RISK PREVENTION AND DEDUCTION IN SOFTWARE DEVELOPMENT USING FUZZY MEMBERSHIP FUNCTION

***Faizul Azli Mohd-Rahim*[1], *Chen Wang* [2], *Halim Boussabaine* [3], *Hamzah Abdul-Rahman* [4],**

[1,2] Department of Quantity Surveying, Faculty of Built Environment, University of Malaya, 50603 Kuala Lumpur, Malaysia

[3] School of Architecture, University of Liverpool, Liverpool L69 3BX, United Kingdom

[4] Vice Chancellor's Office, International University of Malaya-Wales, 50480, Kuala Lumpur, Malaysia

Corresponding Author: Assoc. Prof. Dr. Chen Wang

Corresponding Email: [2]derekisleon@um.edu.my
Tel : +603-7967 6860

Abstract

Evidence indicates that risks in IT projects which are not effectively managed and lack of identification and management during the life cycle of a project can contribute to their failures. Traditional risk assessment methods usually model risks with objective probabilities based on the expected frequency of repeatable events. Meanwhile, managers prefer to linguistically represent likelihoods because of the uncertainty and vagueness of risk factors. The objective of this paper is to identify risk mitigation strategies in software development projects from the perspectives of software practitioners and determine the effectiveness of these strategies. We explore the use of fuzzy methods to overcome the problems associated with probabilistic modelling through a set of questionnaire surveys which was conducted among 3000 IT practitioners using Tukey-B test, Kendall's test and Post Hoc Tukey HSD test. We apply Fuzzy Membership Function (Fuzzy-MBF) as an appropriate mechanism in dealing with the subjectivity in the assessment of risk factors in different stages of a software development life cycle. The proposed Fuzzy-MBF offers a quantitative evaluation of risk factors and provides a systemic evaluation of risk and visualization of results.

Keywords: risk prevention, risk strategy, software development, non-technical strategy, IT project

1.0 INTRODUCTION

Software projects are especially subject to bounded rationality, induced by cost and schedule constraints, rescue limitations, and organizational and technological uncertainties [1]. Evidence indicates that risks in IT projects which are not effectively managed and lack of identification and management during the life cycle of a project can lead to their failure [2, 3, 4]. In software projects, the failure may involve increased costs, longer completion times, reduced scope, reduced quality, reduced realization of proposed benefits or reduced stakeholder satisfaction [5]. Besides, the monetary cost of poor performance and failure is high but the value of missed benefits is substantial [6]. Articles emphasized the importance of empirically categorising the sources and types of risks associated with software development projects [7, 8, 9]. Software risk management have been promoted as an approach to reduce project failure and improve software project outcomes [10, 11]. Fuzzy logic has long been recognized as a useful method in handling inexact and vague information because of its ability to utilize natural language in terms of linguistic variables [12]. The arithmetic and calculus of fuzzy sets and fuzzy numbers provide a method for manipulating these imprecise representations. Therefore, by using fuzzy methods, decision making processes can be modelled even with limited project information in the early stage. These methods can provide greater analytic capability while being a good match with the natural inclination of management professionals in the software development discipline [13]. The goal of this paper is to identify risk mitigation strategies in software development projects from the perspectives of software practitioners and determine the effectiveness of these strategies. This paper illustrates the suitability of fuzzy model for solving imprecise and subjective problems, in contrast to the traditional risk matrix-based assessment techniques. Our proposed model aims to offer quantitative values of the risk factors and parameters for decision making. In a probabilistic approach, the impreciseness and uncertainty were modelled by expressing the belief that an event either occurs or does not. In contrast, fuzzy logic membership functions (MBF) express the possibility of an outcome rather than the likelihood of an outcome. Therefore, by using fuzzy logic, the uncertainty is modelled as a degree of membership in the set that defines an outcome.

2.0 PROBLEM STATEMENTS

Software projects are dynamic and tend to have volatile requirements. The requirements may cause project scope to change frequently, making them especially difficult to manage and control [14]. Literature on how to manage a development project often refers to cost, time and quality as the key project success criteria but there are also many different, broad and overlapping definitions of project success and failures [15, 16]. Generally, incomplete project information is available during the very early phases of the software development project. Many decision making processes occur in an environment that the goals, constraints and consequences of possible actions are not precisely known [17]. Within software development processes, , several mathematical programming models have been reported such as multi-attribute decision making and dynamic non-linear programming to improve the decision making process with limited information [18]. However, in practice, many decision makers refrain from using such techniques or models due to complex programming and implementing processes [19]. Mathematical programming methods require precise data to develop and analyse meaningful risk factors. However, most software developers are more comfortable viewing risk factors through linguistic values (e.g., high, moderate, low, and likely or unlikely) rather than using probabilistic terminology [20]. The scoring methods or ranking methods can have a compensatory bias. For example, when one criterion has a low value, other criteria may offset it. Hence, subjective human ratings and evaluation processes can be better approximated using ‘fuzzy’ measures than using the commonly applied additive measures [20]. Quantitative approaches may introduce compensatory biases, e.g., when one criterion has a low value then other criteria may offset it, resulting in a project with a high weighted score being accepted despite significant risk. Such extremely low or high values have a significant influence on averages, potentially resulting in misleading conclusions and interpretations [16]. Rather in these probabilistic approaches, then, subjective human ratings and evaluation processes can be better approximated using fuzzy measures, allowing appropriate decision making processes to be modelled and justified even with limited project information. This is becoming increasingly important as qualitative issues such as behavioural, political and other organizational concerns are becoming increasingly crucial to project success. Despite these possibilities, most quantitative risk assessment methods are still based on probabilistic methods rather than fuzzy methods. Therefore, it remains unclear how fuzzy methods can effectively model the early stages of the project and identify and allow management of probable risk factors [15].

3.0 SOFTWARE RISK MANAGEMENT STRATEGIES: CONVENTIONAL METHODS AND FUZZY METHODS

3.1 Conventional methods in managing software risk

Software risk management usually consisting of quantifying the importance of a risk (i.e., assessing its probability of occurrence and its impact on the project performance) and developing strategies to control it [21]. Literatures suggest a need to improve the management of threats to software projects [22, 23, 24]. Risk analysis techniques can be either qualitative or quantitative depending on the information available and the level of details required [25]. Quantitative techniques rely heavily on statistical approaches including: Monte Carlo Simulation, Fault and Event Tree Analysis, Sensitivity Analysis, Annual Loss Expectancy, Risk exposure and Failure Node and Effect analysis [26, 27, 28]. More generic approaches for software risk management include risk lists, risk action lists, risk strategy models and risk strategy analysis [29, 30, 31, 32]. Many practice-based approaches also exist such as Prince2, CMMI, COBIT, ITIL, ITGI, NIST and COSO [33].

Iterative risk management steps usually include risk identification, risk analysis, risk response and risk monitoring and control [34, 35]. Other approaches of risk management are also conducted as below:
a) Emphasize early development life cycle risk avoidance in favour of late life cycle testing to eliminate software defects [36],
b) Scenario-based risk management [37],
c) Modelling operational risks via Bayesian Networks [38],
d) Software risks within a socio technical model of organizational change [13],
e) Life cycle-based enterprise security risk management [39] and
f) Real options approaches to managing incomplete knowledge in projects [40].

Risk management strategies are typically employed in the development process to reduce the risks inherent in software projects. As risks vary in nature, a particular kind of strategy may only reduce certain aspect of risk but not others [41]. Software project managers need to be aware that only a few IT risks have to do with technical issues, and managerial strategies are always the key strategies [42]. Real-time monitoring and regular updating could ensure a software project progresses well with expected budget, schedule and quality [43]. Previous studies of software development risks focus on generic risk management strategies as summarized in Table 1. Most of these existing studies focus on anecdotal evidence and are limited to a narrow portion of the development processes, or even the broad perspectives of general project performance. However, the frameworks and systematic models of risk management proposed by these researches predominantly dealt with specific techniques. These generic risk management strategies have not been studied to include the perspectives of the software development team personnel such as project managers, developers and IT support staff.

Table 1: Researches undertaken in IT risk management

YEAR	RESEARCHER	RESEARCH AREA	Risk mitigation strategies	Point of view
2010	Mahaney & Lederer [43]	Role of monitoring and shirking in IS project management	Project monitoring Regular updating of project against goals	Project managers
2008	Su et al. [44]	Impact of user review on software responsiveness	Users' involvement	Project managers
2007	Dey et al. [45]	Risk management framework for software development projects from developers' perspective, using a case study of public sector organization in Barbados.	Users' involvement Scope management planning Establish clear client requirements Resource planning Process re-engineering Benchmarking Effective communications Unit or independence testing Establish scope Develop work breakdown structure Control mechanism	Developers
2007	Tesch et al. [46]	IT project risk perspective of project management professionals (PMP)	Team communication Project managers leading role and experience Customers' support Top management backing Plan project in phases Project planning Proper budgeting Develop resource allocation planning Contingency plan to maintain project Re-evaluate project CBA Use change management process Conduct feasibility study Pilot and prototype technology before rolling into organization Alternative technology and development methodology Clear scope requirements Project control mechanism Users' participation commitment Develop approaches to get feedback Set up key milestones	Views of project management professionals (PMP)
2004	Wallace et al. [9]	Identification of risks that posed threat to successful project outcomes. Investigation of dimensions of risk and an exploratory model, on the software project performance.	Strategies related to project scope and requirements Strategies related to project execution Experienced project team members Experienced project managers Project planning and control techniques Identified scope and requirements Planning control mechanism Assembling high-skilled project team Training Users' involvement Top management's involvement Counter risk associated with organizational environment, users, requirements, project complexity. Good project management practices	Project managers
2004	Baccarini et al. [4]	In-depth interviews with IT professionals from leading firms in Western Australia to determine how IT risks were managed in their projects, where the respondents ranked IT risks in terms of	Manage the relationship Project planning and schedule management Manage expectations Obtain management support Develop customer relationship Maintain market entry barrier Establish sound requirements Plan for resources Plan contingency options	Interview with IT professionals

		likelihood and consequences to identify the most important risks.	Assess project staff capability Change project management objectives Manage stakeholders Executive management support Clear scope definition Develop clear requirements definition Adequate documentation Perform group reviews Progressive signoff of milestones Comprehensive testing Customers' support Formal change management process Consult/educate users Project monitoring Project managers experience Roles and responsibilities clearly defined Clear communication External consultants	
2000	Jiang & Klien [42]	Impact of the spectrum of risks on different aspect of systems development and project effectiveness	Interpersonal and team skills Skills training Users' participation and user commitment Clearly defined roles Clear project scope and task Clear communications Users' experience Control of conflicts	Survey of project managers

3.2 Fuzzy methods in managing software risk

Fuzzy Set Theory (FST) was introduced by Zadeh in 1965 [44] in order to overcome problems involving vagueness. Since then, it has been widely applied to situations in management, economics and engineering [45]. Fuzzy logic extends the probabilistic models for estimating costs and risks of software Verification, Validation, and Testing (VVT). It is necessary to meet the assumption that software failures occur largely due to the ineffective performance of software and systems VVT, but only limited at the phase of testing [20]. A hierarchical model was then developed in 1999 to evaluate aggregative risks in software development and rate aggregative risk in a fuzzy environment [46]. Each risk item was categorized into two fuzzy sets with triangular membership functions (i.e., grades of risk, grades of importance and the rate of risk). In subsequent studies, the rate of each individual risk item was evaluated using a two-stage fuzzy assessment method within a group of decision making settings, using 13 linguistic values [47]. However, the two-stage fuzzy assessment increases the complexity and period of modelling. A fuzzy expert system using fuzzy expert rules to support assessment of operational risk of software due to software failures during the very early phases of the software life cycle was developed in [48]. The model was generated with assistance from the experts in software engineering. The model focuses on operational risk factors, especially technology risk and software developers' competence. A Fuzzy Decision Support System (FDSS) was presented in [49] to assist e-commerce project managers to identify potential risk factors and evaluate the corresponding e-commerce development risks. However, the web-based design focusing on risk identification was the core focus of the FDSS, and less development attention was given to the risk management, planning, and monitoring functions of the model. While all these models are useful for software development, each of them is limited by a restricted focus on particular elements of the problem [49].

