

Evaluation of Quantitative and Mining Techniques for Reducing Software Maintenance Risks

Abdelrafe Elzamly

Information and Communication Technology
University Technical Malaysia Malaka (UTeM)
Department of Computer Science
Al-Aqsa University, Gaza, Palestine

Burairah Hussin

Information and Communication Technology
University Technical Malaysia Malaka (UTeM)

Copyright © 2014 Abdelrafe Elzamly and Burairah Hussin. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract

Software risk is not always avoidable, but it is controllable. The aim of this paper is to present new techniques that were performed using quantitative and mining techniques to compare the risk management techniques to each of the software maintenance risks to identify and model if they are effective in mitigating the occurrence of each software maintenance risk in software development life cycle. The model's accuracy slightly improves in fuzzy multiple regression modelling techniques than or quite equal stepwise multiple regression modelling techniques. All models in fuzzy and stepwise acceptable value for MMRE less than 0.25 and Pred (0.25) greater or than 0.75 is desirable. The study has been conducted on a group of software project management. Successful software project risk management will greatly improve the probability of project success.

Keywords: Software Maintenance Project, Software Risk Management, Quantitative Techniques, Mining Techniques, MMRE, Pred (I)

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. Risk is an uncertainty that can have a negative or positive effect on meeting project objectives. Clearly, the success or failure of software projects are generally assessed in three dimensions such as budget, schedule, and product quality [15]. However, the goal of risk management at early identification and recognition of risks and then actively changes the course of actions to mitigate and reduce the risk [16]. In our paper, we identified risk factors and risk management techniques that are guide software project managers to understand and mitigate risks in software development projects. However, Software Development Life Cycle [12], is the process of creating, and the models and methodologies that people use to develop these systems. It includes these phases as Planning, analysis, design, implementation, and maintenance. In addition, we focused on maintenance phase that includes any future updates or expansion of the system. 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 [20]. **The objective of this study is:** To compare the accuracy of prediction between stepwise multiple regression analysis techniques and fuzzy multiple regression analysis by using evaluation techniques.

Literature Review

The new techniques the regression test and effect size test proposed to manage the risks in a software project. The nine of fourteen factors mitigated by using control factors [4]. Furthermore, we used the new stepwise regression technique to mitigate the risks in a software project. 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 in implementation phase [5]. In addition, the new mining technique that uses the fuzzy multiple regression analysis techniques with fuzzy concepts to manage the risks in a software project [6]. More than, we proposed fuzzy multiple regression analysis techniques to manage the software maintenance risks in a software project [7]. Further the new mining technique that uses the fuzzy regression analysis modelling techniques to mitigate the software planning risks in a software development project [8]. In addition, the study improved quality of software projects of the participating companies while estimating the quality-affecting risks in IT software projects [9]. Previous studies had shown that risk mitigation in software project can be classified by 3 categories such as qualitative, quantitative, and mining approaches. Quantitative risk is based on statistical methods that deal with accurate measurement about risk or leading to quantitative inputs that helped forming a regression model to understand how software project

risk factors influence project success such as stepwise regression models and other objective approach. Mining approach is a new way of identifying risk from data that create relationships between data and find the optimum result from them. This includes techniques such as simulation analysis, fuzzy multiple regression, neural network models, and others. In this paper, the author is focusing on quantitative and mining approaches to manage and model the software maintenance project risks. 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 [10].

Top 10 Software Maintenance Risk Factors

We displayed the top software maintenance risk factors in software development project life cycle that most commonly 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. These software maintenance project risks illustrate in table 1 below:

Table 1. Illustrates top ten software maintenance risk factors in software project based on researchers.

No	Software maintenance risk factors	Frequency
1	Inadequate knowledge/skills.	11
2	Inadequate change management.	6
3	Corporate politics with negative effect on software project.	5
4	Lack of resources and reference facilities.	4
5	Lack of top management commitment and support and involvement.	4
6	Shortfalls in externally furnished components, COTS.	3
7	Legacy software project.	1
8	Acquisition and contracting process mismatches.	1
9	User documentation missing or incomplete.	1
10	Harmful competitive actions.	1
Total frequency		37

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 specifica-

tions 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 for assisting 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 in Maintenance phase and thirty control factors were presented to respondents. The method of sample selection referred to as 'snowball' and distribution personal regular sampling was used. The seventy six project managers that participated in this survey are coming from specific mainly software project manager in software development organizations in Palestine. Respondents were presented with various questions, which used scales 1-7. 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.

