Modelling Rate of Software Aggregative Risk Using Fuzzy Logic Technique

Dr Harish Mittal
Director, Sat Priya Group of Institutions,
Rohtak, India
mittalberi[at]yahoo.co.in

Kulbhushan Bansal
M. Tech Scholar,
Department of Computer Science
NITTTR, Chandigarh,India
Kul_bansal[at]yahoo.co.in

Dr. Maitreyee Dutta
Professor and Head, Dept. of CSE
NITTTR, Chandigarh, India
d_maitreyee[at]yahoo.co.in

Abstract— despite numerous available methods of software risk management, software projects have a high rate of failure. Software risk analysis begins with the introduction of software proposal. Risk factors are added at each consecutive step. In this work we will introduce a model using fuzzy logic to efficiently evaluate the software aggregative risk based on some of these factors (risk items) in order to minimize risks in software development and enhance quality and reliability of software.
Keywords- Risk Analysis, Software aggregative risk, Risk aggregation, **Risk design, Risk management cycle.**

1. Introduction

In order to effectively manage a software project, we must learn to identify, analyze, and control software risks. Identifying and dealing with risks early lessens long term costs and helps prevent software disasters[23]. Major software risks identified may be categorized as [21]:

- **Technical risks**
- **Management risks**
- **Financial risks**
- **Legal risks**
- **Personnel risks**

Boehm defines four major reasons for implementing software risk management [3]:

- Avoiding software project disasters.
- Avoiding rework.
- Avoiding overkill.
- Stimulating a win-win software solution.

During the risk analysis, each risk is assessed to determine the probability that the risk will result in a loss and the size of that loss if the risk turns into a problem [3].

Lee et al. [12] uses 11 linguistic values for ranking the grades of risk of the risk items and uses them for ranking the grades of importance of the risk items. They evaluated the fuzzified rate of each individual risk item by the multiplication of these two fuzzy sets, the defuzzified result was obtained by centroid method.

Lee's model for evaluating rate of aggregative risk takes a large amount of time to form the fuzzy assessment and especially when the number of attributes is large, it is a tedious work. So there is a need of a more efficient and less complex method.

II. Literary Review

Many researchers have focused their research on risk identification, risk analysis, risk priority, and risk management [2,4,6,7,8,10,12,15,17]. Boehm [3] pointed out four major reasons for implementing software risk management. He also pointed out 10 risk items.

Lee et al. [12] uses 11 linguistic values for ranking the grades of risk of the risk items and uses 11 linguistic values for ranking the grades of importance of the risk items and represented them by triangular fuzzy numbers, Table I and Table II. Each individual risk item is represented using two fuzzy sets with TFNs: grade of risk and grade of importance.

S. No.	Grades of risk	Triangular Fuzzy Number
1	Definitely low	(0,0,0.1)
2	Extra low	(0,0.1,0.2)
3	Very low	(0.1,0.2,0.3)
4	Low	(0.2,0.3,0.4)
5	Slightly low	(0.3.,0.4,0.5)
6	Middle	(0.4,0.5,0.6)
7	Slightly high	(0.5,0.6,0.7)
8	High	(0.6,0.7,0.8)
9	Very high	(0.7,0.8,0.9)
10	Extra high	(0.8,0.9,1.0)
11	Definitely high	(0.9,1.0,1.0)

TABLE I. GRADES OF RISK

S. No.	Grades of importance	Triangular Fuzzy Number
1	Definitely unimportant	(0,0,0.1)
2	Extra unimportant	(0,0.1,0.2)
3	Very unimportant	(0.1,0.2,0.3)
4	Unimportant	(0.2,0.3,0.4)
5	Slightly unimportant	(0.3.,0.4,0.5)
6	Middle	(0.4,0.5,0.6)

7	Slightly important	(0.5,0.6,0.7)
8	Important	(0.6,0.7,0.8)
9	Very important	(0.7,0.8,0.9)
10	Extra important	(0.8,0.9,1.0)
11	Definitely important	(0.9,1.0,1.0)

TABLE II. GRADES OF IMPORTANCE

With the fast growing demand of software in almost every field there is a great scope of the study to evolve more and more efficient methods to minimise risks in the development of software. This will help to reduce the cost of the software development as well.

