Managing Software Project Risks (Analysis Phase) with Proposed Fuzzy Regression Analysis Modelling Techniques with Fuzzy Concepts

Abdelrafe Elzamly[1,2] and Burairah Hussin[1]

[1] Information and Communication Technology, Tehnical University of Malaysia Melaka (UTeM)
[2] Department of Computer Science, Faculty of Applied Sciences, Al-Aqsa University, Gaza, Palestine

The aim of this paper is to propose new mining techniques by which we can study the impact of different risk management techniques and different software risk factors on software analysis development projects. The new mining technique uses the fuzzy multiple regression analysis techniques with fuzzy concepts to manage the software risks in a software project and mitigating risk with software process improvement. Top ten software risk factors in analysis phase and thirty risk management techniques were presented to respondents. The results show that all software risks in software projects were very important from software project manager perspective, whereas all risk management techniques are used most of the time, and often. However, these mining tests were performed using fuzzy multiple regression analysis techniques to compare the risk management techniques with each of the software risk factors to determine if they are effective in reducing the occurrence of each software risk factor. The study has been conducted on a group of software project managers. Successful software project risk management will greatly improve the probability of software project success.

Keywords: software risk management, analysis phase, software risk factors, risk management techniques, correlation analysis, fuzzy regression analysis techniques with fuzzy concepts, mining techniques, coefficient of determination

1. Introduction

Despite much research and progress in the area of software project management, software development projects still fail to deliver acceptable systems on time and within budget. Much of the failure could be avoided by managers' pro-active maintenance and dealing with risk factors rather than waiting for problems to occur and then trying to react. Project management and risk management have been proposed as a solution to preserve the quality and integrity of a project by reducing cost escalation (Schwalbe, 2010). Due to the involvement of risk management in monitoring the success of a software project, analyzing potential risks, and making decisions about what to do about potential risks, the risk management is consideredthe planned control of risk. Integrating formal risk management with project management is a new phenomenon in software engineering and product management community. In addition to that, risk is an uncertainty that can have a negative or positive effect on meeting project objectives. Risk management is the process of identifying, analyzing and controlling risk throughout the life of a project, to meet the project objectives (Schwalbe, 2010).

However, an intelligent performance analysis approach is adapted for decision making to select the optimization techniques to apply in real word problem solving approach, particularly related to industrial engineering problems (Vasant, 2013) The mining approach is a new way of identifying risk from data that creates relationships between data and finds the optimum result from them. This includes techniques such as simulation analysis, fuzzy logic models, fuzzy multiple regression, neural network models, genetic algorithm, and heuristic algorithm. However, the goal of risk management at early iden-

tification and recognition of risks and then actively changes the course of actions to mitigate and reduce the risk (Miler & Górski, 2002). In the process of understanding the factors that contribute to software project success, risk is becoming increasingly important. This is a result of the size, complexity and strategic importance of many of the information systems currently being developed. Today, we must think of risk as a part of software project lifecycle which is important for a software project survival (Pandian, 2007) On the other hand, risk management aims to read risks as improvement opportunities and provide inputs to growth plans (Pandian, 2007).

In our paper, we identified software risk factors and risk management techniques that guide software project managers to understand and mitigate risks in software analysis development projects. However, Software Development Life Cycle according to (Hoffer, George, & Valacich, 2011), is the process of creating or altering systems, and the models and methodologies that people use to develop these systems. Also, it includes these phases as follows (Hoffer et al., 2011): Planning, analysis, design, implementation, and maintenance. In addition to that, we focused on the analysis phase: It includes looking at any existing system to see what it is doing for the organization and how well that system is doing its job. According to Taylor, we should apply techniques consistently throughout the software project risk management process (J. Taylor, 2004). Risk management is a practice of controlling risk and the practice consists of processes, methods, and tools for managing risks in a software project before they become problems (Sodhi & Sodhi, 2001).

This study will guide software managers to apply software risk management practices with real world software development organizations and verify the effectiveness of the modelling techniques on a software project. We hope that the approaches will succeed in the software risk management methodology, which will improve the probability of software project success. **The objective of this study is:** To identify the software risk factors of software analysis projects in the Palestinian software development organizations, to rank the software risk factors according to their importance, severity and frequency, based on data source, to identify the

activities performed by software project managers to model and mitigate the identified software analysis project risks. The organization of this paper will be as follows. Section 2 presents an overview of the literature. Section 3 introduces the software risk factors (analysis phase) relevant for the study. Section 4 introduces the common risk management techniques for these software risks. Section 5 presents the empirical work. Section 6 concludes the article and glimpses on future work.

2. Literature Review

According to (H. Taylor, 2005), the key risks are identified by a group of Hong Kong project managers working for vendor IT firms who offered package implementation solutions, both locally and overseas. In that study a number of new risks from the vendor perspective have been identified, which indicate that vendor project managers typically have a broader focus on risks than their in-house counterparts. Addison & Vallabh (Addison & Vallabh, 2002) focused on the experienced project manager's perceptions of software project risks and controls. This work reports on the more significant risks and controls that are utilized to reduce the occurrence of the risk factors. The effectiveness of various controls to reduce the occurrence of risk factors was also identified and discussed. We improved quality of software projects of the participating companies while estimating the quality–affecting risks in IT software projects (Elzamly & Hussin, 2011a). The new technique used the chi-square ($\chi2$) test to control the risks in a software project (Khanfar et al., 2008). However, we also used new techniques the regression test and effect size test proposed to manage the risks in a software project and reducing risk with software process improvement (Elzamly & Hussin, 2011b). Furthermore, we used the new stepwise regression technique to manage the risks in a software project. These tests were performed using regression analysis to compare the controls with each of the risk factors to determine if they are effective in mitigating the occurrence of each risk factor implementation phase (Elzamly & Hussin, 2013). According to (Dash & Dash, 2010) risk management consists of the processes, methodologies and tools that are used to deal with risk

factors in the Software Development Life Cycle (SDLC) process of Software Project. Also, Oracle corporation described risk management solutions that enable a standardized approach for identifying, assessing and mitigating risk throughout the software project lifecycle (Oracle, 2010). Risk management methodology has five phases: Risk identification (planning, identification, prioritization), risk analysis and evaluation (risk analysis, risk evaluation), risk treatment, risk controlling, risk communication and documentation. These relied on three categories of techniques such as risk qualitative analysis, risk quantitative analysis and risk mining analysis throughout the life of a software project to meet the goals(Elzamly & Hussin, 2014).