4.0 Research Design and Methods

Risk management strategies for software development projects extracted from literatures are listed in Table 1. These risk strategies are then validated in a pilot study with fifteen experienced academics. The reason for conducting the pilot study is to test the suitability and comprehensibility of the questionnaires used in this survey. The main purpose of this set of generic strategies is to obtain the perceptions of IT practitioners on managerial risks rather than technical issues. In the main survey, thirty strategies are included in the questionnaires as shown in Table 2. Respondents were requested to rate the effectiveness of the risk strategies in response to the risk factors using the Likert scales of 0-6.

Table 2: Risk mitigation strategies used in the research

No	Ref	RISK MITIGATION STRATEGIES
1	S1	Define a clear goals and objectives of the project
2	S2	Conduct a thorough analysis feasibility study
3	S3	Use of project tracking system and regular updating
4	S4	Proper project planning and scheduling
5	S5	Identify critical and non-critical activities
6	S6	Set key performance indicators and standards for stages/processes
7	S7	Lesson learned from past software development projects
8	S8	Identify success criteria
9	S9	Consistent commitment of management
10	S10	Quality control procedure
11	S11	Risk management methodology/techniques/tools
12	S12	Hire external expertise/consultant
13	S13	Contingency plan
14	S14	Conduct pilot testing
15	S15	Prototyping
16	S16	Thorough analysis of development methodology
17	S17	Proper timeframe for testing
18	S18	Conduct a thorough user acceptance test
19	S19	Planned for parallel or phased conversion
20	S20	Developed a clear and detail requirements
21	S21	Incorporate alternative development methodology
22	S22	Backup the system thoroughly
23	S23	Software security checklist and authentication process
24	S24	Cost control procedure
25	S25	Technical support team
26	S26	Proper planning of resources
27	S27	Effective training for staff
28	S28	Effective lines of communication
29	S29	Good project management and leadership
30	S30	Greater degree of users' involvement and commitment

The Kendall's test is used to determine whether there are differences between respondents' rankings of risk mitigating strategies. Further, one-way analysis of variance (ANOVA) is conducted to compare the means of respondents and determine if there are any significant differences among them.

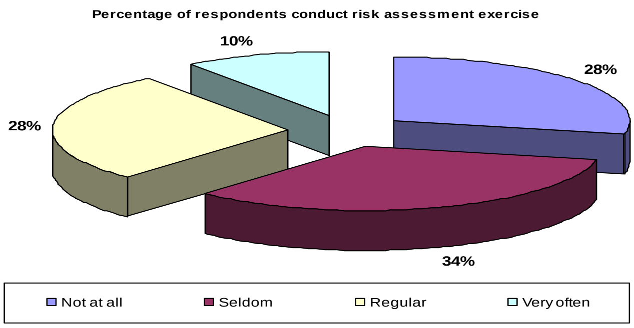

Fig. 1: Percentage of respondents who conduct risk assessment exercise

Due to the uneven sample sizes of the IT professional respondents, the Tukey-B test is employed to check the validity of the results. 3000 sets of questionnaires were distributed and 324 sets of them were answered with validity, forming a responding rate of 10.8%. More than half of the respondents (i.e., 61%) have more than 10 years of experience in software development with an overall average of 11.8 years and standard deviation of 5.29 years. Furthermore, the overall average number and standard deviation of software projects that the respondents were involved with is 9 and 5.31, respectively. These statistics prove that the respondents have sufficient insight into software development processes. The wealth of experience among the respondents gives reasonable support for the concluding arguments in the discussion section. The respondents were also requested to answer whether risk assessment exercise had been carried out in their software projects development. From the responses, 28% of the respondents did not conduct risk assessment exercise at all, while the other 72% of respondents conducted the risk assessment occasionally and regularly according to Fig. 1. Furthermore, 68% of the companies have 1 to 4 members who are expert in risk management. However, 40% of the respondents were not satisfied with their risk management practice in software development as shown in Fig. 2.

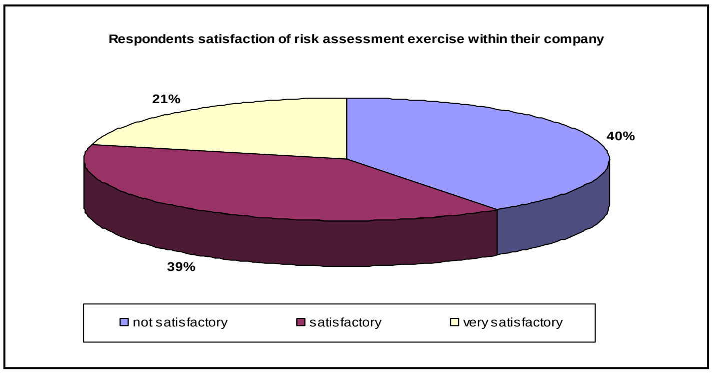

Fig. 2: Respondents' satisfaction in risk assessment exercise

4.2 Mathematical Development of Fuzzy Method in This Study

Although fuzzy theory deals with imprecise information, it is based on mathematical theory [50] and usually involves defined phases of modelling development (Fig. 1). A variable in fuzzy logic has a set of values characterised by linguistic expression (e.g., high, medium and low). Linguistic variables provide a means of modelling human tolerance for imprecision by encoding decision-relevant information into labelled fuzzy sets. These linguistic expressions therefore become numerically represented by fuzzy sets. Eq. 1 takes into account the weighting of each risk factor. Extraction of the Membership Function (MBF) from the sets is the most important aspect in the development of FDSS. Each fuzzy set carries a distinct MBF in [0, 1]. The degree of membership varies from 0 (non-membership) to 1 (full membership). This is in contrast to crisp or conventional sets, where an element is either a part of or not part of the sets. Besides, authors in [51] developed an approximation technique that applies alpha cuts horizontal lines which creates cross-section at the level of membership. MBF for each risk factor is calculated based on the estimated alpha cuts. Then, an average weighted membership based on MBF of its factors is computed using Eq. 2 for the score for each category of risk factors, where $F_{ij}(x)$ is the MBF at a certain alpha cut and W_i is the weighting coefficient. The MBF can be constructed based on statistical characteristics; i.e., the average weighted mean and standard deviation. The real values of the risk factors are then transformed into linguistic values (viz., low, moderate and high).

$$F(y) = \sum W_i F_i(x) \qquad \text{for sum } W_i = 1 \qquad (1)$$
$$Fi(y) = \sum W_i F_{ij}(x) / \sum W_i \qquad (2)$$

To define a representative MBF, conditions are imposed to make characteristics consistent with decision makers' subjective judgement. The mean and standard deviation of the risk factors are used to develop the MBF of risk factors, where the fuzzy MBF of x is defined as $F(x)$, in [0,1], as estimated by Eq. (3), (4), (5), and (6) [45].

i. For low-level of significance of a defined risk factor:
$$F(x) = |(a-x)/b| \qquad \text{for} \quad a-b < x < a \qquad (3)$$
ii. For medium-level of significance of a defined risk factor:
$$F(x) = |(x-a+b)/b| \qquad \text{for} \quad x < a \qquad (4)$$
$$F(x) = |(x-a-b)/b| \qquad \text{for} \quad x > a \quad \text{or} \quad x = a \qquad (5)$$
iii. For high-level of significance of a defined risk factor:
$$F(x) = |(x-a)/b| \qquad \text{for} \quad a < x < a+b \qquad (6)$$

In Eq. (3) – (6), the 'a' is the average mean and 'b' is the standard deviation. There is a focal central member 'a' to make $F(x)$ greater than other members of the set. Meanwhile, 'b' is a controlling scale factor parameter. As shown in Fig. 3, these parameters influence the shape and distribution of the equations. The horizontal scale values represent the level of significance of a risk factor. The fuzziness and MBF are increased or decreased by the parameters 'a' and 'b'.

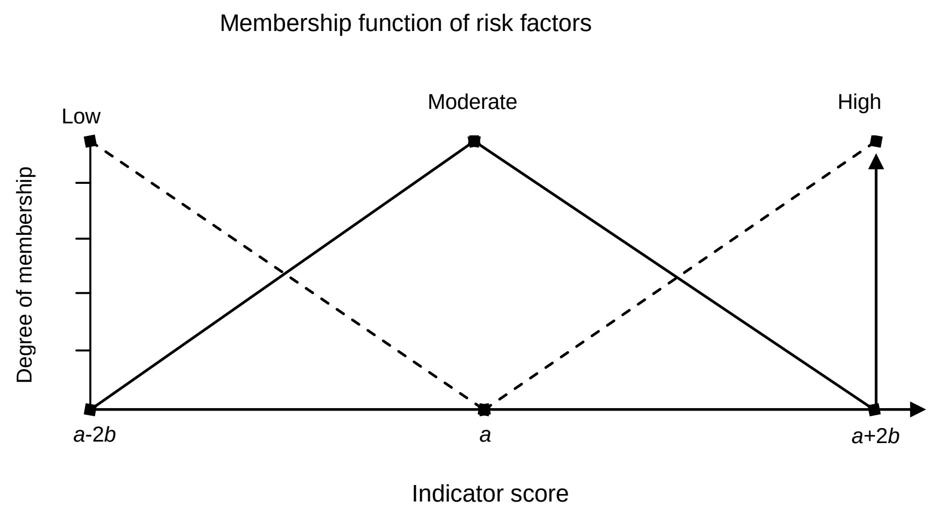

Fig. 3: Membership function of risk factors

Since the mean value is an unbiased estimate for any sample set, it is an ideal choice for '*a*'. Fig. 3 shows the range of moderate-level of significance of risk factors from (*a*-2*b*) to (*a*+2*b*), with the highest degree of membership occurring at the value of (*a*-2*b*) for low-level of significance. For medium-level of significance, '*a*' is used. (*a*+2*b*) is used for high-level of significance. In developing the scoring system, the values of the risk factors are transformed into linguistic values through linguistic variables. More linguistic variables can accommodate a wider range of scenarios and possibly greater accuracy, at the expense of greater complexity. The fuzzy approach in this work was modelled by three linguistic terms, i.e., low, moderate and high. Fig. 4 shows the MBF of the degree of weights for risk factors. The fuzzy combination process of the scores and weights, using the concept of alpha cut point (horizontal cross-sections at various levels of membership) is represented in Fig. 5, adopted from [45]. One triangle represents the MBF of risk factors and the other triangle represents the MBF of degree of significance of risk factors (probability of occurrence). For example, from point P on the vertical axis, degree of membership, a horizontal line is drawn and point P shows the alpha cut point. This creates two intersections with the other two sides of triangle. These intersections are shown as P1 and P2 in triangle A. Meanwhile, they are shown as P3 and P4 in triangle B. The extrapolation of P1 and P2 have scores of (*a*-*c*) and (*a*+*c*). As for P3 and P4 in triangle B, the extrapolation shows the extracted weight of (f) and (h) relating to the score in triangle A. It sums as $WjFij\ (x) = \{(a - 2b) * e\} + \{(a + 2b) * i\} + \{(a – c) * f\} + \{(a + c) * h\} + \{a * g\}$. The fuzzy computation is shown in Fig. 6.