II. PROPOSED MODEL

In recent years, many researchers focused on the research of risk identification, risk analysis, risk priority, and risk management. Risk aggregation refers integrating multiple types/sources of risk into a single metric. Normally it is helpful to aggregate risks based on their interrelationships, and to develop options at an aggregate level.

Because most decision makers or project managers usually evaluate the rate of risk factors by linguistic values, fuzzy set theory provides us a useful tool to deal with the ambiguity involved in the data evaluation process [23].

This work will consist of various stages, starting with problem identification followed by research procedure and data collection, and will end with analysis of data. The risk items will first be fuzzified and then be processed as proposed in the model and a fuzzified result will be obtained which will then be defuzzified as per proposed model to provide aggregative risks values. Figure 1 provides an overview of methodology for modelling aggregative risk.

Project managers first identify sub-metrics for each of the five metrics, technical risk, management risk, financial risk, legal risk, and personnel risk represented by M_1, M_2, M_3, M_4 and M_5 respectively. Each metric can further be divided into r sub-metrics [23]. Data is collected from different sources based on a comprehensive literature review and interviewed with practitioners.

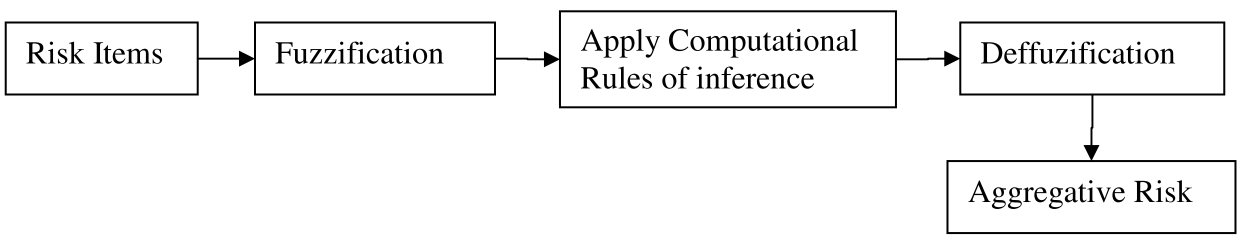

Figure 1: Methodology of evaluating Aggregative risks in software development

Potential risks associated with software development

Technical Risks (M_1)

M_{11} Hacker gaining unauthorized access
M_{12} Absence of firewall
M_{13} Lack of using cryptography and Poor ''key'' management
M_{14} Wrong project size estimation
M_{15} Malicious code attacks
M_{16} Disclosure of sensitive information
M_{17} Language barrier
M_{18} Project complexity
M_{19} Loss of control over information technology
M_{110} Continuous change of system requirements
M_{111} Technological newness
M_{112} Software or hardware problem-caused system failure
M_{113} Poor design, code or maintenance procedure
M_{114} Wrong functions and properties development
M_{115} Wrong user interface development

Financial Risks (M_2)

M_{21} Loss of audit trail
M_{22} Wrong schedule estimation
M_{23} Project behind schedule
M_{24} Project over budget
M_{25} Inadequate cash flow
M_{26} Hidden cost

Personnel risks (M_3)

M_{31} Personnel shortfalls
M_{32} Lack of expertise and experience in e-commerce
M_{33} Loss of key person

Managerial Risks (M_4)

M_{41} Poor project planning
M_{42} Unclear project objectives and scope
M_{43} Business process redesign
M_{44} Organizational restructuring
M_{45} Lack of trust between your organization and merchant or customer
M_{46} Lack of top management support

Legal Risks (M_5)

M_{51} Lack of international legal standards
M_{52} New laws, regulations, and judicial decisions constantly change the online legal landscape
M_{53} Uncertain legal jurisdiction
M_{54} Incompletion of contract terms

Each sub-metric is assigned a weight, w_{ij}, where i=1,2,2...5 and j=1,2,3...r. .

- The sum of weights of all the sub-metrics of any metric is taken as 1.
- The total weight of a metric is divided equally in all its sub-metrics.
- Each sub-metric is then given two grades, grade of risk (Gr) and grade of importance (Gi), as per rules given in Table III and IV.
- Input value of risk and importance are taken as mid values of the Triangular Fuzzy numbers corresponding to Gr and Gi.
- Input values of risk and importance are then multiplied by the weights of the corresponding sub-metric to get pij, input value of sub-metric.
- The input value Pi, of a metric Mi, is taken as sum of input values of pij of sub-metrics Mij.
- Input values are then fuzzified and defuzzified as per rules.