Although there are many methods in software risk management, software development projects have a high rate of risk failure. Thus, if the complexity and the size of the software projects are increased, managing software development risk becomes more difficult (Hoodat and Rashidi, 2009). Additionally, the optimization method was tested with various software project risk prediction models that have been developed (Reyes, Cerpa, Candia, & Bardeen, 2011). There are several software risk management approaches, models, and framework according to a literature review, so these models and approaches are listed in this section. Furthermore (Bannerman, 2010), risk management approach practices need to be increased with extra analysis to identify, analyze and assess structural risks, to mitigate software risks and the delivery of software project quality. Büyüközkan and Ruan (2010) present incorporated multi-criteria to estimate the methodology for software managers to mitigate software risks. The method relied on a special fuzzy operator, namely a two-additive Choquet integral that enables modeling various effects of importance and interactions among software risks. In addition to that, Dhlamini et al. (2009) demonstrated the need for an intelligent risk assessment and management tool for either agile or traditional methods in a software development. Therefore, they proposed a model which could be investigated for use in developing intelligent software risk management tools (Dhlamini, Nhamu, & Kachepa, 2009). Finally, the approaches and methods reviewed above do not focus on modelling software risks based on quantitative and mining techniques for

predicting the reliability of a software project. Furthermore, there is no integration between the software development life cycle and the real software risk management phases based on techniques to manage software risks. Therefore, previous studies for approach in software risk management limited phases and techniques, thus they do not create the relation between software risk factors in software development life cycle and risk management techniques to mitigate risks. This study attempts to propose the modelling software risk management for successful software project, based on fuzzy regression analysis techniques with fuzzy concepts.

3. Top 10 Software Risk Factors in Analysis Phase

This study displays the top ten software risk factors in software development life cycle (analysis phase) that are common in the literature review based on 'top-ten' based on Boehm (1991), Miler (2005). The 'Top 10 software risk factors' lists differ to some extent from author to author, but some essential software risk factors that appear on almost any list can be distinguished. These factors need to be addressed and thereafter controlled. Consequently, we focus on the work by others, such as Elzamly and Hussin (2011); Aritua et al. (2010); Christopher Jones et al. (2010); Chen and Huang (2009); Hoodat and Rashidi (2009); Nakatsu and Iacovou (2009); Caper Jones's Risk (2008); Khanfar, Elzamly et al. (2008); Pare et al. (2008); Han and Huang (2007-2008); Aloini et al. (2007); Boehm's 10 risk items (2006-2007); Taimour (2005); Wallace et al. (2004); Shafer and Officer (2004); Lyons and Skitmore (2004); Addison (2003); Kweku (2003); Boehm's 10 risk items (2002); Addison and Vallabh (2002); Mark et al. (2002); Mitchell (2002); Schmidt et al. (2001); Houston et al. (2001); Lawrence et al. (2001); Sumner (2000); Mark et al. (1998); The Standish Group survey (1995); a survey of Boehm's 10 risk items in 1991 on software risk management; Boehm and Ross (1989); and many other researchers in software engineering to obtain software risk factors and risk management techniques. The next discussions consist of the 10 most serious risks of a software project ranked from one to

ten, each risk's status, and the plan for addressing each risk as recommended by researchers and experts when studying the software risk factors in software development life cycle. These software project risks are illustrated in Table 1:

4. Risk Management Techniques

Through reading the existing literature on software risk management, we listed thirty control factors that are considered important in reducing the software risk factors identified; these controls are:

C1: Using of requirements scrubbing, C2: Stabilizing requirements and specifications as early as possible, C3: Assessing cost and scheduling the impact of each change to requirements and specifications, C4: Develop prototyping and have the requirements reviewed by the client, C5: Developing and adhering a software project plan, C6: Implementing and following a communication plan, C7: Developing contingency plans to cope with staffing problems, C8: Assigning responsibilities to team members and rotating jobs, C9: Have team-building sessions, C10: Reviewing and communicating progress to date and setting objectives for the next phase, C11: Dividing the software project into controllable portions, C12: Reusable source code and interface methods, C13:Reusable test plans and test cases, C14: Reusable database and data

mining structures, C15: Reusable user documents early, C16: Implementing/Utilizing automated version control tools, C17: Implementing/utilizing benchmarking and tools of technical analysis, C18: Creating and analyzing process by simulation and modeling, C19: Providing scenarios methods and using of the reference checking, C20: Involving management during the entire software project lifecycle, C21: Including formal and periodic risk assessment, C22: Utilizing change control board and exercising quality change control practices, C23: Educating users on the impact of changes during the software project, C24: Ensuring that quality-factor deliverables and task analysis, C25: Avoiding having too many new functions on software projects, C26: Incremental development (deferring changes to later increments), C27: Combining internal evaluations by external reviews, C28: Maintaining proper documentation of each individual's work, C29: Providing training in the new technology and organizing domain knowledge training, C30: Participating of users during the entire software project lifecycle.

5. Empirical Strategy

Data collection was achieved through the use of a structured questionnaire and historical data

Dimension	No	Software risk factors	Frequency
Analysis	1	Unclear, incorrect, continually and rapidly changing software project requirements	19
	2	Failure to incomplete or missing detailed requirements analysis and specification documentation	9
	3	Developer software gold-plating	7
	4	Lack of software project/IT Management	6
	5	Software project requirements not adequately identified and mismatched	6
	6	Inadequate knowledge about tools and programming techniques	5
	7	Lack of traceability, confidentiality, correctness and inspection of the software project planning	4
	8	Major requirements change after software project plan phase	3
	9	Changing software project specifications	2
	10	Inadequate value analysis and a measurement software project to measure progress	2
		Total frequency	**63**

Table 1. Illustration of top ten software risk factors in software project based on researchers.

for assisting in estimating the quality of software through determining the risks that were common to the majority of software projects in the analyzed software companies. Top ten software risk factors and thirty risk management techniques were presented to respondents. The method of sample selection referred to as 'snowball' and distribution of personal regular sampling was used. This procedure is appropriate when members of homogeneous groups (such as software project managers, IT managers) are difficult to locate. The seventy six software project managers participated in this study. The project managements that participated in this survey came from specific, mainly software project managements in software development organizations.