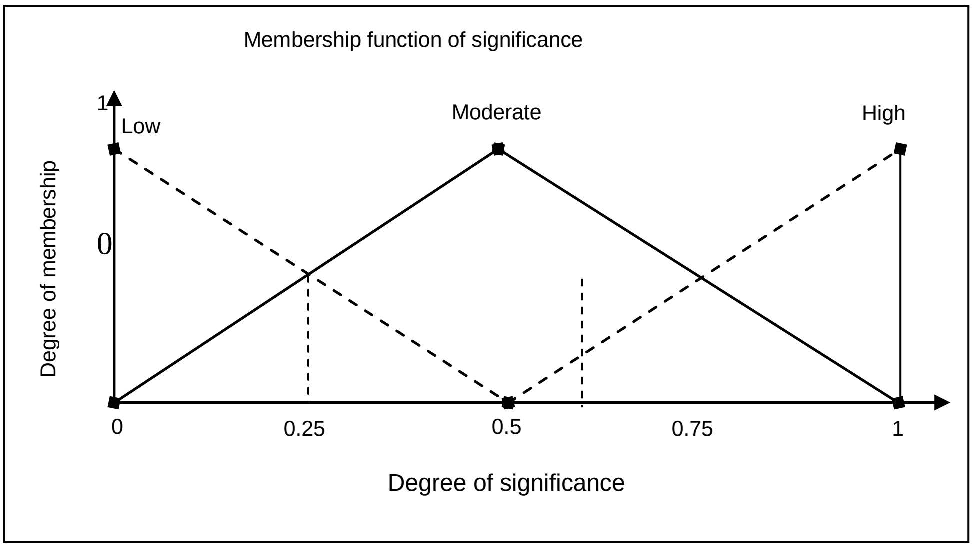

Fig. 4: Membership function of degree of significance for the risk factors

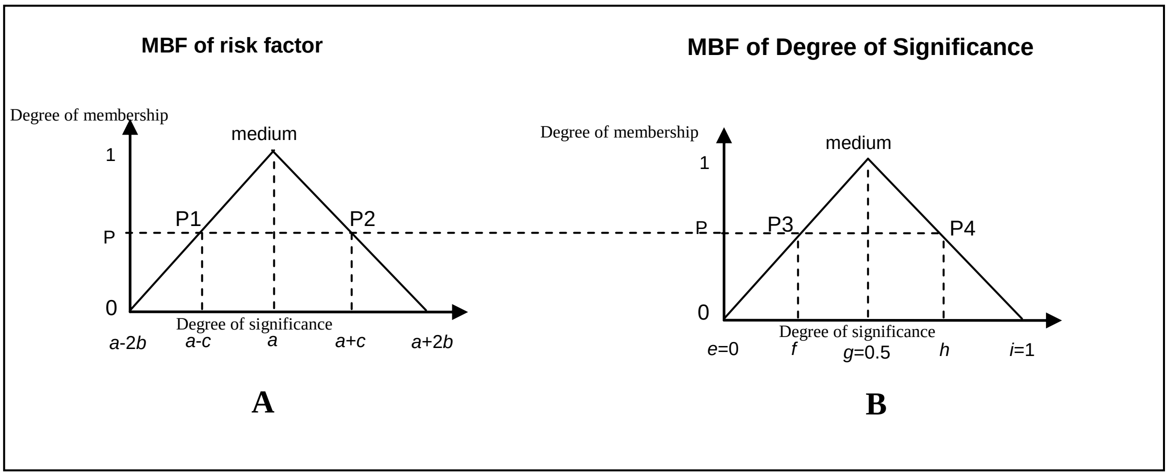

Fig. 5: Fuzzy combination process

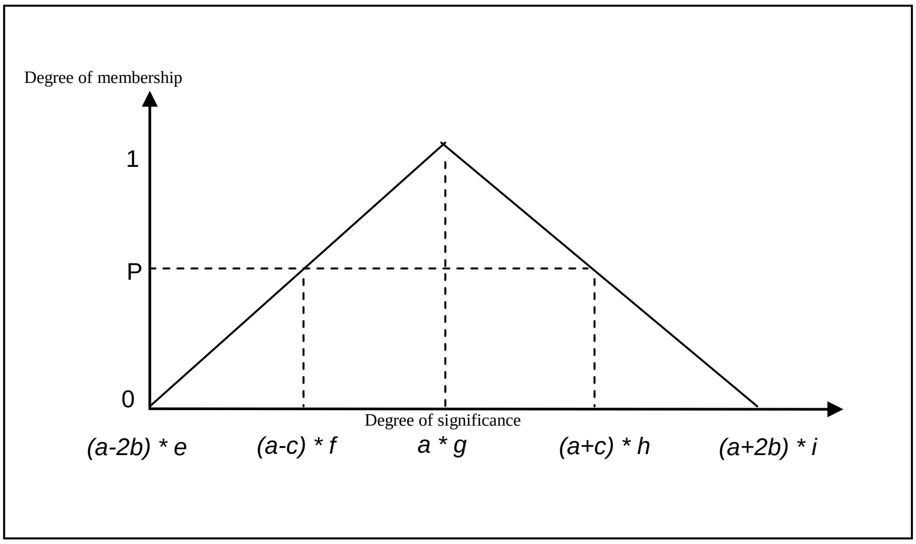

Fig. 6: Fuzzy computation

5.0 RESULTS AND FINDINGS

5.1 Kendall's mean rank for risk mitigation strategies

The rating of the risk mitigation strategies is shown in Table 3. The average rating of the mitigation strategies varies from 1.32 to 5.31. The overall top five strategies are s30, s28, s26, s20 and s21. The lowest strategy is s19 (mean=1.77). All four groups of respondents agreed that s30 (i.e., greater degree of users' involvement and commitment) deserved the top ranking (overall mean=5.19), where the developer group rated s30 with the highest mean value of 5.31. The mean values of s30 given by the other three groups including board of directors, project managers and IT staff are 5.09, 5.19 and 4.84, respectively. A close scrutiny of the results reveals that the two most effective strategies (s30 and s28) as perceived by the board of directors are mostly consistent with those perceived by project managers and developers. However, IT support staff perceived s28 less important than s29. The director group perceived s29 as the second important strategy while the project manager group and the developer group perceived otherwise. Specifically, s29 was perceived by the IT staff group as unimportant. Such differences in ranking the effectiveness of risk strategies reflect the differences in roles and responsibilities in the management of software development.

Table 3: Kendall’s mean rank for risk mitigation strategies

Strategy	Board		Project Manager		Developer		IT staff		Overall		Kendall’s mean rank
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	
S1	5.02	0.802	5.01	0.718	5.14	0.707	4.76	0.723	5.04	0.731	22.85
S2	4.33	1.012	4.50	1.085	4.61	1.078	4.12	0.927	4.49	1.066	18.78
S3	2.78	0.786	2.79	0.823	2.88	0.818	2.48	0.770	2.80	0.815	8.53
S4	4.98	0.774	5.04	0.711	5.11	0.714	4.84	0.688	5.04	0.720	22.77
S5	5.02	0.802	5.01	0.718	5.14	0.707	4.76	0.723	5.04	0.731	22.85
S6	2.61	0.714	2.75	0.817	2.81	0.816	2.40	0.645	2.73	0.796	8.61
S7	5.02	0.802	5.00	0.723	5.14	0.707	4.76	0.723	5.04	0.733	22.83
S8	5.02	0.745	5.05	0.705	5.11	0.714	4.84	0.688	5.05	0.713	22.85
S9	3.39	1.085	3.33	1.037	3.63	1.052	2.92	0.954	3.42	1.057	11.34
S10	1.78	0.841	1.82	0.809	1.86	0.794	1.44	0.651	1.80	0.801	3.90
S11	4.37	1.123	4.42	1.089	4.68	1.069	3.92	0.954	4.47	1.091	18.50
S12	1.67	0.701	1.81	0.793	1.89	0.814	1.40	0.645	1.79	0.786	3.78
S13	4.22	1.114	4.13	1.025	4.42	1.089	3.76	0.831	4.22	1.061	16.83
S14	4.37	1.040	4.65	1.074	4.77	1.041	4.16	1.028	4.62	1.065	19.56
S15	4.98	0.830	4.93	0.714	5.04	0.744	4.72	0.678	4.96	0.741	22.23
S16	1.72	0.750	1.90	0.791	1.99	0.779	1.48	0.714	1.88	0.785	4.09
S17	2.70	0.785	2.61	0.773	2.77	0.810	2.24	0.523	2.65	0.782	7.90
S18	3.33	0.967	3.51	1.006	3.72	1.029	3.08	0.909	3.53	1.015	12.25
S19	1.78	0.728	1.79	0.783	1.85	0.769	1.32	0.557	1.77	0.764	3.61
S20	5.07	0.772	5.11	0.730	5.26	0.659	4.80	0.645	5.14	0.713	23.53
S21	5.04	0.893	5.07	0.755	5.13	0.723	4.80	0.764	5.06	0.766	23.01
S22	1.80	0.806	1.87	0.771	2.06	0.731	1.32	0.557	1.89	0.769	4.17
S23	3.30	1.030	3.36	1.033	3.44	1.067	3.00	1.041	3.35	1.047	11.32
S24	2.74	0.773	2.77	0.801	2.98	0.784	2.32	0.557	2.81	0.791	8.82
S25	2.61	0.745	2.60	0.765	2.68	0.794	2.40	0.707	2.61	0.769	7.76
S26	5.07	0.772	5.13	0.706	5.26	0.659	4.80	0.645	5.14	0.703	23.58
S27	4.39	1.000	4.42	1.011	4.50	1.036	4.00	1.041	4.41	1.024	18.12
S28	5.07	0.772	5.13	0.706	5.26	0.659	4.80	0.645	5.14	0.703	23.58
S29	5.09	0.755	5.08	0.702	5.14	0.719	4.76	0.779	5.08	0.725	23.14
S30	5.09	0.812	5.19	0.714	5.31	0.636	4.84	0.624	5.19	0.704	23.93

The top ten strategies were dominated by S30, S29, S28, S26, S21, S20, S8, S5, S4 and S1 with mean values of 4.76 and above. The top five strategies have clear objectives and requirements, planning of scheduling and resources, identification of success criteria and critical activities, project leadership, users’ commitment and effective lines of communication. It is interesting to point out these top ten strategies were predominantly non-technical risks including project planning or organizational issues and project management related matters. Table 3 also indicates S10, S12 and S19 as the least effective strategies (i.e., quality control procedure, hiring of external consultant or expertise and undertaking a parallel or phased conversion) with mean values below 2.0. It is interesting to observe that quality control procedures are not effective, probably because the survey participants perceive quality control as part of project management function. Noticeably, the developer group consistently rated all the mitigation strategies higher than other respondents. The perceptions of the managing director group and project manager group are nearly the same for all the strategies except for s13.