Complexity Of risk	**TFN**	**Grade of risk(Gr)**
Very low	(0,0.1,0.2)	1
Low	(0.2,0.3,0.4)	2
Medium	(0.4,0.5,0.6)	3
High	(0.6,0.7,0.8)	4
Very high	(0.8,0.9,1.0)	5

TABLE III. COMPLEXITY MATRIX OF RISK FOR SUB-METRICS

Complexity of importance	**TFN**	**Grade of importance (G_i)**
Very low	(0,0.1,0.2)	1
Low	(0.2,0.3,0.4)	2
Medium	(0.4,0.5,0.6)	3
High	(0.6,0.7,0.8)	4
Very high	(0.8,0.9,1.0)	5

TABLE IV. COMPLEXITY MATRIX OF IMPORTANCE FOR SUB-METRICS

Metric	**Sub metric (M_{ij})**	**Weight** (w_{ij})	**Grade of Risk (G_r)**	**Grade of Importance (G_i)**	**Input value of risk of M_{ij}**	**Input value of importance of M_{ij}**	**Input value of M_{ij} (P_{ij}) (w_{ij}*M_{ij} risk, w_{ij} *M_{ij} imp)**	**Input value of Metric M_i (P_i)**
M_1	M_{11}	$1/r_1$	4	5	0.7	0.9	0.047 0.06	
$r_{1=15}$	M_{12}	$1/r_1$	4	5	.7	.9	0.047 0.06	
	M_{13}	$1/r_1$	4	5	.7	.9	0.047 0.06	1.605
		$1/r_1$	4	5	.7	.9	0.047 0.06	
	M_{1r1}	$1/r_1$	4	5	.7	.9	0.047 0.06	
M_2	M_{21}	$1/r_2$	3	5	0.5	0.9	0.083 0.15	
$r_{2=6}$	M_{22}	$1/r_2$	3	5	.5	.9	0.083 0.15	1.398
	..	$1/r_2$	3	5	.5	.9	0.083 0.15	
	M_{2r2}	$1/r_2$	3	5	.5	.9	0.083 0.15	
M_3	M_{31}	$1/r_3$	2	4	.3	0.7	0.1 0.23	
$r_{3=3}$	M_{32}	$1/r_3$	2	4	.3	0.7	0.1 0.23	0.99
	M_{3r3}	$1/r_3$	2	4	.3	0.7	0.1 0.23	
M_4	M_{11}	$1/r_4$	2	3	.3	0.5	0.05 0.083	
$r_{4=6}$	M_{12}	$1/r_4$	2	3	.3	.5	0.05 0.083	0.8
	..	$1/r_4$	2	3	.3	.5	0.05 0.083	
	M_{1r4}	$1/r_4$	2	3	.3	.5	0.05 0.083	

M_5 $r_{5=4}$	M_{11}	$1/r_5$	1	3	.1	.5	0.025 0.125	0.6
	M_{12}	$1/r_5$	1	3	.1	.5	0.025 0.125	
	..	$1/r_5$	1	3	.1	.5	0.025 0.125	
	M_{1r5}	$1/r_5$	1	3	.1	.5	0.025 0.125	

TABLE V. SUB-METRICS OF METRICS (ILLUSTRATIVE EXAMPLE)

M_i	Complexity	Co-efficient R_i
0-0.4	Low	w_1
0.4-0.8	Medium	w_2
0.8 or more	High	w_3

TABLE VI. COMPLEXITY MATRIX OF RISK FOR METRICS.

