A Comparison of Fuzzy and Stepwise Multiple Regression Analysis Techniques for Managing Software Project Risks: Implementation Phase

Abdelrafe Elzamly[1], Burairah Hussin[2]
[1]**abd_elzamly@yahoo.com;** [2]burarah@utem.edu.my
Universiti Teknikal Malaysia Melaka (UTeM), Malaysia

[Abstract] This paper aims to present new techniques to determine if fuzzy and stepwise regression are effective in mitigating the occurrence software risk factor in the implementation phase. The proposed process compares the accuracy of prediction between stepwise multiple regression analysis techniques and fuzzy multiple regression. In addition, After applying MRE, the results show that the most value of MMRE in fuzzy multiple regression modelling for risks were slightly higher than the value of MMRE in stepwise multiple regression except risk 1 models around about fuzzy regression. Therefore, the most value of Pred (25) fuzzy multiple regression model for risks were slightly higher than or equal the value of pride (25) stepwise multiple regression except 9 were slightly higher than stepwise. The model's accuracy slightly improves in fuzzy multiple regression than stepwise multiple regression. Successful application of software project risk management will greatly improve the probability of project success.

[Keywords] software project management; software risk management; implementation phase; software risk factors; risk management techniques; stepwise multiple regression analysis techniques, fuzzy multiple regression analysis techniques; evaluating techniques

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. For some of these reasons corrective action is often difficult to cost-justify or to implement efficiently in practice (Masticola, 2007). The risk management process is becomes critical in controlling financial risk. 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). Clearly, the success or failure of software projects are generally assessed in three dimensions: budget, schedule, product functionality and quality (Miler, 2005). However, the goal of risk management is to achieve early identification of risks and then to actively change the course of actions required to mitigate the identified risks (Miler & Górski, 2002).

In our paper, we identified risk factors and risk management techniques that may guide software project managers to understand and mitigate risks associated with software development projects. Software Development Life Cycle is a process of creating and risk management techniques is used to mitigate risk it should involve in all phases include: Planning, analysis, design, implementation, and maintenance. In addition (Hoffer et al., 2011), we focused on implementation phase: It involves the actual construction and installation of a system. According to Taylor we should be applied techniques consistently throughout the software project risk management process (Taylor, 2004). Risk management is a practice of controlling risk and practice consists of processes, methods, and tools for managing risks in a software project before they become problems (Sodhi & Sodhi, 2001). Therefore, Boehm talked about value-based risk management, including principles and practices for risk identification, analysis, prioritization, and mitigation (Boehm, 2003)

The objective of this study is: To identify the software risk factors of software projects in the Palestinian software development organizations, to rank the software risk factors according to their importance, severity and occurrence frequency based on data source, to identify the activities performed by software project managers to manage the software project risks which identified, To compare the accuracy of prediction between stepwise multiple regression analysis techniques and fuzzy multiple regression analysis with concepts fuzzy by evaluation techniques.

Literature Review

The new technique used the chi-square ($\chi 2$) test to control the risks in a software project (Khanfar, Elzamly and et al. 2008). In addition, we also used new techniques the regression test and effect size test proposed to managing the risks in a software project and reducing risk with software process improvement. The nine of fourteen factors mitigated by using control factors (Elzamly & Hussin, 2011). Furthermore, the new stepwise regression technique manages the risks in a software project.

Top ten software risk factors in the implementation phase and thirty control factors were presented to respondents. These tests were performed using regression analysis to compare the controls to each of the risk factors to determine if they are effective in mitigating the occurrence of each risk factor and selecting best model (Elzamly & Hussin, 2013a). In addition, we proposed the new mining technique that uses the multiple regression analysis techniques with fuzzy concepts to managing the risks in a software project and reducing risk with software process improvement. Top ten software risk factors in design phase and thirty risk management techniques were presented to respondents (Elzamly & Hussin, 2013b).

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 dash described risk management is defined as the activity that identifies a risk; assesses the risk and defines the policies or strategies to alleviate or lessen the risk. Also Oracle corporation described risk management solutions enable a standardized approach for identifying, assessing and mitigating risk throughout the software project lifecycle (Oracle, 2010). Finally, risk management methodology that 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 techniques as risk qualitative analysis, risk quantitative analysis and risk mining analysis throughout the life of a software project to meet the goals.