Kendall’s coefficient of concordance test (W) was conducted to measure the agreement among the respondents. Kendall (W) ranges between 0 (non-agreement) and 1 (complete agreement). From the Kendall (W) rank test, the mean ranks of the risk factors were in the range of 3.61 and 23.93. The top ten strategies are S30, S29, S28, S26, S21, S20, S8, S5, S4 and S1, which are consistent with the rating by the mean values. With a significance level of 0.05, the W-value of 0.842 in Table 4 indicates a significant level of agreement among respondents. Kendal coefficient concordance test indicates a common concordance among the respondents regarding the most important strategy and the least important strategy.

Table 4: Kendall's coefficient of concordance

N	324
Kendall's W	0.842
Chi-Square	7910.714
d.f.	29
Asymp. Sig.	0.000

5.2 ANOVA analysis

ANOVA analysis was conducted to identify the statistical variances among variables. The testing hypothesis was:

H_0 ($p > 0.05$): There is no significant difference among the respondents' rating for the effectiveness of risk mitigation strategies.

H_1 ($p < 0.05$): At least one group's rating for the effectiveness of risk mitigation strategies significantly differs from others.

Table 5 indicates a statistically significant difference among some risk mitigation strategies. The four groups of respondents' responses differ significantly on 15 out of 30 mitigation strategies, including: S9, S11, S12, S13, S14, S16, S17, S18, S19, S20, S22, S24, S26, S28 and S30. In other words, at least one group significantly differs from others and thus, the null hypothesis must be rejected. In order to discover which specific means are different from another, a follow-up test called Post Hoc Multiple Comparison Test was conducted. Due to the uneven sample sizes, the Tukey Post Hoc test was conducted. The fifteen risk strategies that were significantly different are shown in Table 6.

Table 5: ANOVA test for risk strategies

		Sum of Squares	df	Mean Square	F	Sig.
S1	Between Groups	3.397	3	1.132	2.143	0.095
	Within Groups	169.082	320	0.528		
	Total	172.478	323			
S2	Between Groups	6.386	3	2.129	1.889	0.131
	Within Groups	360.565	320	1.127		
	Total	366.951	323			
S3	Between Groups	3.380	3	1.127	1.707	0.165
	Within Groups	211.175	320	0.660		
	Total	214.556	323			
S4	Between Groups	1.756	3	0.585	1.131	0.337
	Within Groups	165.639	320	0.518		
	Total	167.395	323			
S5	Between Groups	3.397	3	1.132	2.143	0.095
	Within Groups	169.082	320	0.528		
	Total	172.478	323			
S6	Between Groups	4.261	3	1.420	2.269	0.080
	Within Groups	200.292	320	0.626		
	Total	204.552	323			
S7	Between Groups	3.466	3	1.155	2.174	0.091
	Within Groups	170.089	320	0.532		
	Total	173.556	323			
S8	Between Groups	1.565	3	0.522	1.027	0.381
	Within Groups	162.543	320	0.508		
	Total	164.108	323			
S9	Between Groups	12.360	3	4.120	3.784	**** 0.011***
	Within Groups	348.390	320	1.089		
	Total	360.750	323			
S10	Between Groups	3.808	3	1.269	1.996	0.115
	Within Groups	203.550	320	0.636		
	Total	207.358	323			
S11	Between Groups	13.438	3	4.479	3.861	**** 0.010***
	Within Groups	371.253	320	1.160		
	Total	384.691	323			
S12	Between Groups	5.682	3	1.894	3.123	**** 0.026***
	Within Groups	194.047	320	0.606		
	Total	199.728	323			
S13	Between Groups	11.382	3	3.794	3.449	**** 0.017***
	Within Groups	352.059	320	1.100		
	Total	363.441	323			

S14	Between Groups	11.007	3	3.669	3.302	***0.021**
	Within Groups	355.536	320	1.111		
	Total	366.543	323			
S15	Between Groups	2.349	3	0.783	1.430	0.234
	Within Groups	175.206	320	0.548		
	Total	177.556	323			
S16	Between Groups	6.756	3	2.252	3.747	***0.011**
	Within Groups	192.306	320	0.601		
	Total	199.062	323			
S17	Between Groups	6.192	3	2.064	3.457	***0.017**
	Within Groups	191.092	320	.597		
	Total	197.284	323			
S18	Between Groups	11.297	3	3.766	3.749	***0.011**
	Within Groups	321.453	320	1.005		
	Total	332.750	323			
S19	Between Groups	5.840	3	1.947	3.409	***0.018**
	Within Groups	182.713	320	0.571		
	Total	188.552	323			
S20	Between Groups	5.031	3	1.677	3.375	***0.019**
	Within Groups	158.994	320	0.497		
	Total	164.025	323			
S21	Between Groups	2.233	3	0.744	1.271	0.284
	Within Groups	187.406	320	0.586		
	Total	189.639	323			
S22	Between Groups	11.911	3	3.970	7.103	***0.000**
	Within Groups	178.864	320	0.559		
	Total	190.775	323			
S23	Between Groups	4.132	3	1.377	1.260	0.288
	Within Groups	349.757	320	1.093		
	Total	353.889	323			
S24	Between Groups	9.979	3	3.326	5.539	***0.001**
	Within Groups	192.157	320	0.600		
	Total	202.136	323			
S25	Between Groups	1.655	3	0.552	0.934	0.425
	Within Groups	189.119	320	0.591		
	Total	190.775	323			
S26	Between Groups	4.950	3	1.650	3.417	***0.018**
	Within Groups	154.520	320	0.483		
	Total	159.469	323			
S27	Between Groups	5.190	3	1.730	1.661	0.175
	Within Groups	333.390	320	1.042		
	Total	338.580	323			
S28	Between Groups	4.950	3	1.650	3.417	***0.018**
	Within Groups	154.520	320	.483		
	Total	159.469	323			
S29	Between Groups	3.047	3	1.016	1.948	0.122
	Within Groups	166.867	320	0.521		
	Total	169.914	323			
S30	Between Groups	5.355	3	1.785	3.690	***0.012**
	Within Groups	154.781	320	0.484		
	Total	160.136	323			

According to the Tukey Post Hoc test results as presented in Table 6, the significant value which is less than 0.05 (i.e., $p < 0.05$) shows a significant difference between the respondents' group regarding the listed strategies. For example, the IT staff group and the developer group have significantly different rating for strategies S9, S11, S13, S14, S18, S20, S26, and S28. The IT staff group also has different rating from the other three groups for strategies S17, S19, S22, and S24. The managing director group and the project manager group reveal a consensus of agreements in ratings the strategies. Even though the Kendal concordance test showed a common consensus, the ANOVA and Post Hoc test proved that the four respondents groups rated the strategies quite differently.

Table 6: Tukey HSD Post Hoc Multiple Comparison Test

Strategies	Groups	N	Subset for alpha = 0.05	
			1	2
S9	IT staff	25	2.92	
	project manager	135	3.33	3.33
	board of directors	46	3.39	3.39
	developer	118		3.63
S11	IT staff	25	3.92	
	board of directors	46	4.37	4.37
	project manager	135	4.42	4.42
	developer	118		4.68
S12	IT staff	25	1.40	
	board of directors	46	1.67	1.67
	project manager	135		1.81
	developer	118		1.89
S13	IT staff	25	3.76	
	project manager	135	4.13	4.13
	board of directors	46	4.22	4.22
	developer	118		4.42
S14	IT staff	25	4.16	
	board of directors	46	4.37	4.37
	project manager	135	4.65	4.65
	developer	118		4.77
S16	IT staff	25	1.48	
	board of directors	46	1.72	1.72
	project manager	135		1.90
	developer	118		1.99
S17	IT staff	25	2.24	
	project manager	135		2.61
	board of directors	46		2.70
	developer	118		2.77
S18	IT staff	25	3.08	
	board of directors	46	3.33	3.33
	project manager	135	3.51	3.51
	developer	118		3.72
S19	IT staff	25	1.32	
	board of directors	46		1.78
	project manager	135		1.79
	developer	118		1.85
S20	IT staff	25	4.80	
	board of directors	46	5.07	5.07
	project manager	135	5.11	5.11
	developer	118		5.26
S22	IT staff	25	1.32	
	board of directors	46		1.80
	project manager	135		1.87
	developer	118		2.06
S24	IT staff	25	2.32	
	board of directors	46		2.74
	project manager	135		2.77
	developer	118		2.98
S26	IT staff	25	*4.80*	
	board of directors	46	*5.07*	*5.07*
	project manager	135	*5.13*	*5.13*
	developer	118		*5.26*
S28	IT staff	25	4.80	
	board of directors	46	5.07	5.07
	project manager	135	5.13	5.13
	developer	118		5.26
S30	IT staff	25	4.84	
	board of directors	46	5.09	5.09
	project manager	135		5.19
	developer	118		5.31

Statistically significant differences were found in four strategies including: S30, S28, S26, and S20, whose overall *F*-values were: $F(3,320)=3.690$, $p=0.012$; $F(3,320)=3.417$, $p=0.018$; $F(3,320)=3.417$, $p=0.018$; and $F(3,320)=3.375$, $p=0.019$, respectively. Owing to the difference in group sizes, Post Hoc Tukey HSD was conducted to determine where these differences belong to. Results show that the developer group and the IT staff group differed significantly over their responses for strategies S30 (p=0.012); S28 (p=0.014); S26 (p=0.014); and S20 (p=0.016). To look for significant agreements between the four respondents groups, the Kendall's nonparametric test was conducted. Kendall's coefficient of concordance shown in Table 7 and Table 8 provide a measure of agreement between 0 and 1. Specifically, 0 indicates no agreement and 1 implies perfect concordance. In this survey, Kendall's coefficients range between 0.83 and 0.88. These high values of Kendall's coefficient indicates a strong agreement among groups and rejects the null hypothesis that there is no agreement among survey respondents ($p < 0.05$).