5.1 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 [3].

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 [2]. 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.

5.2 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 [14]:

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

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 between 0 and 1.

5.3 Fuzzy Parameters

A group of equations to obtain the fuzzy parameters are provided as [11], [17]:

$$\begin{aligned} s_{11}b_1 + s_{12}b_2 + \dots + s_{1k}b_k &= s_{1y} \\ s_{21}b_1 + s_{22}b_2 + \dots + s_{2k}b_k &= s_{2y} \\ s_{31}b_1 + s_{32}b_2 + \dots + s_{3k}b_k &= s_{3y} \\ s_{41}b_1 + s_{42}b_2 + \dots + s_{4k}b_k &= s_{4y} \\ s_{51}b_1 + s_{52}b_2 + \dots + s_{5k}b_k &= s_{5y} \\ \dots & \\ s_{k1}b_1 + s_{k2}b_2 + \dots + s_{kk}b_k &= s_{ky} \end{aligned} \tag{2}$$

Here

$$s_{ij} = \sum u \sum u x_i x_j - \sum u x_i \sum u x_j$$

, and

$$s_{iy} = \sum u \sum u x_i y - \sum u x_i \sum u y$$

According to this group of equations, first we can obtain the values of variables b_1, b_2, \dots, b_k , and finally b_0 is gained by:

$$b_0 = \frac{\sum uy}{\sum u} - b_1 \frac{\sum ux_1}{\sum u} - b_2 \frac{\sum ux_2}{\sum u} \dots - b_k \frac{\sum ux_k}{\sum u} \tag{3}$$

5.4 Stepwise Multiple Regression (Adds and Removes Variables)

Lan and Guo (2008) reported stepwise multiple regression analysis method is

a stepwise optimization process of regression analysis method which is better to describe the relations between dependent variable and independent variables, and simulates the each kind of nature and economic phenomena with a better result. In addition [18], [21], [13], it is particularly useful when we need to predict a set of dependent variables from a large set of independent variables.

5.5 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%) [19]. 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 [19]. Therefore, with aggregation of MRE on all data set, the mean magnitude of relative error (MMRE) is achieved with the equation below:

$$\text{MMRE} = \frac{1}{n} \sum_{i=1}^n \frac{|E_i - \bar{E}_i|}{E_i} \quad (4)$$

Therefore, we used Pred (25) according to the equation:

$$\text{Pred (l)} = \frac{k}{N} \quad (5)$$

To explain parameters k is the number of observations, where MRE is less than or equal to l.

5.6 Relationships between Software Maintenance Risks and Risk Management Techniques

These tests were performed using fuzzy multiple regression analysis and stepwise multiple regression analysis, to compare the risk management technique to each of the software maintenance risks to identify and model if they are effective in reducing the occurrence of each software maintenance risk factor. Therefore, we used evaluation techniques to compare the accuracy of prediction between stepwise multiple regression analysis techniques and fuzzy multiple regression analysis such as MMRE, Pred (l).

5.7 Comparison between Estimation Stepwise and Fuzzy Multiple Regression by Evaluation Techniques

Table 2 illustrates an evaluation between stepwise multiple regression modelling and fuzzy multiple regression modelling by using MMRE and Pred (l) that comparing among various software project risk models. Thus, the model's accuracy slightly improves in fuzzy multiple regression than stepwise multiple regression. Also, all models in fuzzy and stepwise acceptable value for MMRE less than 0.25 and Pred (0.25) greater than 0.75 is desirable [1]. 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.

Table 2. Comparison between estimation stepwise and fuzzy multiple regression by evaluation techniques.