The respondents were presented with various questions, which used scales 1-7. For presentation purposes in this paper and for effectiveness, the point scale was the following: For choices, being headed, 'unimportant' equals one and 'extremely important' equals seven. Similarly, seven frequency categories were scaled into 'never' equals one and 'always' equals seven. All questions in software risk factors were measured on a seven–point Likert scale from unimportant to extremely important and software control factors were measured on a seven–point Likert scale from never to always. Therefore, the more extreme categories were combined in a way such that seven-point scale were reduced to five-point scale as follows: A category called 'somewhat important' was created, combining the two ratings 'very slightly important' and 'slightly important'. Similarly, a category called 'very important' combined the two ratings 'very important' and 'extremely important'. Similarly, seven frequency categories were re-scaled into five sub-categories for presentation purposes. 'Rarely' combined the two ratings: 'rather seldom' and 'seldom'. 'Never', 'sometimes' and 'often' was unchanged, while 'most of the time', combined the two ratings: 'usually' and 'always'. However, to describe "Software Development Company in Palestine" having in-house development software and supplier of software for local or international market, we depended on Palestinian Information Technology Association (PITA) Members' webpage at PITA's website [www.pita.ps/ PITA 2012], Palestinian investment promotion agency [http://www.pipa.gov.ps/ PIPA 2012] to select top IT manager and software project managers. In order to mitigate risk, we can use qualitative, quantitative, and mining approaches. In this study, the mining approaches are proposed to manage software risk. In order to establish the mining approaches, first we need to model the relationship between software risks (analysis phase) and risk management techniques. The model that we propose a fuzzy multiple regression analysis to mitigate risks. Indeed, our approaches focused on identifying software risk factors, and risk management techniques in software development and how to manage and model the software risk factors with statistical and mining techniques.

5.1. Correlation Analysis

Clearly, the preceding analysis states that there are correlations between determining variables besides correlation between risk factors and all determining control factors (Rui-ge & Bing-rong, 2011). However, the equation of Correlation Coefficient is the following (Martin, Pasquier, M., & T., 2005; Marza & Seyyedi, 2009):

$$R = \frac{n[\sum(X_i \cdot Y_i)] - (\sum X_i)(\sum Y_i)}{\sqrt{[n(\sum X_i^2) - (\sum X_i)^2][n(\sum Y_i) - (\sum Y_i)]}} \quad (1)$$

5.2. Regression Analysis Model

Regression modeling is one of the most widely used statistical modeling techniques for fitting a response (dependent) variable as a function of predictor (independent) variables (Martin et al., 2005). Indeed, software risk factor is a dependent variable while control factors are independent variables. A linear equation between one and many independent variables (multiple regression) may be expressed as:

$$Y = b0 + b1x1 + b2x2 + \ldots + bnxn \quad (2)$$

where $b0, b1, b2, \ldots$ and bn are constants; $x1, x2, \ldots$ and xn are the independent variables, and y is the dependent variable. The values of $b0, b1, b2, \ldots$ and bn of the multiple regression equation may be obtained solving the next linear equations system (Martin et al., 2005).

5.3. Regression Model with Fuzzy Concepts

Fuzzy regression analysis is an extension of the classical regression analysis in which some elements of the models are represented by fuzzy numbers (Dom, Abidin, Kareem, Ismail, & Daud, 2012). Additionally, mining algorithms usually assume that the input data is precise and clean, which is unrealistic in practical situations (Chen & Weng, 2008). In other words, fuzzy multiple regression model in which response variable is a fuzzy variable and part of the covariates are crisp variables (Lin, Zhuang, & Huang, 2012). Therefore, fuzzy regression methods have considerably they are helpful or widened the field of application of classical regression methods in such a way that in determining regression relations from fuzzy initial data, which can be either quantitative and qualitative (Gu, Song, & Xiao, 2006). Also, the same authors explained, when one or some data points are greatly influenced by random factors, there must be a difference between the result gained and the actual values. At that time, the classical regression analysis is not applicable.

Although real data tends to be imprecise, no previous research has ever developed mining algorithms to find knowledge directly from imprecise data (Chen & Weng, 2008). However, it identifies the various data types that may appear in a questionnaire. Then, we introduce the questionnaire data mining problem and define the rule patterns that can be mined from questionnaire data. A unified approach is developed based on fuzzy techniques so that all different data types can be handled in a uniform manner (Chen & Weng, 2009). Therefore, in order to discover rules from a questionnaire dataset, we need a brand new approach that can deal with different data types occurring (Chen & Weng, 2009). Therefore the same authors explained that all data types could be represented and operated from fuzzy points of view. Furthermore, we must extend the crisp association rules to fuzzy association rules from questionnaire data.

5.4. Fuzzy Concepts with Membership Function

Fuzzy concepts help us to find the deviation of each data from fitness equation, so we define a normal distribution membership function as follows (Marza & Seyyedi, 2009):

$$U_i = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{y_i - \mu}{\sigma}\right)^2} \quad (3)$$

where μ is an average of sample points and σ is square root of variance math. If we add fuzzy domain to regression method, the effect of discrete data points on the fitness result will be reduced and the effect of concentrated data points on the fitness result will be enhanced. Indeed, a membership function is a curve that defines how each point in the input space is mapped to a membership value (or degree of membership) between 0 and 1 (Dom et al., 2012). Specifically, the simplest membership functions are formed using straight lines such as triangular membership function (specified by 3 parameters) and trapezoidal membership function (Aali, Parsinejad, & Rahmani, 2009; Dom et al., 2012; Ir. Wahidin, 2009).