Table 7: Kendall's concordance test

Respondents' category	Degree of freedom	Chi-square	Kendall's coefficient (W)	Significance
Board	29	1109.789	0.832	0.000
Project Manager	29	3247.840	0.830	0.000
Developer	29	2926.970	0.855	0.000
IT staff	29	643.489	0.888	0.000

Table 8: The Kendall's mean rank and significant value for ANOVA

	Board			Project Manager			Developer			IT staff			Overall			K	Sig. Value
	M	SD	R	M	SD	R	M	SD	R	M	SD	R	M	SD	R		
S1	5.02	0.802	10	5.01	0.718	10	5.14	0.707	8	4.76	0.723	11	5.04	0.731	10	22.85	0.095
S2	4.33	1.012	16	4.50	1.085	14	4.61	1.078	15	4.12	0.927	14	4.49	1.066	14	18.78	0.131
S3	2.78	0.786	21	2.79	0.823	21	2.88	0.818	22	2.48	0.770	21	2.80	0.815	22	8.53	0.165
S4	4.98	0.774	12	5.04	0.711	8	5.11	0.714	11	4.84	0.688	3	5.04	0.720	8	22.77	0.337
S5	5.02	0.802	9	5.01	0.718	9	5.14	0.707	7	4.76	0.723	10	5.04	0.731	9	22.85	0.095
S6	2.61	0.714	25	2.75	0.817	23	2.81	0.816	23	2.40	0.645	22	2.73	0.796	23	8.61	0.080
S7	5.02	0.802	8	5.00	0.723	11	5.14	0.707	6	4.76	0.723	9	5.04	0.733	11	22.83	0.091
S8	5.02	0.745	7	5.05	0.705	7	5.11	0.714	10	4.84	0.688	2	5.05	0.713	7	22.85	0.381
S9	3.39	1.085	18	3.33	1.037	20	3.63	1.052	19	2.92	0.954	20	3.42	1.057	19	11.34	***0.011***
S10	1.78	0.841	27	1.82	0.809	28	1.86	0.794	29	1.44	0.651	27	1.80	0.801	28	3.90	0.115
S11	4.37	1.123	15	4.42	1.089	15	4.68	1.069	14	3.92	0.954	16	4.47	1.091	15	18.50	***** ***0.010***
S12	1.67	0.701	30	1.81	0.793	29	1.89	0.814	28	1.40	0.645	28	1.79	0.786	29	3.78	***** ***0.026***
S13	4.22	1.114	17	4.13	1.025	17	4.42	1.089	17	3.76	0.831	17	4.22	1.061	17	16.83	***** ***0.017***
S14	4.37	1.040	14	4.65	1.074	13	4.77	1.041	13	4.16	1.028	13	4.62	1.065	13	19.56	***** ***0.021***
S15	4.98	0.830	11	4.93	0.714	12	5.04	0.744	12	4.72	0.678	12	4.96	0.741	12	22.23	0.234
S16	1.72	0.750	29	1.90	0.791	26	1.99	0.779	27	1.48	0.714	26	1.88	0.785	27	4.09	***** ***0.011***
S17	2.70	0.785	23	2.61	0.773	24	2.77	0.810	24	2.24	0.523	25	2.65	0.782	24	7.90	***** ***0.017***
S18	3.33	0.967	19	3.51	1.006	18	3.72	1.029	18	3.08	0.909	18	3.53	1.015	18	12.25	***** ***0.011***
S19	1.78	0.728	28	1.79	0.783	30	1.85	0.769	30	1.32	0.557	29	1.77	0.764	30	3.61	***** ***0.018***
S20	5.07	0.772	5	5.11	0.730	4	5.26	0.659	4	4.80	0.645	6	5.14	0.713	4	23.53	***** ***0.019***
S21	5.04	0.893	6	5.07	0.755	6	5.13	0.723	9	4.80	0.764	7	5.06	0.766	6	23.01	0.284
S22	1.80	0.806	26	1.87	0.771	27	2.06	0.731	26	1.32	0.557	30	1.89	0.769	26	4.17	***** ***0.000***
S23	3.30	1.030	20	3.36	1.033	19	3.44	1.067	20	3.00	1.041	19	3.35	1.047	20	11.32	0.288
S24	2.74	0.773	22	2.77	0.801	22	2.98	0.784	21	2.32	0.557	24	2.81	0.791	21	8.82	***** ***0.001***
S25	2.61	0.745	24	2.60	0.765	25	2.68	0.794	25	2.40	0.707	23	2.61	0.769	25	7.76	0.425
S26	5.07	0.772	4	5.13	0.706	3	5.26	0.659	3	4.80	0.645	5	5.14	0.703	3	23.58	***** ***0.018***
S27	4.39	1.000	13	4.42	1.011	16	4.50	1.036	16	4.00	1.041	15	4.41	1.024	16	18.12	0.175
S28	5.07	0.772	3	5.13	0.706	2	5.26	0.659	2	4.80	0.645	4	5.14	0.703	2	23.58	***** ***0.018***
S29	5.09	0.755	2	5.08	0.702	5	5.14	0.719	5	4.76	0.779	8	5.08	0.725	5	23.14	0.122
S30	5.09	0.812	1	5.19	0.714	1	5.31	0.636	1	4.84	0.624	1	5.19	0.704	1	23.93	***** ***0.012***

5.3 Development of Fuzzy Membership Function (Fuzzy-MBF)

The parameters 'a' and 'b' are used to obtain the MBF for various levels of significance for each risk factor. The data used for developing the MBF of the likelihood occurrence risk factors impact on cost overrun (Table 9). The MBF for the development stage of likelihood occurrence is shown in Fig. 7.

Table 9: Membership function (MBF) for likelihood occurrence and risk impact on cost overrun

Stage	Risk factor	Average mean, (a)	Standard deviation (b)	*a – 2b*	*a - b*	*a + b*	*a + 2b*
MBF for Likelihood Occurrence							
Feasibility study	F1: Inproper justification of cost benefit analysis	3.6	0.94	1.72	2.66	4.54	5.48
	F2: Too narrow focus on the technical IT issues	3.249	0.886	1.477	2.363	4.135	5.02
	F3: Overlooked the management and business impact issues	3.47	1.06	1.35	2.41	4.53	5.59
	F4: Wrong justification of investment alternatives and opportunity cost	2.28	0.835	0.61	1.445	3.115	3.95
	F5: Inappropriate technology chosen from the feasibility study	2.44	0.862	0.716	1.578	3.302	4.164
Project Planning stage	P1: Unclear project scope & objectives	4.17	1.07	2.03	3.1	5.24	6.31
	P2: Undefined project success criteria	3.69	0.829	2.032	2.861	4.519	5.348
	P3: Lack of quality control procedure and mechanism	2.72	0.964	0.792	1.756	3.684	4.648
	P4: Project milestones for stages not well established	2.41	0.829	0.752	1.581	3.239	4.068
	P5: Improper change management plan	3.56	0.976	1.608	2.584	4.536	5.512
	P6: Inaccurate estimate of resources	4.44	0.854	2.732	3.586	5.294	6.148
	P7: Unrealistic project schedule	4.41	0.784	2.842	3.626	5.194	5.978
	P8: Inadequate detail work breakdown structure	2.56	0.986	0.588	1.574	3.546	4.532
	P9: Critical & non-critical activities of project not identified	3.73	1.019	1.692	2.711	4.749	5.768
	P10: Project management & development team not properly set up	1.9	0.94	0.02	0.96	2.84	3.78
	P11: Unclear line of decision making authority throughout project	3.99	0.897	2.196	3.093	4.887	5.784
	P12: Lack on contingency plan/back up	3.66	0.615	2.43	3.045	4.275	4.89
	P13: System conversion method not well planned	3.325	0.94	1.445	2.385	4.265	5.205
	P14: Improper planning of timeframe for project reviews and updating	3.187	0.821	1.545	2.366	4.008	4.829
Requirement stage	R1: Unclear & inadequate identification of systems requirements	4.08	0.767	2.546	3.313	4.847	5.614
	R2: Incorrect systems requirements	2.3	1.1	0.1	1.2	3.4	4.5
	R3: Misinterpretations of systems requirements	3.32	0.852	1.616	2.468	4.172	5.024
	R4: Conflicting system requirements	2.44	0.986	0.468	1.454	3.426	4.412
	R5: Gold plating or unnecessary functions and requirements	2.05	1.02	0.01	1.03	3.07	4.09
	R6: Inadequate validation of systems requirements	3.82	0.974	1.872	2.846	4.794	5.768
	R7: Lack of users involvement in requirement stage	3.41	1.658	0.094	1.752	5.068	6.726
Development stage	D1: Improper handover from the requirement team	3.168	1.06	1.048	2.108	4.228	5.288
	D2: Inappropriate development methodology used	4.16	0.854	2.452	3.306	5.014	5.868
	D3: Unsuitable working model and prototype	2.53	0.83	0.87	1.7	3.36	4.19
	D4: Programming language and CASE tool selected not adequate	2.51	1.139	0.232	1.371	3.649	4.788
	D5: High level of technical complexities	3.79	0.944	1.902	2.846	4.734	5.678
	D6: Project involves the use of new technology	2.34	1.036	0.268	1.304	3.376	4.412
	D7: Difficulty in defining the input and output of system	3.127	1.174	0.779	1.953	4.301	5.475
	D8: Immature technology	2.22	0.887	0.446	1.333	3.107	3.994
	D9: Technological advancements and changes	2.06	0.819	0.422	1.241	2.879	3.698
	D10: Failures and inconsistencies of unit/modules test results	2.25	0.975	0.3	1.275	3.225	4.2
	D11: Failure of user acceptance test	3.21	0.937	1.336	2.273	4.147	5.084
	D12: Time consuming for testing	3.69	0.849	1.992	2.841	4.539	5.388
	D13: Resources shifted from project during development due to organisational priorities	3.287	1.034	1.219	2.253	4.321	5.355
	D14: Changes in management of organisation during development	2.02	1.005	0.01	1.015	3.025	4.03
	D15: Lack of users involvement and commitment	3.92	1.139	1.642	2.781	5.059	6.198
	D17: Ineffective communication within development team members	4.3	0.9	2.5	3.4	5.2	6.1
	D21: Inexperienced team members	2.18	1.079	0.022	1.101	3.259	4.338
	D22: Lack of commitment to project among development team members	2.9	1.036	0.828	1.864	3.936	4.972
	D23: Ineffective and inexperienced project manager	4.02	0.721	2.578	3.299	4.741	5.462
MBF for Risk Impact on Cost Overrun							
Feasibility study	F1: Improper justification of cost benefit analysis	3.24	1.208	0.824	2.032	4.448	5.656
	F2: Too narrow focus on the technical IT issues	2.58	0.867	0.846	1.713	3.447	4.314
	F3: Overlooked the management and business impact issues	3.44	0.995	1.45	2.445	4.435	5.43
	F4: Wrong justification of investment alternatives and opportunity cost	2.29	0.977	0.336	1.313	3.267	4.244
	F5: Inappropriate technology chosen from the feasibility study	2.02	0.71	0.6	1.31	2.73	3.44
Project Planning stage	P1: Unclear project scope & objectives	4.33	0.572	3.186	3.758	4.902	5.474
	P2: Undefined project success criteria	3.67	0.809	2.052	2.861	4.479	5.288
	P3: Lack of quality control procedure and mechanism	2.56	0.854	0.852	1.706	3.414	4.268
	P4: Project milestones for stages not well established	3.138	1.029	1.08	2.109	4.167	5.196
	P5: Improper change management plan	3.82	0.818	2.184	3.002	4.638	5.456
	P6: Inaccurate estimate of resources	4.24	0.886	2.468	3.354	5.126	6.012
	P7: Unrealistic project schedule	4.32	0.776	2.768	3.544	5.096	5.872
	P8: Inadequate detail work breakdown structure	2.29	0.978	0.334	1.312	3.268	4.246
	P9: Critical & non-critical activities of project not identified	3.53	0.884	1.762	2.646	4.414	5.298
	P10: Project management & development team not properly set up	1.99	0.978	0.034	1.012	2.968	3.946
	P11: Unclear line of decision making authority throughout project	3.043	1.131	0.781	1.912	4.174	5.305
	P12: Lack on contingency plan/back up	4.21	1.019	2.172	3.191	5.229	6.248
	P13: System conversion method not well planned	2.1	0.783	0.534	1.317	2.883	3.666
	P14: Improper planning of timeframe for project reviews and updating	3.27	1.516	0.238	1.754	4.786	6.302
Requirement stage	R1: Unclear & inadequate identification of systems requirements	3.99	0.867	2.256	3.123	4.857	5.724
	R2: Incorrect systems requirements	2.38	0.98	0.42	1.4	3.36	4.34
	R3: Misinterpretations of systems requirements	3.89	0.827	2.236	3.063	4.717	5.544
	R4: Conflicting system requirements	1.98	0.684	0.612	1.296	2.664	3.348
	R5: Gold plating or unnecessary functions and requirements	1.91	0.899	0.112	1.011	2.809	3.708
	R6: Inadequate validation of systems requirements	3.84	1.456	0.928	2.384	5.296	6.752
	R7: Lack of users involvement in requirement stage	3.241	0.98	1.281	2.261	4.221	5.201
Development stage	D1: Improper handover from the requirement team	3.19	1.366	0.458	1.824	4.556	5.922
	D2: Inappropriate development methodology used	3.62	1.102	1.416	2.518	4.722	5.824
	D3: Unsuitable working model and prototype	2.29	0.983	0.324	1.307	3.273	4.256
	D4: Programming language and CASE tool selected not adequate	2.54	1.133	0.274	1.407	3.673	4.806
	D5: High level of technical complexities	3.28	1.113	1.054	2.167	4.393	5.506
	D6: Project involves the use of new technology	2.67	0.973	0.724	1.697	3.643	4.616
	D7: Difficulty in defining the input and output of system	2.31	0.968	0.374	1.342	3.278	4.246
	D8: Immature technology	2.06	0.825	0.41	1.235	2.885	3.71
	D9: Technological advancements and changes	2.16	1.02	0.12	1.14	3.18	4.2
	D10: Failures and inconsistencies of unit/modules test results	2.29	0.983	0.324	1.307	3.273	4.256
	D11: Failure of user acceptance test	4.0	1.452	1.096	2.548	5.452	6.904
	D12: Time consuming for testing	2.43	0.854	0.722	1.576	3.284	4.138
	D13: Resources shifted from project during development due to organisational priorities	3.141	1.056	1.029	2.085	4.197	5.253
	D14: Changes in management of organisation during development	2.14	1.068	0.004	1.072	3.208	4.276
	D15: Lack of users involvement and commitment	3.90	1.182	1.536	2.718	5.082	6.264
	D17: Ineffective communication within development team members	3.12	1.458	0.204	1.662	4.578	6.036
	D19: Inadequately trained development team members	1.95	0.751	0.448	1.199	2.701	3.452
	D22: Lack of commitment to project among development team members	2.55	1.116	0.318	1.434	3.666	4.782
	D23: Ineffective and inexperienced project manager	3.94	0.766	2.408	3.174	4.706	5.472