Fuzzification-

The complexity attributes are taken as Triangular Fuzzy Numbers (TFN). Membership functions are evaluated using complexity and co-efficient matrices. Fuzzy pictorial representation of a TFN is shown in figure 2

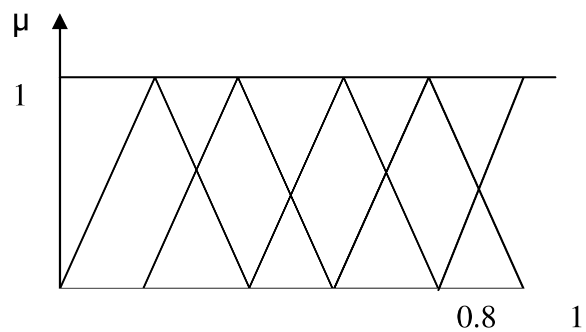

Figure 2: Triangular Fuzzy Number

The membership function μ(x) is defined as:

$$\mu(x) = \begin{cases} 0, x \le \alpha \\ \dfrac{x-\alpha}{m-\alpha}, \alpha \le x \le m \\ \dfrac{\beta - x}{\beta - m}, m \le x \le \beta \\ 0, x \ge \beta \end{cases} \quad \text{---(1)}$$

For Example

$$\mu\ (0.99) = \frac{0.99-0.8}{1.0-0.8} = \frac{0.19}{.2} = 0.95$$

Defuzzification-

Defuzzification approach used in this work represents a substantive departure from the conventional fuzzy logic techniques [24,25]. The criteria for defuzzification which are not directly related to theoretical concepts and fundamentals are formulated.

We define

$$Ri = \begin{cases} \mu * w_1,\ 0 \le Pi \le 0.4 \\ \mu * w_1 + \bar{\mu} * w_2, 0.4 \le Pi \le 0.8 \\ \mu * w_2 + \bar{\mu} * w_3, 0.8 < Pi < 1.2 \\ \mu * w_3,\ Pi \ge 1.2 \end{cases} \quad \text{---(2)}$$

Here μ is grade of membership which lies between 0 and 1

$\bar{\mu} = 1 - \mu$

Pi is **Input value of Metric M_i**

w_1, w_2, w_3 are weights, For current example application we consider w_1=0.1, w_2= .15, w_3=.2

For Example

$R_3 = \mu * w_2 + \bar{\mu} * w_3, 0.8 < Pi < 1.2$

$= 0.95 * 0.15 + 0.05 * 0.2 = 0.1525$

Aggregate Risk = $\sum_i^5 R_i$ …(3)

Aggregative risk rate $= \sum_1^5 \frac{R_i * W_i}{\sum_1^5 R_i}$ …(4)

Here, W_i is weight of each metric depending on grade and importance of Risk

Considering Numerical Example in Table V, we get

Using equation 3 & 4

Aggregate Risk = 0.2 + 0.198 + 0.1525 + 0.15 + 0.1
= 0.8005

Rate of Aggregative risk = 0.2127

III. CONCLUSION AND FUTURE WORK

This paper describes modified aggregative risk assessment model which may be utilised to estimate and subsequently minimise risks in software development and enhance quality and reliability of software. The proposed assessment method in this paper uses the fuzzy numbers for the risk items to reduce the degree of subjectivity of the evaluator. The present model is an easy and effective technique to evaluate the aggregative risk of a software project and rate of Aggregative Risk. Future work involves case study on real projects data and involvements of groups and practitioners and experts for Aggregative Risk Assessment.

REFERENCES

[1]. Alex Down, Michael Coleman, Peter Absolon, Risk Management for Software Projects, London, McGraw-Hill Book Company, 1994.

[2]. Boehm, B. W. 1988. A spiral model of software development and enhancement. IEEE Computer 21s 5: 61- 72.

[3]. Barry W. Boehm, Tutorial: Software Risk Management, Les Alamitos, CA, IEEE Computer Society, 1989.

[4]. Boehm, B. W. 1991. Software risk management: Principles and practices. IEEE Software 8s 1. : 32- 41.

[5]. Chen, S. M. 1994. Using fuzzy reasoning techniques for fault diagnosis of the J-85 jet engines. Proceedings of the 3rd National Conference on Science and Technology of Nation al Defence, Taoyuan, Taiwan, R.O .C., pp. 29-34.

[6]. Chen, S. M., and S. Y. Lin. 1995. A new method for fuzzy risk analysis. Proceedings of 1995 Artificial Intelligence Workshop, Taipei, Taiwan, R.O.C. pp. 245-250.

[7]. Chen, S. M. 1996. New methods for subjective mental workload assessment and fuzzy risk analysis. Cybernetics and Systems: An International Journal 27s 5.: 449-472.