5.5. Fuzzy Parameters

A group of equations to obtain the fuzzy parameters are provided as (Gu et al., 2006; Popescu & Giuclea, 2007):

$$
\begin{aligned}
s11b1 + s12b2 + \ldots + s1kbk &= s1y \\
s21b1 + s22b2 + \ldots + s2kbk &= s2y \\
s31b1 + s32b2 + \ldots + s3kbk &= s3y \\
s41b1 + s42b2 + \ldots + s4kbk &= s4y \\
s51b1 + s52b2 + \ldots + s5kbk &= s5y \\
&\vdots \\
sk1b1 + sk2b2 + \ldots + skkbk &= sky
\end{aligned}
\quad (4)
$$

here

$$s_{ij} = \sum u \sum uxixj - \sum uxi \sum ux$$

and

$$s_{iy} = \sum u \sum uxiy - \sum uxi \sum uy$$

According to this group of equations, first we can obtain the values of variables $b1, b2, \ldots, bk$ and finally $b0$ is gained by:

$$b_0 = \frac{\sum uy}{\sum u} - b_1 \frac{\sum ux1}{\sum u} - b_2 \frac{\sum ux2}{\sum u} - \ldots - b_k \frac{\sum uxk}{\sum u}$$

$$(5)$$

5.6. Coefficient of Determination

Coefficient of determination is used to assess the quality of estimation models and is expressed by R^2. The coefficient R^2 calculated by (Gu et al., 2006):

$$R^2 = \frac{\sum_{i=1}^{n}(\breve{y} - \overline{y})^2}{\sum_{n=1}^{n}(y - \overline{y})^2} \qquad (6)$$

Here, $\overline{y}$ expresses the mean value of random variables. Obviously, the coefficient R^2 describes the percentage of variability and the value is between 0 and 1; when an R^2 is close to 1, it indicates that this model can explain variability in response to the predictive variables, so called there is a strong relationship between the independent and dependent variables.

5.7. Importance of Software Risk Factors in Analysis Phase

Table 2 illustrates all respondents which indicated that the risk of "developer software gold–plating" was the highest risk factors and very important. In fact, the software risk factors from risk number 3, 4, 5, 6, 1, 8, 2 were identified as very important, the software risk factors from risk number 9, 7, 10 in descending means were identified as important, aggregating the responses resulted in the following ranking of the importance of the listed risks (in order of importance): Risk 3, Risk 4, Risk5, Risk 6, Risk 1, Risk 8, Risk 2, Risk 9, Risk 7, Risk 10.

Risk	N	Mean	Std. Deviation	%
R3	76	4.145	.743	82.895
R4	76	4.092	.819	81.842
R5	76	4.079	.796	81.579
R6	76	4.026	.748	80.526
R1	76	4.026	.588	80.526
R8	76	4.013	.792	80.263
R2	76	4	.849	80
R9	76	3.947	.728	78.947
R7	76	3.921	.963	78.421
R10	76	3.895	.793	77.895
Total	76	4.014	0.544	80.289

Table 2. Mean score for each software risk factor (analysis phase).

5.8. Ranking of Importance of Software Risk Factors for Project Managers' Experience

As we see, the results in Table 3 show that most of the risks are very important and important the overall ranking of importance of each software risk factor for the three categories of project managers' experience.

Phase	Risk	Experience 2-5 years	Experience 6-10 years	Experience 10 years
	R1	R3	R3	R5
	R2	R2	R4	R4
	R3	R1	R6	R3
	R4	R6	R5	R6
Analysis	R5	R8	R7	R7
	R6	R5	R1	R2
	R7	R4	R9	R10
	R8	R9	R8	R9
	R9	R10	R10	R6
	R10	R7	R2	R1

Table 3. Overall risk ranking of each software risk factor (analysis phase).

5.9. Frequency of Occurrence of Controls

Table 4 shows the mean and the standard deviation for each control factor. The results of this paper show that most of the controls are used most of the time and often.

5.10. Relationships between Software Risks and Risk Management Techniques in Analysis Phase

Regression technique was performed on the data to determine whether there were significant relationships between control factors and software risk factors. These tests were performed using fuzzy regression analysis, to compare the controls to each of the software risk factors to determine if they are effective in reducing the occurrence of each risk factor. Relationships between software risks and controls, which were significant and insignificant, any control is not significant, we are not reported according to the best model with fuzzy concepts.

Control	Mean	Std. Deviation	% percent
C29	4.408	0.803	88.15789
C30	4.368	0.907	87.36842
C20	4.184	0.668	83.68421
C27	4.171	0.755	83.42105
C21	4.171	0.7	83.42105
C19	4.158	0.612	83.15789
C28	4.158	0.767	83.15789
C25	4.132	0.718	82.63158
C26	4.118	0.653	82.36842
C23	4.105	0.741	82.10526
C22	4.092	0.786	81.84211
C18	4.079	0.726	81.57895
C10	4.079	0.726	81.57895
C17	4.066	0.718	81.31579
C24	4.066	0.639	81.31579
C8	4.066	0.736	81.31579
C5	4.053	0.728	81.05263
C11	4.039	0.756	80.78947
C15	4.039	0.621	80.78947
C9	4.039	0.756	80.78947
C14	4.013	0.683	80.26316
C7	4.013	0.721	80.26316
C16	4	0.693	80
C12	3.987	0.841	79.73684
C6	3.987	0.739	79.73684
C4	3.987	0.757	79.73684
C3	3.974	0.783	79.47368
C2	3.934	0.66	78.68421
C1	3.895	0.665	77.89474
C13	3.868	0.754	77.36842

Table 4. Mean score for each control factor.

R1: Risk of 'Unclear, Incorrect, Continually and Rapid Changing Software Project Requirements' Compared to Controls.

C1	C6	C11
.282*	.238*	.235*

* Correlation is significant at the 0.05 level (2-tailed).
**Correlation is significant at the 0.01 level (2-tailed).

Table 5. Illustration of the value of correlation.