Top 10 Software Risk Factors (Implementation Phase)

We displayed the top software risk factors in software development project lifecycle (implementation phase) that most common used by researchers when studying the risk in software projects. However, the list consists of the 10 most serious risks to a project ranked from one to ten, each risk's status, and the plan for addressing each risk. These factors need to be addressed and thereafter need to be controlled. Consequently, we presented 'top-ten' based on survey Boehm's 10 risk items 1991 on software risk management (Boehm, 1991), the top 10 risk items according to a survey of experienced project managers, Boehm's 10 risk items 2002 and Boehm's 10 risk items 2006-2007, (Miler, 2005), The Standish Group survey (CHAOS, 1995), (Addison & Vallabh, 2002), (Addison, 2003), Khanfar, Elzamly et al. (Khanfar et al., 2008), (Elzamly & Hussin, 2011), (Ezamly & Hussin, 2011), (Aloini, Dulmin, & Mininno, 2007), (Han & Huang, 2007)-(Huang & Han, 2008), (Aritua, Smith, & Bower, 2010), (Schmidt, Lyytinen, Keil, & Cule, 2001), (Keil, Cule, Lyytinen, & Schmidt, 1998), (Nakatsu & Iacovou, 2009), (J.-C. Chen & Huang, 2009), (Keil, Tiwana, & Bush, 2002), (Wallace, Keil, & Rai, 2004), (Sumner, 2000), (Boehm & Ross, 1989), (Kweku Ewusi-Mensah, 2003), (Paré, Sicotte, Jaana, & Girouard, 2008), (Houston, Mackulak, & Collofello, 2001), (Lawrence, Wiegers, & Ebert, 2001), (Shafer & Officer, 2004), (Hoodat & Rashidi, 2009), (Christopher Jones, Glen, Anna, & Miller, 2010), (Capers Jones, 2008), (Taimour, 2005), and another scholars, researchers in software engineering to obtain software risk factors and risk management techniques, these software risks are:

Table 1
Illustrates top ten software risk factors in software project lifecycle (implementation phase) based on researchers

Phase	No	Software risk factors	frequency
Implementation	1	Failure to gain user commitment	5
	2	Personnel shortfalls	4
	3	Failure to utilize a phased delivery approach	2
	4	Too little attention to breaking development and implementation into manageable steps	2
	5	Inadequate training team members	1
	6	Inadequacy of source code comments	1
	7	Inadequate test cases and generate test data	1
	8	Real-time performance shortfalls	1
	9	Test case design and Unit-level testing turns out very difficult	1
	10	Lack of adherence to programming standards	1
	Total frequency		19

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 rotate 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: Implement/ utilize benchmarking and tools of technical analysis, C18: Creating and analyzing process by simulation and modeling, C19: Provide 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 exercise 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: Maintain proper documentation of each individual's work, C29: Provide training in the new technology and organize domain knowledge training, C30: Participating users during the entire software project lifecycle.

Empirical Strategy

Data collection was achieved through the use of a structured questionnaire and historical data for assist in estimating the quality of software through determine risks that were common to the majority of software projects in the analyzed software companies. Top ten software risk factors and thirty control factors were presented to respondents. The method of sample selection referred to as ‘snowball’ and distribution 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 have participated in this study. The project managers that participated in this survey are coming from specific mainly software project manager in software development organizations.

Respondents were presented with various questions, which used scales1-7. For presentation purposes in this paper and for effectiveness, the point scale as the following: For choices, being headed 'unimportant' equal one and 'extremely important' equal seven. Similarly, seven frequency categories were scaled into 'never' equal one and 'always' equal 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. However to describe "software Development Company in Palestine" that have 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 [PITA 2012www.pita.ps/], Palestinian investment promotion agency [PIPA 2012 http://www.pipa.gov.ps/] to select top IT manager, software project managers. In order to find the relation among risks that the software projects confronts and the countermeasures that should be done to reduce risks, many researchers used different statistical methods. In this thesis, we used correlation analysis, stepwise multiple regression analysis, fuzzy multiple regression, MMRE, and Pred (l).

Regression Analysis 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). On the other words, fuzzy multiple regression model in which response variable is fuzzy variable and part of the covariates are crisp variables (Lin, Zhuang, & Huang, 2012). However, 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 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.

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):

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

Where μ is 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).