Model	Technique	Stepwise Multiple Regression	Fuzzy Multiple Regression
R1	MMRE	0.1166563	0.1243272
	Pred (25)	0.894736842	0.973684211
R2	MMRE	0.133269518	0.142860116
	Pred (25)	0.907894737	0.907894737
R3	MMRE	0.113166447	0.122261327
	Pred (25)	0.921052632	0.960526316
R4	MMRE	0.118555044	0.130245449
	Pred (25)	0.921052632	0.947368421
R5	MMRE	0.118524154	0.124903847
	Pred (25)	0.907894737	0.921052632
R6	MMRE	0.126987061	0.133983843
	Pred (25)	0.947368421	0.947368421
R7	MMRE	0.115816353	0.133061907
	Pred (25)	0.894736842	0.907894737
R8	MMRE	0.115386867	0.115386867
	Pred (25)	0.907894737	0.907894737
R9	MMRE	0.101817	0.110289
	Pred (25)	0.960526316	0.960526316
R10	MMRE	0.106521053	0.111920085
	Pred (25)	0.960526316	0.960526316

5.8 Software Maintenance Risk Factors Identification Checklists and Risk Management Techniques

Table 3 shows a software maintenance risks 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 risk factors on life cycle software projects by risk management techniques.

Conclusion

The results show that tests were performed using stepwise regression analysis, fuzzy regression, to compare the controls to each of the software risk factors to model if they are effective in mitigating the occurrence of each risk factor. Relationships between software maintenance risks and risk management techniques, which were significant and insignificant, any risk management technique is no significant, we are not reported. However, we referred the risk management techniques were mitigated on software maintenance risk factors based on a formula model in Table 3. Table 2 illustrates after applying MRE, the results show that the most value of MMRE in fuzzy multiple regression modelling for risks were slightly higher than or equal the value of MMRE in stepwise multiple regression. Therefore, the most value of Pred (25) fuzzy multiple regression model for risks were slightly higher than or equal the value of Pred (25) stepwise multiple regression. The model’s accuracy slightly improves in fuzzy multiple regression than stepwise multiple regression. So, all models in stepwise and fuzzy acceptable value for MMRE less than 0.25 and Pred (0.25)

greater or than 0.75 is desirable. In addition, we can't obtain historical data from database to use 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 artificial intelligence techniques useful to manage and model software maintenance project risks in phase.

Table 3. Formulas for software maintenance risks were mitigated by using risk management Techniques

No	Software Maintenance Risks	Risk Management Techniques
1	Inadequate knowledge/skills.	C1: Using of requirements scrubbing.
2	Inadequate change management.	C1: Using of requirements scrubbing, C20: Involving management during the entire software project Lifecycle.
3	Corporate politics with negative effect on software project.	C1: Using of requirements scrubbing, C23: Educating users on the impact of changes during the software project, C22: Utilizing change control board and exercise quality change control practices, C9: Have team-building sessions, C7: Developing contingency plans to cope with staffing problems
4	Lack of Resources, research and reference facilities.	C1: Using of requirements scrubbing, C23: Educating users on the impact of changes during the software project
5	Lack of top management commitment and support and involvement.	C11: Dividing the software project into controllable portions, C3: Assessing cost and scheduling the impact of each change to requirements and specifications.
6	Shortfalls in externally furnished components, Commercially available Off-The-Shelf (COTS).	C2: Stabilizing requirements and specifications as early as possible, C12: Reusable source code and interface methods.
7	Legacy Software project.	C3: Assessing cost and scheduling the impact of each change to requirements and specifications, C12: Reusable source code and interface methods, C26: Incremental development (deferring changes to later increments), C14: Reusable database and data mining structures.
8	Acquisition and contracting process mismatches.	C8: Assigning responsibilities to team members and rotate jobs, C12: Reusable source code and interface methods, C5: Developing and adhering a software project plan.
9	User documentation missing or incomplete.	C8: Assigning responsibilities to team members and rotate jobs, C24: Ensuring that quality-factor deliverables and task analysis, C1: Using of requirements scrubbing.
10	Harmful competitive actions.	C1: Using of requirements scrubbing, C9: Have team-building sessions.

Acknowledgements.

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

References

- [1] S. Basha and D. P, "Analysis of Empirical Software Effort Estimation Models," *Int. J. Comput. Sci. Inf. Secur.*, vol. 7, no. 3, pp. 68–77, 2010.