Table 5 shows that the significant value is less than the assumed value at the $\alpha = 0.05$ level of significance, so there is a positive relation be-

tween controls 1, 6, 11 and risk 1. Also the control 1 has an impact on the risk 1. Additionally, the results show that the control 1 has positive impact value of 0.282, and the value of R^2 is 0.013367. This is interpreted as a percentage of 1.33 % from the dependent variable of risk1. According to the fuzzy concepts in multiple regression analysis to produce a fuzzy multiple regression model by solving these equations, the final fuzzy equation is:

$$FuzzyRisk1 = 4.571930941 + 0.090749888 * C1 \tag{7}$$

R2: Risk of 'Failure to Incomplete or Missing Detailed Requirements Analysis and Specification Documentation' Compared to Controls.

C1	C2	C3	C4	C5
.386**	.441**	.521**	.474**	.339**
C6	C7	C8	C9	C10
.455**	.476**	.314**	.310**	.338**
C11	C16	C19	C20	C25
.304**	.241*	.286*	.251*	.349**
C26	C29	C30		
.233*	.274*	.291*		

Table 6. Illustration of the value of correlation.

Table 6 shows that the significant value is less than the assumed value at the $\alpha = 0.05$ level of significance, so there is a positive relation between controls 1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 16, 19, 20, 25, 26, 29, 30, and risk 2. Controls 3 and 7 have an impact on the risk 2. Additionally, the results show that control 3 and 7 have a positive impact values of 0.521 and 0.251 respectively; a multiple correlation value is 0.575, and the value of R^2 is 0.1383. This is interpreted as a percentage of 13.83 % from the dependent variable of risk 2. According to the fuzzy concepts in multiple regression analysis to produce a fuzzy multiple regression model by solving these equations, the final fuzzy equation is:

$$FuzzyRisk2 = 2.613484124 + 0.239186282 * C3 + 0.246909848 * C7 \tag{8}$$

R3: Risk of 'Developer Software Gold-Plating' Compared to Controls.

C1	C2	C3	C4	C5
.350**	.341**	.366**	.352**	.269*
C6	C7	C8	C9	
.332**	.407**	.330**	.257*	
C10	C11	C15	C19	
.317**	.306**	.258*	.284*	
C21	C23	C25	C28	
.268*	.275*	.307**	.232*	

Table 7. Illustration of the value of correlation.

Table 7 shows that the significant value is less than the assumed value at the $\alpha = 0.05$ level of significance, so there is a positive relation between controls 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 15, 19, 21, 23, 25, 28, and risk 3. Also, controls 7 and 25 have an impact on risk 3. Additionally, the results show that controls 7 and 25 have a positive impact values of 0.407 and 0.307 respectively, also a multiple correlation value of 0.466, and the value of R^2 is 0.1432. This is interpreted as a percentage of 14.32 % from the dependent variable of risk 3. According to the fuzzy concepts in multiple regression analysis to produce a fuzzy multiple regression model by solving these equations above, the final fuzzy equation is:

$$FuzzyRisk3 = 2.707681169 + 0.239186282 * C7 + 0.246909848 * C25 \tag{9}$$

R4: Risk of 'Lack of IT Management' Compared to Controls.

C1	C2	C3	C4	C5
.424**	.398**	.448**	.348**	.340**
C6	C7	C8	C9	C10
.534**	.403**	.404**	.383**	.342**
C11	C12	C16	C18	C19
.308**	.240*	.229*	.322**	.238*
C20	C28	C29	C30	
.250*	.331**	.284*	0.236*	

Table 8. Illustration of the value of correlation.

Table 8 shows that the significant value is less than the assumed value at the $\alpha = 0.05$ level of significance, so there is a positive relation between controls 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 16, 19, 20, 28, 29, 30 and risk 4. Also, controls 3 and 6 have an impact on risk 4. Additionally, the results show that controls 3, and 6 have a positive impact values of 0.448, and 0.534 respectively, also a multiple correlation value of 0.573, and the value of R^2 is 0.14499. This is interpreted as a percentage of 14.499 % from the dependent variable of risk 4. According to the fuzzy concepts in multiple regression analysis to produce a fuzzy multiple regression model by solving these equations, the final fuzzy equation is:

$$FuzzyRisk4 = 3.102883775 + 0.079997397 * C3 + 0.325755516 * C6 \tag{10}$$

R5: Risk of 'Software Project Requirements Not Adequately Identified and Mismatched' Compared to Controls.

C1	C2	C3	C4
.390**	.297**	.347**	.239*
C5	C6	C7	C10
.271*	.264*	.355**	.252*
C11	C21	C25	C29
.279*	.285*	.231*	.249*

Table 9. Illustration of the value of correlation.

Table 9 shows that the significant value is less than the assumed value at the $\alpha = 0.05$ level of significance, so there is a positive relation between controls 1, 2, 3, 4, 5, 6, 7, 10, 11, 21, 25, 29 and risk 5. Controls 1 and 21 have an impact on risk 5. Additionally, the results show that controls 1, 21 have positive impact values of 0.390, and 0.285 respectively; a multiple correlation value is 0.517 and the value of R^2 is 0.0570. This is interpreted as a percentage of 5.70 % from the dependent variable of risk 5. According to the fuzzy concepts in multiple regression analysis to produce a fuzzy multiple regression model by solving these equations,

the final fuzzy equation is:

$$FuzzyRisk5 = 4.15600521 + 0.204598689 * C1$$
$$- 0.153570292 * C20$$
$$+ 0.149643258 * C21$$

$$(11)$$

R6: Risk of 'Inadequate Knowledge about Tools and Programming Techniques' Compared to Controls.

C1	C2	C3	C4	C5
.483**	.369**	.430**	.345**	.346**
C6	C7	C8	C9	C10
.448**	.469**	.305**	.337**	.344**
C11	C14	C16	C17	C18
.346**	.303**	.313**	.244*	.281*
C19	C23	C25	C29	C30
.269*	.230*	.269*	.249*	.228*

Table 10. Illustration of the value of correlation.