Fuzzy Parameters

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

s11b1 + s12b2 + ………….+ s1kbk = s1y
s21 b1 + s22b2 + ………….+ s2kbk = s2y
s31 b1 + s32b2 + ………….+ s3kbk = s3y
s41 b1 + s42b2 + ………….+ s4kbk = s4y
s51 b1 + s52b2 + ………….+ s5kbk = s5y
…………….
sk1 b1 + sk2b2 + ………….+ skkbk = sky (2)

Here

$$sij = \sum u \sum uXiXj - \sum uXi \sum uXj$$

, *and*

$$siy = \sum u \sum uXiy - \sum uXi \sum uy$$

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

$$b0 = \frac{\sum uy}{\sum u} - b1\frac{\sum ux1}{\sum u} - b2\frac{\sum ux2}{\sum u} ..... - bk\frac{\sum uxk}{\sum u} \quad (3)$$

Coefficient of Determination

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

$$R^2 = \frac{\sum_{i=1}^{n} (\tilde{y} - \bar{y})^2}{\sum_{i=1}^{n} (y - \bar{y})^2} \quad (4)$$

Here, $\tilde{y}$ express 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 the response to the predictive variables.

Stepwise Regression (adds and removes variables)

According to (Jin & Xu, 2012; Lan & Guo, 2008), stepwise regression method (SRM) combines and alternates between forward selection and backward elimination. At each step, the best remaining variable is added, provided it passes the significant at 5% criterion, then all variables currently in the regression are checked to see if any can be removed, using the greater than 10% significance criterion. In addition (Lan & Guo, 2008), the MSRA method is a stepwise optimization process of the multiple regression analysis method. Also, a stepwise-regression method that systematically adds and removes modal components based on statistical test to automatically identify the risks for a large scale data in operation(Zhou, Member, Pierre, & Trudnowski, 2012). Therefore (Jin & Xu, 2012), It is particularly useful when we need to predict a set of dependent variables from a large set of independent variables.

Evaluation Techniques Criteria

In order to validate the model with respect to its fitting accuracy we used the Mean Magnitude of Relative Error (MMRE) and PRED(25%) (Sentas, Angelis, & Stamelos, 2008). A common criterion for measurement of software effort estimation model performance (Kumar, Mandala, Chaitanya, & Prasad, 2011), which is calculated for each observation, is the magnitude of relative error (MRE) that the absolute value of the relative error (Alyahya, Ahmad, & Lee, 2009; Marza & Seyyedi, 2009):

$$MREi = \frac{|\text{Actual Effort}_i - \text{Predicted Effort}_i|}{\text{Actual Effort}_i} \quad (5)$$

We evaluated the impact of estimation accuracy using (MRE, MMRE) evaluation criteria, for each model. The mean magnitude of relative error (MMRE) is the average of all magnitudes of relative errors. Pred (25%) is the percentage of software projects with an MRE of 25% or less (Sentas et al., 2008). Therefore, with aggregation of MRE on all data set, the mean magnitude of relative error (MMRE) is achieved with equation below:

$$MMRE= \frac{1}{n}\sum_{i=1}^{i=n}\frac{|E_i - \overline{E_i}\,|}{E_i} \quad (6)$$

Therefore, we used pred(25) or more than according to the equation:

$$Pred(l) = \frac{K}{N} \quad (7)$$

To explain parameters k is the number of observations ,where MRE is less than or equal to l, for example, Pred(25) gives percentage of projects which were predicated with a counting the number of MRE less or equal than 0.25 and dividing by the number of projects (Alyahya et al., 2009). However, the accuracy of an estimation technique is proportional to pred(25) and inversely proportional to the MMRE (Martín, Pasquier, M., & T., 2005; Marza & Seyyedi, 2009). According to (Stensrud, 2003) MMRE is used for two kinds of assessments (at least). MMRE is to select between competing prediction models and to provide a quantitative measure of the uncertainty of a prediction.

Importance of Risk Factors in Implementation Phase

All respondents indicated that the risk of “Inadequacy of source code comments” was the highest risk factors and importance. In fact, all risk factors important, aggregating the responses resulted in the following ranking of the importance of the listed risks (in order of importance): Risk 6, Risk 3, Risk2, Risk 10, Risk 1, Risk 4, Risk 9, Risk 5, Risk 7, and Risk 8.