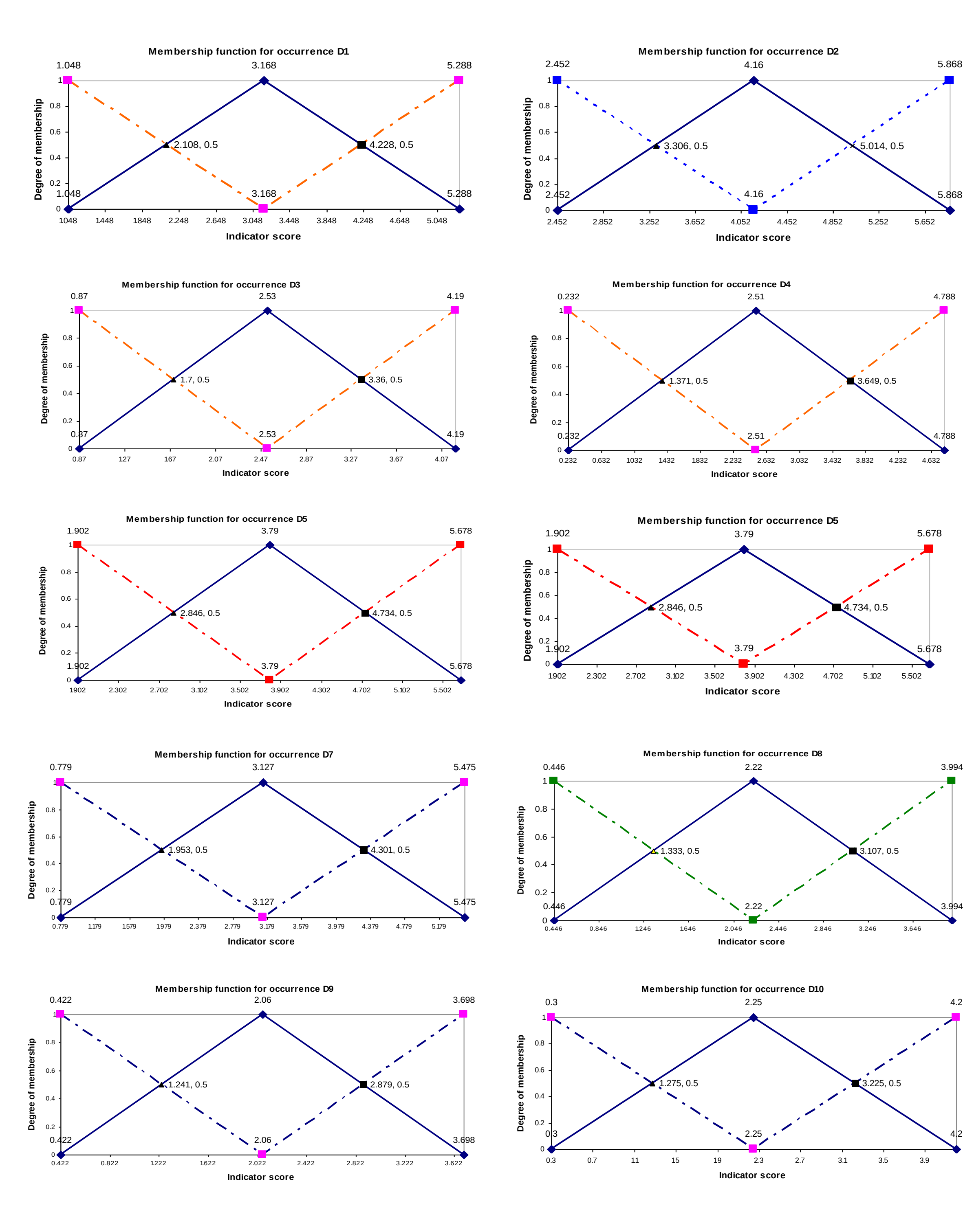

Fig. 7: Membership function for the development stage of likelihood occurrence

The range of fuzzy combinations and scenarios for each risk factor is presented in Fig. 8. Each scenario has several alternatives depending on the degree of membership and level of alpha cuts of the selected combination. The MBF for each risk factor is calculated based on the possible combinations and weights scored from the fuzzy computation. While weights can be expressed in either numeric (crisp) or linguistics (fuzzy) terms, all the weights must be defined in the same manner. As a result, the fuzzy weighted average used in this work is extracted using the *linguistic weights*. The combination of MBF for risk factors and MBF for degree of significance is used to develop the model. The fuzzy computation for the combination of risk factors is shown in Table 10 and the summation of fuzzy computations for each stage is listed in Table 11.