Table 10 shows that the significant value is less than the assumed value at the $\alpha = 0.05$ level of significance, so there is a positive relation between controls 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 16, 17, 18, 23, 25, 29, 30, and risk 6. Additionally, controls 1, 7, and 16 have an impact on risk 6. Besides, the results show that controls 1, 7, and 16 have a positive impact values of 0.483, 0.469, and 0.313 respectively, also a multiple correlation value is 0.594, and the value of R^2 is 0.332048. This is interpreted as a percentage of 33.20 % from the dependent variable of risk 6. According to the fuzzy concepts in multiple regression analysis to produce a fuzzy multiple regression model by solving these equations above, the final fuzzy equation is:

$$FuzzyRisk6 = 0.751844957 + 0.26037583 * C1$$
$$+ 0.293694496 * C7$$
$$+ 0.302258836 * C16$$

$$(12)$$

R7: Risk of 'Lack of Traceability, Confidentiality, Correctness and Inspection of the Software Project Planning' Compared to Controls.

C1	C2	C3	C3	C4
.470**	.361**	.453**	.268*	.289*
C6	C7	C8	C9	C10
.464**	.428**	.485**	.393**	.462**
C11	C12	C14	C16	C18
.289*	.316**	.264*	.321**	.259*
C19	C21	C23	C24	C29
.239*	.316**	.322**	.252*	.257*

Table 11. Illustration of the value of correlation.

Table 11 shows that the significant value is less than the assumed value at the $\alpha = 0.05$ level of significance, so there is a positive relation between controls 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 14, 16, 18, 19, 21, 23, 24 , 29 and risk 7. Also, controls 1 and 8 have an impact on risk 7. Additionally, the results show that controls 1 and 8 have a positive impact values of 0.470 and 0.485 respectively; a multiple correlation value is 0.581, and the value of R^2 is 0.0949. This is interpreted as a percentage of 9.49 % from the dependent variable of risk 7. According to the fuzzy concepts in multiple regression analysis to produce a fuzzy multiple regression model by solving these equations, the final fuzzy equation is:

$$FuzzyRisk7 = 2.745257135 + 0.225369972 * C1$$
$$+ 0.228982572 * C8$$

$$(13)$$

R8: Risk of 'Major Requirements Change After Software Project Plan Phase' Compared to Controls.

C1	C2	C3	C4	C5
.475**	.476**	.477**	.387**	.355**
C6	C7	C8	C9	C10
.460**	.501**	.396**	.423**	.557**
C11	C12	C13	C14	C16
.456**	.333**	.257*	.297**	.241*
C17	C18	C19	C21	C22
.226*	.400*	.394*	.267*	.240*
C23	C24	C25	C26	C27
.352**	.246*	.290*	.263*	.273*
C28	C29	C30		
.339**	.374**	.306**		

Table 12. Illustration of the value of correlation.

Table 12 shows that the significant value is less than the assumed value at the $\alpha = 0.05$ level of significance, so there is a positive relation between controls 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 16, 17, 18, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, and risk 18. Also, controls 7 and 10 have an impact on risk 8. Additionally, the results show that controls 7, and 10 have positive impact values of 0.501, and 0.557 respectively, also, the multiple correlation value is 0.613, and the value of R^2 is 0.1060. This is interpreted as a percentage of 10.60 % from the dependent variable of risk 8. According to the fuzzy concepts in multiple regression analysis to produce a fuzzy multiple regression model by solving these equations, the final fuzzy equation is:

$$FuzzyRisk8 = 4.565963809 \\ + 0.018349831 * C7 \qquad (14) \\ + 0.080708422 * c10$$

R9: Risk of 'Changing Software Project Specifications' Compared to Controls.

C1	C2	C3	C6	C7
.363**	.367**	.393**	.239*	.293*
C8	C9	C10	C11	C12
.284*	.296**	.337**	.277*	.301**
C13	C14	C16	C19	C21
.234*	.289*	.320**	.248*	.407*
C22	C23	C24	C29	C30
.236*	.384*	.233*	.258*	.290*

Table 13. Illustration of the value of correlation.

Table 13 shows that the significant value is less than the assumed value at the $\alpha = 0.05$ level of significance, so there is a positive relation between controls 1, 2, 3, 6, 7, 8, 9, 10, 11, 12, 13, 14, 16, 21, 22, 23, 24, 29, 30, and risk 9. Also, controls 3 and 21 have an impact on risk 9. Additionally, the results show that controls 3 and 21 have positive impact values of 0.393 and 0.407 respectively; a multiple correlation value is 0.535, and the value of R^2 is 0.0414. This is interpreted as a percentage of 4.14 % from the dependent variable of risk 9. According to the fuzzy concepts in multiple regression analysis to produce a fuzzy multiple regression model by

solving these equations, the final fuzzy equation is:

$$FuzzyRisk9 = 3.639559441 \\ + 0.091936178 * C3 \qquad (15) \\ + 0.167103717 * C21$$

R10: Risk of 'Inadequate Value Analysis to Measure Progress' Compared to Controls.

C1	C2	C3	C7	C10	C12
.269*	.255*	.295*	.243*	.247*	.236*
C19	C20	C21	C22	C23	
.284*	.305**	.284*	.276*	.337**	

Table 14. Illustration of the value of correlation.

Table 14 shows that the significant value is less than the assumed value at the $\alpha = 0.05$ level of significance, so there is a positive relation between controls 1, 2, 3, 7, 10, 12, 19, 20, 21, 22, 23 and risk 10. Also, controls 20 and 23 have an impact on risk 10. Additionally, the results show that controls 20 and 23 have a positive impact values of 0.305 and 0.337 respectively, also a multiple correlation value is 0.400, and the value of R^2 is 0.0282. This is interpreted as a percentage of 2.82 % from the dependent variable of risk 10. According to the fuzzy concepts in multiple regression analysis to produce a fuzzy multiple regression model by solving these solving these equations, the final fuzzy equation is:

$$FuzzyRisk10 = 3.728926746 \\ + 0.124254526 * C20 \qquad (16) \\ + 0.10610371 * C23$$

5.11. Software Risk Factors Identification Checklists and Control Factors (Risk Management Techniques)

Table 15 shows a software risk factors identification checklist with control techniques based on a questionnaire of experienced software project managers. We can use the checklist on software analysis projects to identify model software risk factors on software development life cycle by risk management techniques.