Table 2

Mean Score for Each Risk Factor (Implementation Phase)

Risk	N	Mean	Std. Deviation	% percent
r6	76	3.671	0.661	73.421
r3	76	3.658	0.793	73.158
r2	76	3.632	0.746	72.632
r10	76	3.553	0.79	71.053
r1	76	3.553	0.807	71.053
r4	76	3.513	0.757	70.263
r9	76	3.5	0.808	70
r5	76	3.487	0.808	69.737
r7	76	3.474	0.739	69.474
r8	76	3.474	0.774	69.474
Total	76	3.551	0.562	71.026

Ranking of Importance of Risk Factors for Project Managers' Experience

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

Table 3
The Overall Risk Ranking of Each Risk Factor

Phase	Risk	Experience 2-5 years	Experience 6-10 years	Experience >10 years
Implementation	R 1	r6	r6	r3
	R 2	r2	r3	r10
	R 3	r3	r1	r9
	R 4	r7	r2	r2
	R 5	r10	r8	r4
	R 6	r5	r10	r1
	R 7	r1	r9	r5
	R 8	r4	r4	r7
	R 9	r8	r7	r6
	R 10	r9	r5	r8

Frequency of Occurrence of Controls

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

Table 4
The Mean Score for Each Control Factor

Control	N	Mean	Std. Deviation	% percent
c29	76	4.408	0.803	88.15789
c30	76	4.368	0.907	87.36842
c20	76	4.184	0.668	83.68421
c27	76	4.171	0.755	83.42105
c21	76	4.171	0.7	83.42105
c19	76	4.158	0.612	83.15789
c28	76	4.158	0.767	83.15789
c25	76	4.132	0.718	82.63158
c26	76	4.118	0.653	82.36842
c23	76	4.105	0.741	82.10526
C13	76	3.868	0.754	77.36842

Relationships between Risks in Implementation Phase and Risk Management Techniques

Regression technique was performed on the data to determine whether there were significant relationships between risk management techniques and risk factors. These tests were performed using fuzzy multiple regression analysis and stepwise multiple regression analysis, to compare the controls to each of the software in implementation phase risk factors to determine if they are effective in mitigating the occurrence of each risk factor. Relationships between risks and controls, which were significant and insignificant, any control is no significant.

Table 5

Comparison between Estimation Stepwise and Fuzzy Multiple Regression by Evaluation Techniques

Model	Technique	Stepwise Multiple Regression	Fuzzy Multiple Regression
R1	MMRE	0.151405	0.145696
	PRED(25)	0.907894737	0.921052632
	R2	0.085	0.011942
R2	MMRE	0.122285526	0.129085815
	PRED(25)	0.894736842	0.894736842
	R2	0.181	0.071707055
R3	MMRE	0.138662	0.145134
	PRED(25)	0.868421	0.907895
	R2	0.133	0.032261
R4	MMRE	0.117376316	0.122682889
	PRED(25)	0.855263158	0.881578947
	R2	0.254	0.157685307
R5	MMRE	0.145222368	0.158727459
	PRED(25)	0.86842105	0.881578947
	R2	0.252	0.080967953
R6	MMRE	0.117546	0.106474
	PRED(25)	0.973684	0.986842
	R2	0.056	0.034275
R7	MMRE	0.14869	0.156454
	PRED(25)	0.89473684	0.88157895
	R2	0.105	0.03228
R8	MMRE	0.121075877	0.128156904
	PRED(25)	0.921052632	0.921052632
	R2	0.301	0.142030235
R9	MMRE	0.144827412	0.159808801
	PRED(25)	0.907894737	0.868421053
	R2	0.281	0.07865084
R10	MMRE	0.13562052	0.146130334
	PRED(25)	0.868421053	0.881578947
	R2	0.327	0.101139195

Software Risk Factors Identification Checklists and Control Factors (risk management techniques)

Table 6 shows software risk factors identification checklist with risk software projects based on a questionnaire of experienced software project managers. He can use the checklist on software projects to identify and mitigate software risk factors (implementation phase) on software projects by risk management techniques.