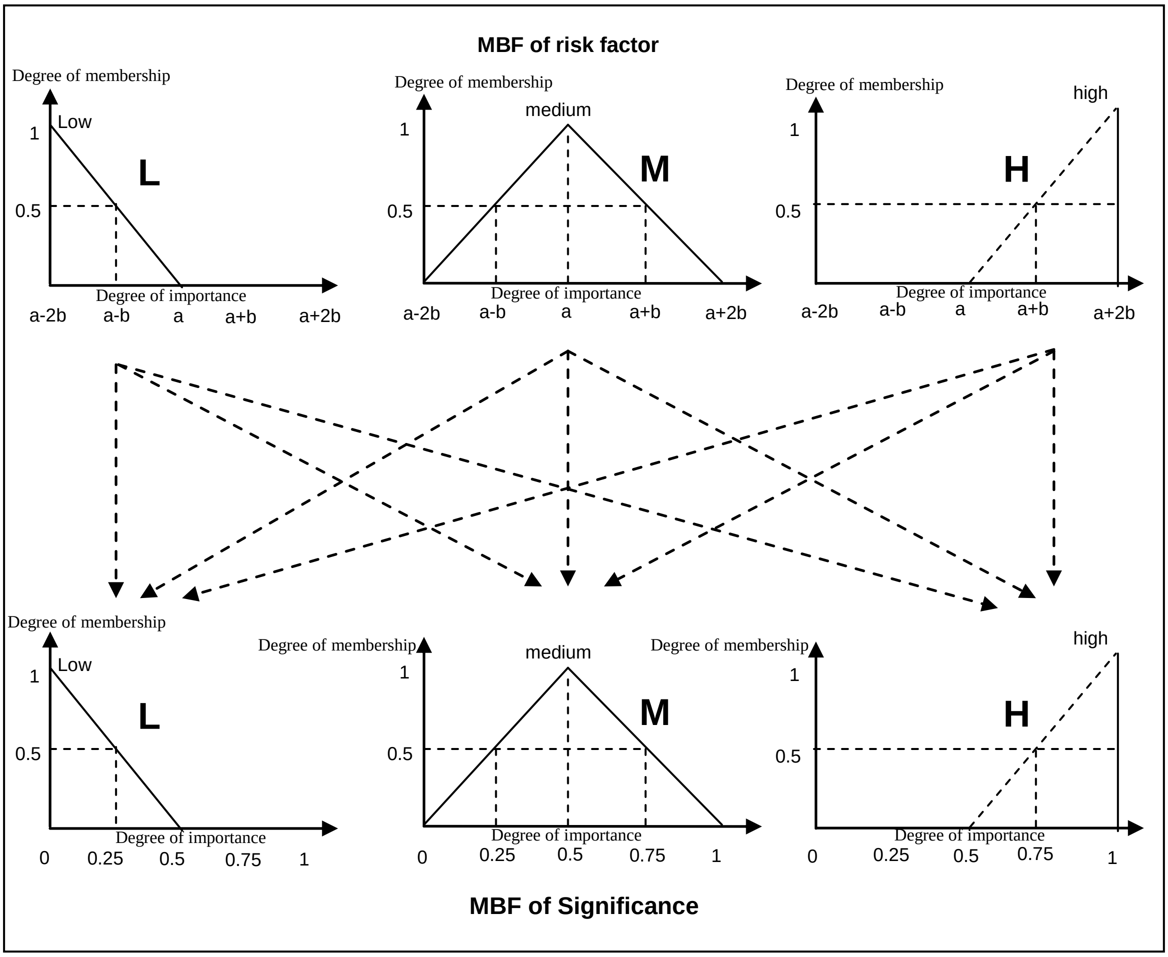

Fig. 8: Possible fuzzy computation combinations in Fuzzy-MBF

Table 10: Fuzzy computation for Feasibility stage of likelihood occurrence

alpha cuts	Low			Medium			High		
	Belief (W)	Membership score, x	WX	Belief (W)	Membership score, x	WX	Belief (W)	Membership score, x	WX
F1									
alpha cut : 0	0.3	3.02	0.906	0.4	4.72	1.888	0.8	5.12	4.096
alphs cut : 0.5	0.4	2.85	1.14	0.5	4.54	2.27	0.7	4.92	3.444
alpha cut : 1	0.5	2.66	1.33	0.8	4	3.2	0.6	4.72	2.832
	1.2	Sum WX	**3.376**	**1.7**	Sum WX	**7.358**	**2.1**	Sum WX	**10.372**
F2									
alpha cut : 0	0.2	2.87	0.574	0.4	4.27	1.708	0.6	4.3	2.58
alphs cut : 0.5	0.3	2.67	0.801	0.4	4.27	1.708	0.7	4.5	3.15
alpha cut : 1	0.4	2.5	1	0.7	3.8	2.66	0.7	4.5	3.15
	0.9	Sum WX	**2.375**	**1.5**	Sum WX	**6.076**	**2**	Sum WX	**8.88**
F3									
alpha cut : 0	0.3	2.8	0.84	0.4	4.75	1.9	0.8	5.15	4.12
alphs cut : 0.5	0.4	2.6	1.04	0.5	4.53	2.265	0.7	4.95	3.465
alpha cut : 1	0.5	2.41	1.205	0.8	3.9	3.12	0.6	4.75	2.85
	1.2	Sum WX	**3.085**	**1.7**	Sum WX	**7.285**	**2.1**	Sum WX	**10.435**
F4									
alpha cut : 0	0.2	1.91	0.382	0.3	3.45	1.035	0.7	3.45	2.415
alphs cut : 0.5	0.3	1.75	0.525	0.5	3.115	1.5575	0.6	3.25	1.95
alpha cut : 1	0.4	1.61	0.644	0.6	2.95	1.77	0.5	3.115	1.5575
	0.9	Sum WX	**1.551**	**1.4**	Sum WX	**4.3625**	**1.8**	Sum WX	**5.9225**
F5									
alpha cut : 0	0.2	2.1	0.42	0.3	3.65	1.095	0.7	3.65	2.555
alphs cut : 0.5	0.3	1.9	0.57	0.5	3.302	1.651	0.6	3.5	2.1
alpha cut : 1	0.4	1.72	0.688	0.6	3.2	1.92	0.5	3.302	1.651
	0.9	Sum WX	**1.678**	**1.4**	Sum WX	**4.666**	**1.8**	Sum WX	**6.306**
	Sum WX = 12.065 **Sum W = 5.1** **Sum WX / Sum W = 2.37**			**Sum WX = 29.748** **Sum W = 7.7** **Sum WX / Sum W = 3.86**			**Sum WX = 41.916** **Sum W = 9.8** **Sum WX / Sum W = 4.28**		

The combination of different possibilities and scenarios of the MBF are calculated for each risk factor using the different significance levels of linguistic variables. The low, moderate, and high-level of risk factors for each stage are calculated with the summation of the alternatives (as presented on the spider net diagrams in Fig. 9 and Fig. 10; the axis uses the radius of circles as the measurement scale). The points for each stage are connected, forming the looped line around the diagram. These connected lines represent the likelihood occurrence and risk impact on cost overrun. The $W*F(x)$ for each risk factor was calculated and the summation of $W*F(x)$ was divided for the summation of the weights (W).

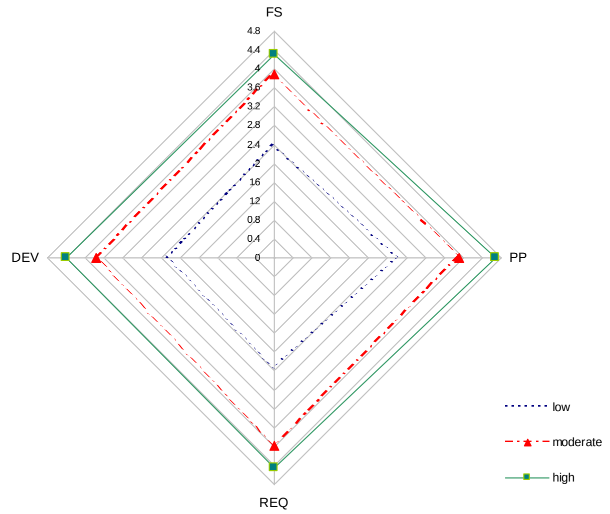

Fig 9: Level of likelihood occurrence for software project risk for each stage

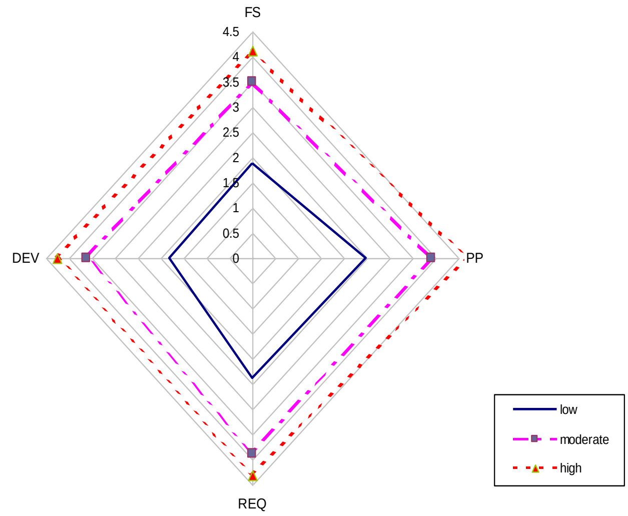

Fig 10: Level of risk impact on cost overrun for each stage

Table 11: Summation of fuzzy computation for each stage.

Stage	Low		Moderate		High	
	Likelihood occurrence	Impact of risk factor on cost overrun	Likelihood occurrence	Impact of risk factor on cost overrun	Likelihood occurrence	Impact of risk factor on cost overrun
Feasibility study	2.37	1.89	3.86	3.49	4.28	4.13
Project Planning	2.58	2.47	3.9	3.9	4.66	4.67
Requirement	2.33	2.38	3.96	3.87	4.43	4.32
Development	2.25	1.82	3.76	3.63	4.37	4.26

6.0 DISCUSSIONS

Both ANOVA analysis and Post Hoc test reveal significant difference between the IT staff group and the other three groups. The five most effective strategies in reducing the software development risk are: users' involvement (S30), good project management (S29), lines of communications (S28), planning of resources (S26) and developed a clear systems requirements (S20), which confirm the views expressed by [52] on the pivotal role of users' involvement in the development of software project. However, authors in [31, 53] argued that it is difficult to predict the users' expectations on the timeframe and budget allocated. Bannerman in [33] stressed that absence of effective governance resulted in risk exposures in these areas, i.e., clarity and relevance of objectives, scope and requirements; provide guidance, direction and a common sense of purpose. Communication issues among human beings were relatively complex and unpredictable. Jiang & Klein in [42] stressed that, as a result of poor communications among development team members, much of the time might be spent on duplication of efforts and progress towards individual's goal rather than the project goal. Bannerman [33] also suggested that the integration of teams has lead to greatly improved project communication, interaction, issue resolution and progress tracking. These views collaborate with the findings in Table 3, for example, S1, S4, S5, S7 and S8 were rated as highly effective strategies. These risk management strategies include defining clear objectives, identifying critical activities, specifying the project success criteria, consistent commitment from management and the lessons learned from past software projects. For a successful software project development, considerable time is usually spent on the planning phase but sometimes it takes longer completion time as compared to the rest of the project. It is clear that in light of existing approaches to assess risks in the software development life cycle, fuzzy modelling has an important role to play. It addresses some of the gaps in the application of other methods by reducing subjectivity often found in qualitative approaches in supporting quantitative assessment of risks. Usually, incomplete project information is available during the very early phases of the project and many decision making processes occur in an environment in which the goals, constraints and consequences of possible actions are not precisely known. Fuzzy measures could improve the decision making process even there is lack of information available.

7.0 CONTRIBUTIONS AND NOVELTIES

This article is the first academic paper that develops the Fuzzy membership function for likelihood occurrence and Risk Impact on Cost Overrun for IT projects. To the best of our knowledge, we are the first to demonstrate how fuzzy modelling can aid identification of risk factors at the early stage of project. Specifically, the subjectivity was transferred to a fuzzy membership function so as to improve visualization of comparisons between factors and interpretations of emerging factors. The fuzzy computation of various combinations can clearly assist the IT practitioners in formalizing and assessing risk environments in real-time and improve the decision making process over the entire software development life cycle by systematizing the process and improving visualization opportunities.

The results of this study do not agree with Shenhar and Dvir's [54] suggestion that project management is unnecessary for software development but are in line with the findings of Bannerman [33] that good project management is necessary yet not essential in mitigating software development risks. Our studies suggest that non-technical strategies are more effective than technical approaches for software development risks. Analysis results show risk management strategies related to users' involvement, project planning & control and communication skills are influential in reducing risks. Lessons learned from the past software projects are highly effective, thus organizational learning techniques might be considered for risk prevention in software development projects.