[8]. Chen, S. M. 1997. A new method to evaluate the rate of the aggregative risk in software development. Proceedings of the 8th International Conference on Information Management, Taipei, Taiwan, R.O.C., pp. 791- 798.

[9]. Conger, S. A. 1994. The New Software Engineering. CA: Wadsworth Publishing. Gleason, J. M. 1991. Fuzzy set computational process in risk analysis. IEEE Trans. Enrgr. Manage. 38s 2.: 177-178.

[10]. Kangari, R., and L. S. Riggs. 1989. Construction risk assessment by linguistics. IEEE Trans. Enrgr. Manage. 36s 2.: 126-131.

[11]. Kaufmann, A., and M. M. Gupta. 1988. Fuzzy Mathematical Models in Engineering and Management Science. Amsterdam: North-Holland. Kaufmann, A., and M. M. Gupta. 1991. Introduction to Fuzzy Arithmetic Theory and Application s. New York: Van Nostrand Reinhold.

[12]. Lee, H. M. 1996. Applying fuzzy set theory to evaluate the rate of aggregative risk in software development. Fuzzy Sets and Systems 79s 3.: 323-336.

[13]. Martyn Ould, Strategies for Software Engineering: The Management of Risk and Quality, Chichester, England, John Wiley & Sons, 1990.

[14]. Marvin J. Carr, Suresh L. Konda, Ira Monarch, F. Carol Ulrich, Clay F. Walker, Taxonomy –Based Risk Identification, CMU/SEI-93-TR-006, Pittsburgh, PA, Software Engineering Institute, 1993.

[15]. Schmucker, K. J. 1984. Fuzzy Sets, Natural Language Computations, and Risk Analysis. Rockville, MD: Computer Science Press.

[16]. Tom Gilb, Principles of Software Engineering Management, Wokingham, England: Addison-Wesley, 1988.

[17]. Yu, P. 1993. Modelling Cost and Risk for Software Information System. Master Work. National Defence College, Taiwan, R.O .C. Zadeh, L. A. 1965. Fuzzy sets. Inform. Control 8:338-353.

[18]. Zimmermann, H.-J. 1991. Fuzzy Set Theory and Its Applications. Boston: Kluwer Academic Publishers.

[19]. L.A. Zadeh, Fuzzy sets, Information and Control 8 (1965) 338–353

[20]. Lin, Lily, Huey-Ming Lee, Shu-Yen Lee, and Tsung-Yen Lee. "Fuzzy evaluation for the rate of aggregative risk in software development." In Proceedings of the 6th WSEAS International Conference on Applied Computer Science, Hangzhou, China, pp. 43-47. 2007.

[21]. Team, Trinity. "Software Project Management Plan." Evolution (2012).

[22]. Lee, Huey-Ming, and Lily Lin. "Fuzzy evaluating the rate of aggregative risk in software development." Machine Learning and Cybernetics (ICMLC), 2010 International Conference on. Vol. 5. IEEE, 2010.

[23]. Bansal, Kulbhushan, and Harish Mittal. "Analysis and Evaluation of Software Aggregative Risk Using Soft Computing Techniques." Advanced Computing & Communication Technologies (ACCT), 2014 Fourth International Conference on. IEEE, 2014.

[24]. Lima, O.S.J., Farias P.P.M. and Belchio A.D., “A Fuzzy Model for Function Point Analysis to Development and Enhancement Project Assessments”, CLEI Electronic Journal, Volume 5: Number 2, January 2002.

[25]. Leekwijck, W. V., Kerre, E. E., Defuzzification: criteria and classification, Fuzzy sets and systems, 108, 2, 159-178, 1999.

[26]. Gupta, Deepak, Harish Kumar Mittal, and Vinay Goyal. "Comparative Study of Soft Computing Techniques for Software Quality." International Journal of Software Engineering Research and Practices 1.1 (2011): 33-37.

[27]. Gupta, Deepak, Vinay Kr Goyal, and Harish Mittal. "Analysis of clustering techniques for software quality prediction." 2012 Second International Conference on Advanced Computing & Communication Technologies. IEEE, 2012.

[28]. Gupta, Deepak, Vinay Kr Goyal, and Harish Mittal. "Estimating of Software Quality with Clustering Techniques." Advanced Computing and Communication Technologies (ACCT), 2013 Third International Conference on. IEEE, 2013.