Table 6
Software Risk Factors in the implementation phase Mitigated By Risk Management Techniques

No	Software Risk Factors	Risk Management Techniques
1	Failure to gain user commitment.	C3: Assessing cost and scheduling the impact of each change to requirements and specifications.
2	Personnel shortfalls.	C3: Assessing cost and scheduling the impact of each change to requirements and specifications.
3	Failure to utilize a phased delivery approach.	C7: Developing contingency plans to cope with staffing problems.
4	Too little attention to breaking development and implementation into manageable steps.	C3: Assessing cost and scheduling the impact of each change to requirements and specifications, C29: Provide training in the new technology and organize domain knowledge training.
5	Inadequate training team members.	C3: Assessing cost and scheduling the impact of each change to requirements and specifications, C29: Provide training in the new technology and organize domain knowledge training.
6	Inadequacy of source code comments.	C4: Develop prototyping and have the requirements reviewed by the client.
7	Inadequate test cases and generate test data.	C3:Assessing cost and scheduling the impact of each change to requirements and specifications
8	Real-time performance shortfalls.	C29: Provide training in the new technology and organize domain knowledge training, C3: Assessing cost and scheduling the impact of each change to requirements and specifications.
9	Test case design and Unit-level testing turns out very difficult.	C2: Stabilizing requirements and specifications as early as possible, C28: Maintain proper documentation of each individual's work.
10	Lack of adherence to programming standards.	C7: Developing contingency plans to cope with staffing problems, C29: Provide training in the new technology and organize domain knowledge training, C1: Using of requirements scrubbing.

Conclusion

The concern of our paper is the managing risks of software projects. 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 time, and often. These tests were performed using stepwise multiple regression analysis and fuzzy multiple regression, to compare the controls to each of the risk software factors to determine if they are effective in mitigating the occurrence of each risk factor. However, we referred the control factors were mitigated on risk factors in Table 6. Through the results, we found out that some control haven't impacted, so the important controls should be considered by the software development companies in Palestinian.

Briefly, the results of comparison of the two models based on MMRE, PRED (25), and R^2. We notice the values of MMRE, and PRED (25) are quite similar or slightly higher than another model in the two models. While R^2 is not. This is explained by the non-deterministic (fuzzy) nature or fuzzy regression. If the problem at hand, involves non-deterministic (fuzzy) variable (fuzzy regression) is recommended which supports the need to use hybrid models in future research as proposed by Martín (Martín et al., 2005). In addition, we can’t 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 approach 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 so on.

Acknowledgements

This work is supported by Faculty of Information and Communication Technology, Technical University of Malaysia Malacca (UTeM), Malaysia and Al-Aqsa University, Palestine.

References

Addison, T. (2003). E-commerce project development risks : Evidence from a Delphi survey 1. *International Journal of Information Management, 23*(June 2001), 25–40.

Addison, T., & Vallabh, S. (2002). *Controlling Software Project Risks – an Empirical Study of Methods used by Experienced Project Managers.* In Proceedings of SAICSIT (pp. 128 – 140).

Aloini, D., Dulmin, R., & Mininno, V. (2007). Risk management in ERP project introduction: Review of the literature. *Information & Management, 44*(6), 547–567. doi:10.1016/j.im.2007.05.004

Alyahya, M. A., Ahmad, R., & Lee, S. P. (2009). Effect of CMMI-based software process maturity on software schedule estimation. *Malaysian Journal of Computer Science*, *22*(2), 121–137.

Aritua, B., Smith, N. J., & Bower, D. (2010). What risks are common to or amplified in program: Evidence from UK public sector infrastructure schemes. *International Journal of Project Management, 29*(3), 303–312. doi:10.1016/j.ijproman.2010.04.004

Boehm, B. (1991). Software Risk Management: Principles and Practices. *IEEE Software*, *8*(1), 32-41.

Boehm, B. (2003). Value-based software engineering. *ACM SIGSOFT Software Engineering Notes, 28*(2), 3. doi:10.1145/638750.638775

Boehm, B., & Ross, R. (1989). Theory-W Software Project Management: Principles and Examples. IEEE *Trans. Softw. Eng., 15*(7), 902–916.

CHAOS, T. S. G. 1995. (1995). The Standish Group 1995. Reprinted here for sole academic purposes with written permission from The Standish Group. CHAOS. 29/2/2012.