No	Software Risk Factors (Analysis Phase)	Risk Management Techniques
1	Unclear, incorrect, continual and rapid changing software project requirements.	C1: Using of requirements scrubbing.
2	Failure to incomplete or missing detailed requirements analysis and specification documentation.	C3: Assessing cost and scheduling the impact of each change to requirements and specifications, C7: Developing contingency plans to cope with staffing problems.
3	Developer software gold-plating.	C7: Developing contingency plans to cope with staffing problems, C25: Avoiding having too many new functions on software projects.
4	Lack of IT Management.	C6: Implementing and following a communication plan, C3: Assessing cost and scheduling the impact of each change to requirements and specifications.
5	Software project requirements not adequately identified and mismatched.	C1: Using of requirements scrubbing, C21: Including formal and periodic risk assessment, C20: Involving management during the entire software project life cycle.
6	Inadequate knowledge about tools and programming techniques.	C6: Using of requirements scrubbing, C7: Developing contingency plans to cope with staffing problems, C16: Implementing/Utilizing automated version control tools.
7	Lack of traceability, confidentiality, correctness and inspection of the software project planning.	C8: Assigning responsibilities to team members and rotating jobs, C1: Using of requirements scrubbing.
8	Major requirements change after software project plan phase.	C10: Reviewing and communicating progress to date and setting objectives for the next phase, C7: Developing contingency plans to cope with staffing problems.
9	Changing software project specifications.	C21: Including formal and periodic risk assessment, Involving management during the entire software project lifecycle, C3: Assessing cost and scheduling the impact of each change to requirements and specifications.
10	Inadequate value analysis to measure progress.	C23: Educating users on the impact of changes during the software project, C20: Involving management during the entire software project life cycle.

Table 15. Software risk factors were mitigated by risk management techniques.

6. Conclusions

The concern of our paper is mitigating software project risks in the analysis phase. The results show that all risks in software projects were very important and important in software project manager's perspective, whereas all controls are used most of the time, and often. Therefore, the software risk factors in the analysis phase from risk numbers 3, 4, 5, 6, 1, 8, 2 were identified as very important, the risk factors from risk numbers 9, 7, 10 in descending means were identified as important, aggregating the responses resulted in the following ranking of the importance of the listed risks (in order of importance): Risk 3, Risk 4, Risk5, Risk 6, Risk 1, Risk 8, Risk 2, Risk 9, Risk 7, and Risk 10. The results of this paper show also that most of the top ten controls are used most of the time. These tests were performed using fuzzy regression analysis, to compare the control techniques to each of the software risk factors to determine if they are effective in mitigating and modelling the occurrence of each software risk factor. Relationships between risks and controls, which were significant and insignificant, any control techniques is no significant, we are not reported. However, we determined the positive correlation between software risk factors in the analy-

sis phase and risk management techniques, then evaluate and estimated the impact software risk in software development life cycle. We used correlation analysis, fuzzy regression analysis techniques proposed. However, we referred the risk management techniques were mitigated on risk factors in Table 15. Through the results, we found out that some control haven't impact, so the important controls should be considered by the software development companies in Palestinian.

In addition to the above, we cannot obtain historical data form database until using some techniques. As future work, we will intend to apply these study results on a real-world software project to verify the effectiveness of the new techniques and approaches on a software project. We can use more techniques useful to manage software project risks such as neural network, genetic algorithm, and Bayesian statistics and other artificial intelligence approaches to improve the models.

Acknowledgments

References

[1] K. AALI, M. PARSINEJAD, B. RAHMANI, Estimation of Saturation Percentage of Soil Using Multiple Regression, ANN, and ANFIS Techniques. *Comput. Inf. Sci.*, **2**(3) (2009), 127–136.

[2] T. ADDISON, S. VALLABH, Controlling Software Project Risks – an Empirical Study of Methods used by Experienced Project Managers. In *Proceedings of SAICSIT*, (2002), pp. 128 – 140.

[3] P. BANNERMAN, Managing Structure-Related Software Project Risk: A New Role for Project Governance. In *21st Australian Software Engineering Conference*, (2010), pp. 129–138.

[4] Y. CHEN, C. WENG, Mining fuzzy association rules from questionnaire data. *Knowledge-Based Syst.*, **22**(1) (2009), 46–56.

[5] R. DASH, R. DASH, Risk Assessment Techniques for Software Development. *Eur. J. Sci. Res.*, **42**(4) (2010), 629–636.

[6] J. DHLAMINI, I. NHAMU, A. KACHEPA, Intelligent Risk Management Tools for Software Development. In *Proceedings of the 2009 Annual Conference of the Southern African Computer Lecturers Association*, (2009), pp. 33–40.

[7] R. DOM, B. ABIDIN, S. KAREEM, S. ISMAIL, N. DAUD, Determining the Critical Success Factors of Oral Cancer Susceptibility Prediction in Malaysia Using Fuzzy Models. *Sains Malaysiana*, **41**(5) (2012), 633–640.

[8] A. ELZAMLY, B. HUSSIN, Estimating Quality-Affecting Risks in Software Projects. *Int. Manag. Rev. Am. Sch. Press*, **7**(2) (2011), 66–83.

[9] A. ELZAMLY, B. HUSSIN, Managing Software Project Risks with Proposed Regression Model Techniques and Effect Size Technique. *Int. Rev. Comput. Softw.*, **6**(2) (2011), 250–263.

[10] A. ELZAMLY, B. HUSSIN, Managing Software Project Risks (Implementation Phase) with Proposed Stepwise Regression Analysis Techniques. *Int. J. Inf. Technol.*, **1**(4) (2013), 300–312.

[11] A. ELZAMLY, B. HUSSIN, An Enhancement of Framework Software Risk Management Methodology for Successful Software Development. *J. Theor. Appl. Inf. Technol.*, **62**(2) (2014), 410–423.

