It is proposed that the use of Fuzzy Function Point Approach would yield better results than Function Point Approach.

ACKNOWLEDGMENT

We would like to thank Mr. KailashMaheshwari, Dean (Academics), Modern Institute of Technology & Research Centre,Alwar, India and Dr. Rajesh Bhardwaj, Director, Laxmi Devi Institute of Engineering & Technology, Alwar, India, for their valuable support and guidance.

REFERENCES

[1] Mohd. Sadiq, Abdul Rahman Shabbir Ahmad , Mohammad Asim , Javed Ahmad ," esrcTool: ATool to Estimate the Software Risk and Cost ", 2010 IEEE International Conference on Computer Research and Development ," Kuala Lumpur, Malaysia".

[2] Osias De Souza Lima Junior, Pedro PorfirioMuniaz Farias, Arnaldo Dias Belchior, " A Fuzzy Model for Function Point Analysis to Development and Enhancement Project Assessments", CLEI Electronic Journal 5(2), 1999.

[3] Daya Gupta, Mohd sadiq, "Software Risk assessment and estimation Model", International Conference on computer science and Information Technology, IEEE Computer society, Singapore, 2008. pp 963-967

[4] Daya Gupta, SatyaPal Jee kaushal, Mohd. Sadiq, "Software Estimation tool based on Software Engineering Metrics Model", IEEE International Conference on Management of Innovation & Technology, Bangkok, Thailand.pp 623-628

[5] Joseph S. Sherif, "Metrics for software Risk Management", ISMN#0-7803-3274-1, pp 507-513

[6] Ayad Ali Keshlaf and Khairuddin Hashim,"A Model and Protype Tool to Manage Software Risks", IEEE 2000, pp 297-305

[7] M. Francesco, A. Mehmet, , "Improving object-oriented methods by using fuzzy logic", ACM SoftwareReview,2001.

[8] A. F. Sheta, D. Rine, A. Ayesh, "Development of software effort and schedule estimation models using Soft Computing Techniques", IEEE Congress on EvolutionaryComputation, pp. 1283-1289, 2008.

[9] C.L.Martin, J.L. Pasquier, , C.M. Yanez, , A.T. Gutierrez, "Software Development Effort Estimation Using Fuzzy Logic: A Case Study", Proceedings of the Sixth MexicanInternational Conference on Computer Science, pp.113-120, September, 2005.

[10] International Function Point User Group(IFPUG), Function Point Counting PracticesManual, Release 4.0, IFPUG, Westerville,Ohio, April 1990.

[11] Zuse H. 1991., Software Metrics-Methods toInvestigate and Evaluate SoftwareComplexity Measures, Proc. Second AnnualOregon Workshop on Software Metrics,Portland.\

[12] Roger S. Pressman, "Software Engineering A Practitioner's Approach", pp 645-646