Chen, J.-C., & Huang, S.-J. (2009). An empirical analysis of the impact of software development problem factors on software maintainability. *Journal of Systems and Software, 82*(6), 981–992. doi:10.1016/j.jss.2008.12.036

Chen, Y.-L., & Weng, C.-H. (2009). Mining fuzzy association rules from questionnaire data. *Knowledge-Based Systems, 22*(1), 46–56. doi:10.1016/j.knosys.2008.06.003

Dash, R., & Dash, R. (2010). Risk Assessment Techniques for Software Development. *European Journal of Scientific Research, 42*(4), 629–636.

Dom, R., Abidin, B., Kareem, S., Ismail, S., & Daud, N. (2012). Determining the Critical Success Factors of Oral Cancer Susceptibility Prediction in Malaysia Using Fuzzy Models. *Sains Malaysiana, 41*(5), 633–640.

Elzamly, A., & Hussin, B. (2011). Managing Software Project Risks with Proposed Regression Model Techniques and Effect Size Technique. *International Review on Computers and Software (I.RE.CO.S.), 6*(2 March), 250–263.

Elzamly, A., & Hussin, B. (2013a). Managing Software Project Risks (Implementation Phase) with Proposed Stepwise Regression Analysis Techniques. I*nternational Journal on Information Technology (IREIT), 1*(4).

Elzamly, A., & Hussin, B. (2013b). Managing Software Project Risks (Design Phase) with Proposed Fuzzy Regression Analysis Techniques with Fuzzy Concepts.pdf. *International Review on Computers and Software (I.RE.CO.S.), 8*(11).

Ezamly, A., & Hussin, B. (2011). Estimating Quality-Affecting Risks in Software Projects. *International Management Review, 7*(2), 66–83. American Scholars Press.

Gu, X., Song, G., & Xiao, L. (2006). 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).

Han, W.-M., & Huang, S.-J. (2007). An empirical analysis of risk components and performance on software projects. *Journal of Systems and Software, 80*(1), 42–50. doi:10.1016/j.jss.2006.04.030

Hoffer, J., George, J., & Valacich, J. (2011). *Modern Systems Analysis and Design* (6th ed., p. 575). Prentice Hall.
Hoodat, H., & Rashidi, H. (2009). Classification and Analysis of Risks in Software Engineering. *Engineering and Technology, 56,* 446–452.
Houston, D., Mackulak, G., & Collofello, J. (2001). Stochastic simulation of risk factor potential effects for software development risk management. *Journal of Systems and Software, 59*(3), 247–257. doi:10.1016/S0164-1212(01)00066-8
Huang, S.-J., & Han, W.-M. (2008). Exploring the relationship between software project duration and risk exposure: A cluster analysis. *Information & Management, 45*(3), 175–182. doi:10.1016/j.im.2008.02.001
Jin, X., & Xu, X. (2012). Remote sensing of leaf water content for winter wheat using grey relational analysis (GRA), stepwise regression method (SRM) and partial least squares (PLS). In 2012 First International Conference on Agro- Geoinformatics (Agro-Geoinformatics) (pp. 1–5). Ieee. doi:10.1109/Agro-Geoinformatics.2012.6311706
Jones, Capers. (2008). *Applied Software Measurement Global Analysis of Productivity and Quality* (Third Edit., p. 662). McGraw-Hill Companies. doi:10.1036/0071502440
Jones, Christopher, Glen, G., Anna, G., & Miller, D. (2010). Strategies for improving systems development project success. *Issues in Information Systems (IIS), XI*(1), 164–173.
Keil, M., Cule, P., Lyytinen, K., & Schmidt, R. (1998). Aframework for Identifying Software Project Risks. *Communications of the ACM, 41*(11).
Keil, M., Tiwana, A., & Bush, A. (2002). Reconciling user and project manager perceptions of IT project risk: a Delphi study 1. *Information Systems Journal, (*April 1999), 103–119.
Khanfar, K., Elzamly, A., Al-Ahmad, W., El-Qawasmeh, E., Alsamara, K., & Abuleil, S. (2008). Managing Software Project Risks with the Chi-Square Technique. *International Management Review, 4*(2), 18–29.
Kumar, J., Mandala, A., Chaitanya, M. V., & Prasad, G. V. S. N. R. (2011). Fuzzy logic for Software Effort Estimation Using Polynomial Regression as Firing Interval. *IJCTA, 2*(6), 1843–1847.
Kweku Ewusi-Mensah. (2003). *Software Development Failures: Anatomy of Abandoned Projects* (p. 276). The MIT Press.
Lan, Y., & Guo, S. (2008). Multiple Stepwise Regression Analysis on Knowledge Evaluation. In 2008 International Conference on Management of e-Commerce and e-Government (pp. 297–302). Ieee. doi:10.1109/ICMECG.2008.65
Lawrence, B., Wiegers, K., & Ebert, C. (2001). The Top Risks of Requirements Engineering. *IEEE Software* (Vol. NOVEMBER/D, pp. 62–63).
Lin, J.-G., Zhuang, Q.-Y., & Huang, C. (2012). Fuzzy Statistical Analysis of Multiple Regression with Crisp and Fuzzy Covariates and Applications in Analyzing Economic Data of China important. *Computational Economics, 39*(1), 29–49.
Martín, C., Pasquier, J., M., C. Y., & T., A. G. (2005). Software Development Effort Estimation Using Fuzzy Logic: A Case Study. In Proceedings of the Sixth Mexican International Conference on Computer Science (ENC'05).
Marza, V., & Seyyedi, M. A. (2009). Fuzzy Multiple Regression Model for Estimating Software Development Time, 1(2), 31–34.
Masticola, S. (2007). A Simple Estimate of the Cost of Software Project Failures and the Breakeven Effectiveness of Project Risk Management. 2007 First International Workshop on the Economics of Software and Computation. doi:10.1109/ESC.2007.1
Miler, J. (2005). A Method of Software Project Risk Identification and Analysis. Technology. Gdansk University of Technology.
Miler, J., & Górski, J. (2002). Supporting Team Risk Management in Software Procurement and Development Projects. In 4th National Conference on Software Engineering.