[12] X. GU, G. SONG, L. XIAO, Design of a Fuzzy Decision-making Model and Its Application to Software Functional Size Measurement. In *International Conference on Computational Intelligence for Modelling Control and Automation, and International Conference on Intelligent Agents,Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*, (2006), pp. 199–205.

[13] J. HOFFER, J. GEORGE, J. VALACICH, *Modern Systems Analysis and Design*, 6th ed. Prentice Hall, 2011.

[14] H. HOODAT, H. RASHIDI, Classification and Analysis of Risks in Software Engineering. *Eng. Technol.*, **56**(32) (2009), 446–452.

[15] W. IR. WAHIDIN, Chapter 3: Membership Function. In *Knowledge-based Systems*, Teknik Elektro Universitas Indonesia, 2009.

[16] K. KHANFAR, A. ELZAMLY, W. AL-AHMAD, E. EL-QAWASMEH, K. ALSAMARA, S. ABULEIL, Managing Software Project Risks with the Chi-Square Technique. *Int. Manag. Rev.*, **4**(2) (2008), 18–29.

[17] J. LIN, Q. ZHUANG, C. HUANG, Fuzzy Statistical Analysis of Multiple Regression with Crisp and Fuzzy Covariates and Applications in Analyzing Economic Data of China important. *Comput. Econ.*, **39**(1) (2012), 29–49.

[18] C. L. MARTIN, J. L. PASQUIER, C. M. YANEZ A. T. GUTIERREZ, Software Development Effort Estimation Using Fuzzy Logic: A Case Study. In *Proceedings of the Sixth Mexican International Conference on Computer Science (ENC'05)*, (2005), pp. 113–120.

[19] V. MARZA, M. SEYYEDI, Fuzzy Multiple Regression Model for Estimating Software Development Time. *International Journal of Engineering Business Management*, **1**(2) (2009), 31–34.

[20] J. MILER, J. GÓRSKI, Supporting Team Risk Management in Software Procurement and Development Projects. In *4th National Conference on Software Engineering*, (2002), pp. 1–15.

[21] ORACLE, A Standardized Approach to Risk Improves Project Outcomes and Profitability, 2010.

[22] R. PANDIAN, *Applied software risk management: A guide for software project managers*. Auerbach Publications is an imprint of the Taylor & Francis Group, 2007.

[23] C. POPESCU, M. GIUCLEA, A model of multiple linear regression. *Publ. house Rom. Acad.*, **8**(2) (2007), 1–8.

[24] F. REYES, N. CERPA, A. CANDIA, M. BARDEEN, The optimization of success probability for software projects using genetic algorithms. *J. Syst. Softw.*, **84**(5) (2011), 775–785.

[25] L. RUI-GE, W. BING-RONG, The Correlation Influence Factors Analysis in the Initial Dosage of Insulin Therapy of Diabetes. In *Second International Conference on Mechanic Automation and Control Engineering (MACE)*, (2011), pp. 774–777.

[26] K. SCHWALBE, *Information Technology Project Management*, Sixth. Course Technology, Cengage Learning, 2010.

[27] J. SODHI, P. SODHI, *IT Project Management Handbook*. Management Concepts, 2001.

[28] H. TAYLOR, The move to outsourced IT projects: key risks from the provider perspective. In *SIGMIS CPR '05 Proceedings of the 2005 ACM SIGMIS CPR Conference on Computer Personnel Research*, (2005), pp. 149–154.

[29] J. TAYLOR, *Managing Information Technology Projects: Applying Project Management Strategies to Software, Hardware, and Integration Initiatives*. AMACOM©2004, 2004.

[30] P. VASANT, Hybrid Mesh Adaptive Direct Search Genetic Algorithms and Line Search Approaches for Fuzzy Optimization Problems in Production Planning. In *Handbook of Optimization Intelligent Systems Reference Library*, **38**, (I. ZELINKA, V. SNÁŠEL, A. ABRAHAM, Eds.) (2013) pp. 779–799. Berlin, Heidelberg: Springer Berlin Heidelberg.

Contact addresses:

Abdelrafe Elzamly
Department of Computer Science
Faculty of Applied Sciences
Al-Aqsa University
P.O.BOX: 4051
Gaza
Palestine
e-mail: `abd_elzamly@yahoo.com`

Burairah Hussin
Fakulti Maklumat & Komunikasi
Universiti Teknikal Malaysia Melaka
Locked Bag 1752, Durian Tunggal
P.O.BOX: 76109
Durian Tunggal
Melaka Malaysia
e-mail: `burairah@utem.edu.my`

ABDELRAFE ELZAMLY is currently a Ph.D. student in Software and Information Systems Engineering at the Faculty of Information and Communication Technology at Technical University Malaysia Malaka (UTeM), Born on November 30, 1976, in Gaza, Palestine. He received his B.Sc. degree computer in 1999 from Al-Aqsa University, Gaza, and his Master's degree in computer information system in 2006 from The University of Banking and Financial Sciences. He works as lecturer of Computer Science at Al-Aqsa University from 1999 to 2014 full time and worked as lecturer in Islamic University in Gaza from 1999 to 2007 as part time lecturer. Also he worked as a manager in The Mustafa Center for Studies and Scientific Research -Gaza from 2010 to 2012. His research interests are risk management, quality software, software engineering, and data mining.

BURAIRAH HUSSIN got a Ph.D. in Management Science – Condition Monitoring Modelling from University of Salford, UK in 2007. He received his M.Sc. Degree in Numerical Analysis and Programming from University of Dundee, UK in 1998. He received his B.Sc. Degree in Computer Science from University Technology Malaysia in 1996. He is currently working as associate professor in University Technical Malaysia Malaka (UTeM) and as Research Manager at Centre for Advanced Computing Technology (C-ACT), Faculty of Information and Communication Technology at Technical University of Malaysia Malacca (UTeM). Also he worked as Deputy Dean (Research and Post Graduate), of the Faculty of Information and Communication Technology at Technical University of Malaysia Malacca (UTeM). His research interests are in data analysis, data mining, maintenance modelling, artificial intelligence, risk management, numerical analysis, and computer network advisor and development.