8.0 CONCLUSIONS

The developed Fuzzy-MBF offers a quantitative evaluation of risk factors and provides a systemic evaluation of risk and visualization of results. The fuzzy computation of various combinations can assist IT professionals in formalizing and assessing the operational risk environments and thereby improving decision making process during the software development life cycle. This work demonstrates the robust results possible with fuzzy modelling and how it is a better fit with practices and linguistic representations of risk amongst managers. Although it is useful to see these strategies from the point of view in understanding the approaches to a risk situation, it might not be wise to pigeonhole any practical approaches into one or more categories in an exclusive way. Therefore, it is expected to be clarified in a more sophisticated factor deduction or clustering research in future. The development of risk factors could be improved in future research by using partial least squares regression (PLS) to create a model which allows the projection of predicted variables and observable variables to different spaces which could lead to improved visualization in Fuzzy-MBF approaches.

REFERENCES

[1] R. Capilla, J. Bosch, P. Trinidad, A. Ruiz-Cortés, M. Hinchey, "An overview of Dynamic Software Product Line architectures and techniques: Observations from research and industry". *Journal of Systems and Software*, Vol. 91, 2014, pp. 3-23.

[2] L. Willcocks, M. Griffiths, *Information system at work: people, politics and technology*. McGraw Hill, London, 1997.

[3] L. Hedelin, C.M. Allwood, "IT and strategic decision making". *Industrial Management & Data Systems,* Vol. 102, No. 3, 2002, pp. 125-139.

[4] D. Baccarini, G. Salm, P. Surrey, "Management of risks in Information Technology projects". *Journal of Industrial Management & Data Systems,* Vol. 104, No. 4, 2004, pp. 286-295.

[5] S.J. Huang, W.M. Han, "Exploring the relationship between software project duration and risk exposure: a cluster analysis". *Information and management,* Vol. 45, 2008, pp. 175-182.

[6] R. Giuffrida, Y. Dittrich, "Empirical studies on the use of social software in global software development: A systematic mapping study". *Information and Software Technology,* Vol. 55, No. 7, 2013, pp. 1143-1164.

[7] A.M. Magdaleno, C.M.L. Werner, R.M. de Araujo, "Reconciling software development models: A quasi-systematic review". *Journal of Systems and Software,* Vol. 85, No. 2, 2012, pp. 351-369.

[8] Y. Martín, J.C. Yelmo, "Guidance for the Development of Accessibility Evaluation Tools Following the Unified Software Development Process". *Procedia Computer Science,* Vol. 27, 2014, pp. 302-311.

[9] R.N. Charette, *Software engineering risk analysis and management*. McGram-Hill, NY, 1989.

[10] P. Marina, J. Simmonds, H. Astudillo, "Semi-automated Tool Recommender for Software Development Processes". *Electronic Notes in Theoretical Computer Science,* Vol. 302, No. 25, 2014, pp. 95-109.

[11] B. Boehm, "Software risk management: principles and practices". *IEEE Software,* Vol. 8, No. 1, 1991, pp. 32-41.

[12] C. Chen, H. Cheng, "A comprehensive model for selecting information system project under fuzzy environment". *International Journal of Project Management*, Vol. 27, 2009, pp. 389-399.

[13] L.A. Zadeh, "Fuzzy logic, neural networks, and soft computing". *Communications of the ACM,* Vol. 37, No. 3, 1994, pp. 77-84.

[14] G. Büyüközkan, D. Ruan, "Choquet integral based aggregation approach to software development risk assessment". *Information Sciences*, Vol. 180, No. 3, 2010, pp. 441-451.

[15] M. Jorgensen, "Identification of more risks can lead to increased over-optimism of and over-confidence in software development effort estimates". *Information and Software Technology*, Vol. 52, No. 5, 2010, pp. 506-516.

[16] J. Portillo-Rodríguez, A. Vizcaíno, M. Piattini, S. Beecham, "Using agents to manage Socio-Technical Congruence in a Global Software Engineering project". *Information Sciences,* Vol. 264, No. 20, 2014, pp. 230-259.

[17] R. Capilla, J. Bosch, P. Trinidad, A. Ruiz-Cortés, M. Hinchey, "An overview of Dynamic Software Product Line architectures and techniques: Observations from research and industry". *Journal of Systems and Software,* Vol. 91, 2014, pp. 3-23.

[18] T. Thüm, C. Kastner, F. Benduhn, J. Meinicke, G. Saake, T. Leich, "FeatureIDE: An extensible framework for feature-oriented software development". *Science of Computer Programming*, Vol. 79, No. 1, 2014, pp. 70-85.

[19] S. Adolph, P. Kruchten, W. Hall, "Reconciling perspectives: A grounded theory of how people manage the process of software development". *Journal of Systems and Software,* Vol. 85, No. 6, 2012, pp. 1269-1286.

[20] A. Engel, M. Last, "Modeling software testing costs and risks using fuzzy logic paradigm". *Journal of Systems and Software*, Vol. 80, No. 6, 2007, pp. 817-835.

[21] F. Reyes, N. Cerpa, A. Candia-Véjar, M. Bardeen, "The optimization of success probability for software projects using genetic algorithms". *Journal of Systems and Software,* Vol. 84, No. 5, 2011, pp. 775-785.

[22] E.H. Conrow, P.S. Shishido, "Implementing risk management on software intensive projects". *IEEE Software,* Vol. 14, No. 3, 1997, pp. 83-89.

[23] R. Fairley, "Risk management for software projects". *IEEE Software,* Vol. 11, No. 3, 1994, pp. 57-67.

[24] F.J. Heemstra, R.J. Kusters, "Dealing with risk: A practical approach". *Journal of Information Technology,* Vol. 11, No. 4, 1996, pp. 333-346.

[25] T. Dingsoyr, S. Nerur, V. Balijepally, N.B. Moe, "A decade of agile methodologies: Towards explaining agile software development". *Journal of Systems and Software,* Vol. 85, No. 6, 2012, pp. 1213-1221.

[26] D. Stahl, J. Bosch, "Modeling continuous integration practice differences in industry software development". *Journal of Systems and Software,* Vol. 87, 2014, pp. 48-59.

[27] J.C. Bennett, E.M. Bohoris, R.C. Aspinwall, "Risk analysis techniques and their application to software development". *European Journal of Operational Research,* Vol. 95, 1996, pp. 467-475.

[28] R.K.J.R. Rainer, C.A. Snyder, H.H. Carr, "Risk analysis for information technology". *Journal of Management Information Systems,* Vol. 8, No. 1, 1991, pp. 129-147.

[29] H.R. Costa, M.O. Barros, G.H. Travassos, "Evaluating software project portfolio risks". *Journal of Systems & Software,* Vol. 80, No. 1, 2007, pp. 16-31.

[30] J. Iversen, L. Mathiassen, P.A. Neilsen, "Managing risks in software process improvement: an action research approach". *MIS Quarterly,* Vol. 28, No. 3, 2004, pp. 395-433.

[31] J. Jiang, G. Klein, R. Discenza, "Information system success as impacted by risks and development strategies". *IEEE Transactions on Engineering Management,* Vol. 48, No. 1, 2001, pp. 46-55.

[32] E.W.T. Ngai, F.K.T. Wat, "Fuzzy decision support system for risk analysis in e-commerce development". *Decision Support systems,* Vol. 40, No. 2, 2005, pp. 235-255

[33] P.L. Bannerman, "Risk and risk management in software projects: A reassessment". *Journal of Systems and Software,* Vol. 81, No. 12, 2008, pp. 2118-2133.

[34] T. Thum, C. Kastner, F. Benduhn, J. Meinicke, G. Saake, T. Leich, "FeatureIDE: An extensible framework for feature-oriented software development". *Science of Computer Programming,* Vol. 79, No. 1, 2014, pp. 70-85.

[35] C.W. Ibbs, Y.H. Kwak, "Assessing project management maturity". *Project Management Journal,* Vol. 31, No. 1, 2000, pp. 32-43.

[36] T.R. Adler, J.G. Leonard, R.K. Nordgen, "Improving risk management: Moving form risk elimination to risk avoidance". *Information & Software Technology,* Vol. 41, No. 1, 1999, pp. 29-34.

[37] M. Barros, C. Werner, G. Travassos, "Supporting risks in software project management". *Journal of Systems & Software,* Vol. 70, 2004, pp. 21-35.

[38] N. Fenton, P. Krause, M. Neil, "Software measurement: Uncertainty and causal modelling". *IEEE Software,* Vol. 19, No. 4, 2002, pp. 116-122.

[39] S. Drew, "Reducing enterprise risk with effective threat management". *Information Security Management,* Vol. 13, No. 6, 2005, pp. 37-42.

[40] S. Pender, "Managing Incomplete Knowledge: Why risk management is not sufficient". *International Journal of Project Management*, Vol. 19, No. 2, 2001, pp. 79-87.

[41] H.F. Cervone, "Project risk management". *OCLC Systems and Services: International digital library perspectives*, Vol. 22, No. 4, 2006, pp. 256-262.

[42] J. Jiang, G. Klein, "Software development risks to project effectiveness". *Journal of Systems & Software,* Vol. 52, 2000, pp. 3-10.

[43] R.C. Mahaney, A.L. Lederer, "The role of monitoring and shirking in information systems project management". *International Journal of Project Management,* Vol. 28, No. 1, 2010, pp. 14-25.

[44] L.A. Zadeh, "Fuzzy sets. Information Control", Vol. 8, 1965, pp. 338-353.

[45] A. Vakili-Ardebili, A.H. Boussabaine, "Application of fuzzy techniques to develop an assessment framework for building design eco-drivers". *Building and Environment*, Vol. 42, 2007, pp. 3785-3800.

[46] H.M. Lee, "Generalization of the group decision making using fuzzy sets theory for evaluating the rate of aggregative risk in software development". *Information Sciences,* Vol. 113, No. 3–4, 1999, pp. 301-311.

[47] H.M. Lee, T.Y. Lee, J.J. Chen, "A new algorithm for applying fuzzy set theory to evaluate the rate of aggregative risk in software development". *Information Sciences,* Vol. 153, 2003, pp. 177-197.

[48] Z. Xu, T.M. Khoshgoftaar, E.B. Allen, "Application of fuzzy expert systems in assessing operational risk of software". *Information and Software Technology*, Vol. 45, 2003, pp. 373-388.

[49] E.W.T. Ngai, F.K.T. Wat, "Fuzzy decision support system for risk analysis in e-commerce development". *Decision Support systems*, Vol. 40, No. 2, 2005, pp. 235-255.

[50] S.J. Chen, C.L. Hang, *Fuzzy Multiple Attribute Decision-Making: Methods and Applications*. Springer-Verlag, Berlin, 1992.

[51] W. Dong, F. Wong, "Fuzzy computation and risk in decision analysis". *Civil Engineering systems,* Vol. 2, 1985, pp. 201-208

[52] S.J. Su, C. Chan, Y.J. Liu, H. Chen, "The impact of user review on software responsiveness: moderating requirements uncertainty". *Information and Management,* Vol. 45, 2008, pp. 203-210.

[53] H. Barki, S. Rivard, J. Talbot, "An intergrated contingency model of software project risk management". *Journal of Management Information Systems,* Vol. 17, No. 4, 2001, pp. 37-69.

[54] A.J. Shenhar, D. Dvir, "Project management research: the challenge and opportunity". *Project Management Journal,* Vol. 38, No. 2, 2007, pp. 93-99.