Nakatsu, R., & Iacovou, C. (2009). A comparative study of important risk factors involved in offshore and domestic outsourcing of software development projects: A two-panel Delphi study. *Information & Management, 46*(1), 57–68. doi:10.1016/j.im.2008.11.005

Oracle. (2010). A Standardized Approach to Risk Improves Project Outcomes and Profitability. Oracle white paper.

Paré, G., Sicotte, C., Jaana, M., & Girouard, D. (2008). Prioritizing Clinical Information System Project Risk Factors : A Delphi Study. In Proceedings of the 41st Hawaii International Conference on System Sciences - (pp. 1–10).

Popescu, C., & Giuclea, M. (2007). A model of multiple linear regression. The publishing house of the Romanian Academy, 8(2), 1–8.

Schmidt, R. O. Y., Lyytinen, K., Keil, M., & Cule, P. (2001). Identifying Software Project Risks : An International Delphi Study. *Journal tjf Managemenl Information Systems /Spring, 17*(4), 5–36.

Schwalbe, K. (2010). *Information Technology Project Management* (sixth., p. 490). Course Technology, Cengage Learning.

Sentas, P., Angelis, L., & Stamelos, I. (2008). Multinomial Logistic Regression Applied on Software Productivity Prediction.

Shafer, D., & Officer, C. (2004). Software Risk : Why must we keep learning from experience ? Where does Risk Occur in DP Software Development ? In DYNAMIC POSITIONING CONFERENCE (pp. 1–19).

Sodhi, J., & Sodhi, P. (2001). IT Project Management Handbook. IT Project Management Handbook (p. 264). Management Concepts.

Stensrud, E. (2003). A Further Empirical Investigation of the Relationship between MRE and Project Size, 139–161.

Sumner, M. (2000). Risk factors in enterprise-wide/ERP projects. J*ournal of Information Technology, 15*(4), 317–327. doi:10.1080/02683960010009079

Taimour, A. (2005). Why IT Projects Fail; The Project Perfect. White Paper Collection, Retrieve from http://www.projectperfect.com.au/downloads/Info/info_it_projects_fail.pdf

Taylor, J. (2004). *Managing Information Technology Projects: Applying Project Management Strategies to Software, Hardware, and Integration Initiative*s (p. 274). AMACOM © 2004.

Wallace, L., Keil, M., & Rai, A. (2004). How Software Project Risk Affects Project Performance : An Investigation of the Dimensions of Risk and an Exploratory Model. *Decision Sciences, 35*(2).

Zhou, N., Member, S., Pierre, J. W., & Trudnowski, D. (2012). A Stepwise Regression Method for Estimating Dominant Electromechanical Modes. *IEEE Transactions on Power Systems, 27*(2), 1051–1059.