- [2] Y. Chen and C. Weng, "Mining fuzzy association rules from questionnaire data," *Knowledge-Based Syst.*, vol. 22, no. 1, pp. 46–56, Jan. 2009.
- [3] R. Dom, B. Abidin, S. Kareem, S. Ismail, and N. Daud, "Determining the Critical Success Factors of Oral Cancer Susceptibility Prediction in Malaysia Using Fuzzy Models," *Sains Malaysiana*, vol. 41, no. 5, pp. 633–640, 2012.
- [4] A. Elzamly and B. Hussin, "Managing Software Project Risks with Proposed Regression Model Techniques and Effect Size Technique," *Int. Rev. Comput. Softw.*, vol. 6, no. 2, pp. 250–263, 2011.
- [5] A. Elzamly and B. Hussin, "Managing Software Project Risks (Implementation Phase) with Proposed Stepwise Regression Analysis Techniques," *Int. J. Inf. Technol.*, vol. 1, no. 4, pp. 300–312, 2013.
- [6] A. Elzamly and B. Hussin, "Managing Software Project Risks (Design Phase) with Proposed Fuzzy Regression Analysis Techniques with Fuzzy Concepts," *Int. Rev. Comput. Softw.*, vol. 8, no. 11, pp. 2601–2613, 2013.
- [7] A. Elzamly and B. Hussin, "Identifying and Managing Software Project Risks With Proposed Fuzzy Regression Analysis Techniques: Maintenance Phase," in *Conference on Management and Engineering (CME2014)*, 2014, p. 1868–1881.
- [8] A. Elzamly and B. Hussin, "Managing Software Project Risks (Planning Phase) with Proposed Fuzzy Regression Analysis Techniques with Fuzzy Concepts," *Int. J. Inf. Comput. Sci.*, vol. 3, no. 2, pp. 31–40, 2014.
- [9] A. Elzamly and B. Hussin, "Estimating Quality-Affecting Risks in Software Projects," *Int. Manag. Rev. Am. Sch. Press*, vol. 7, no. 2, pp. 66–83, 2011.
- [10] A. Elzamly and B. Hussin, "An Enhancement of Framework Software Risk Management Methodology for Successful Software Development," *J. Theor. Appl. Inf. Technol.*, vol. 62, no. 2, pp. 410–423, 2014.
- [11] X. Gu, G. Song, and 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.
- [12] J. Hoffer, J. George, and J. Valacich, *Modern Systems Analysis and Design*, 6th ed. Prentice Hall, 2011, p. 575.
- [13] X. Jin and X. Xu, "Remote sensing of leaf water content for winter wheat using grey relational analysis (GRA), stepwise regression method (SRM) and partial least squares (PLS)," in *First International Conference on Agro- Geoinformatics (Agro-Geoinformatics)*, 2012, pp. 1–5.
- [14] V. Marza and M. Seyyedi, "Fuzzy Multiple Regression Model for Estimating Software Development Time," vol. 1, no. 2, pp. 31–34, 2009.

- [15] J. Miler, "A Method of Software Project Risk Identification and Analysis," Gdansk University of Technology, 2005.
- [16] J. Miler and J. Górski, "Supporting Team Risk Management in Software Procurement and Development Projects," in 4th National Conference on Software Engineering, 2002, pp. 1–15.
- [17] C. Popescu and M. Giuclea, "A model of multiple linear regression," Publ. house Rom. Acad., vol. 8, no. 2, pp. 1–8, 2007.
- [18] L. Rui-ge and 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.
- [19] P. Sentas, L. Angelis, and I. Stamelos, "Multinomial Logistic Regression Applied on Software Productivity Prediction," in 9th Panhellenic conference in informatics., 2003, pp. 1–12.
- [20] J. Sodhi and P. Sodhi, IT Project Management Handbook. Management Concepts, 2001, p. 264.
- [21] N. Zhou, J. Pierre, and D. Trudnowski, "A Stepwise Regression Method for Estimating Dominant Electromechanical Modes," IEEE Trans. Power Syst., vol. 27, no. 2, pp. 1051–1059, 2012.

Received: March 3, 2014