

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/296692355>

Quantitative and Intelligent Risk Models in Risk Management for Constructing Software Development Projects: A...

Article in *International Journal of Software Engineering and its Applications* · March 2016

DOI: 10.14257/ijseia.2016.10.2.02

CITATIONS

0

READS

267

2 authors:



Abdelrafe Elzamly

Technical University of Malaysia Malacca

27 PUBLICATIONS 147 CITATIONS

[SEE PROFILE](#)



Burairah Hussin

Univesiti Teknikal Malaysia, Melaka

105 PUBLICATIONS 311 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Customer Credit Classification Enhancement for Handling Imbalance Credit Data [View project](#)



Data Management of Malaysian Elderly Care [View project](#)

Quantitative and Intelligent Risk Models in Risk Management for Constructing Software Development Projects: A Review

Abdelrafe Elzamly^{1*} and Burairah Hussin²

¹*Department of Computer Science, Al-Aqsa University, Palestine*

^{1, 2}*Information and Communication Technology, Universiti Teknikal Malaysia Melaka (UTeM)*

** E-mail: Abd_elzamly@yahoo.com¹*

Abstract

Techniques and models for mitigating risk in software development projects classified into three categories—namely, qualitative, quantitative, and intelligent approaches. This paper is to review the quantitative and intelligent risk models in software risk management for software development projects. Indeed, this area needs more effort from scholars and researchers in quantitative and intelligent risk models to mitigate risks. As future work, we will use these hybrid models of quantitative and intelligent for mitigating software risks in cloud computing such as neural network, genetic algorithm and others artificial intelligence techniques.

Keywords: *Quantitative Risk Models, Intelligent Risk Models, Risk Management, Software Development Project*

1. Introduction

Although there are many methods in software risk management, software development projects have a high rate of risk failure. Much of the failure could be avoided by managers pro-actively maintenance and dealing with risk factors rather than waiting for problems to occur and then trying to react. 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 considered the planned control of risk. Integrating formal risk management with project management is a new phenomenon in software engineering and product management community. In addition, risk is an uncertainty that can have a negative or positive effect on meeting project objectives. Han and Huang (2007) [1] reported that the likelihood of occurrence risk is vary and its level of impact on the project budget, schedule, and other variables. 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. Risk management methodology that had five phases includes risk identification, risk analysis and evaluation, risk treatment, risk controlling, risk communication and documentation relied on three categories such as risk qualitative analysis, risk quantitative analysis and risk mining analysis throughout the life of a software project to meet the goals [2]. Today, we must think of risk is a part of software project process and is important for a software project survival. 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 [3]. A software project provides a solution to a functioning software-based information system such as enterprise resource planning system,

* Corresponding Author

software package, reports, tools analysis, reengineering software, and website design [4]. Islam (2009) reported that software project is usually faced with an unexpected problem that is difficult to estimate issues within the software development process. He classified the issues into technical and non-technical during the development of software project. Every software project has challenges which need to be alleviated to make it a successful completion [5]. Besides, the success of software project is increasingly important to the survival of businesses. Islam (2009) also contributed to a risk management project model to reduce risk in requirement stage. A key success for software project factors in software organizations is the software process improvement. Therefore, it is clear that without a good process, a software organization will fail to produce high quality software, mitigating risks and possibly fail to reach its objectives. Such problems in the software process model are missing in the target set for software process and improvement, low involvement of quality control activities, and the absence of standard business expertise practice [7]. Therefore, it is important to identify those Critical Success Factors (CSFs) that increase probability of project success. Surely, we need to focus on software project risk management practice and modelling in order to estimate software project risks. **The objective of this study is:** To review the quantitative and intelligent risk models for constructing software development projects.

2. Failure of Software Projects

Billions of dollars are spent on software projects developments due to their importance to companies, departments, and managers as reported by (Gorla and Lin, 2010). Software development projects are complex to mitigate risks and too many of them end in failure. For example it is reported that United States had spent approximately \$250 billion on software development projects in 1995 [9]. In addition, the 2004 report had shown more statistics about the outcome of software project (third quarter research report, 2004): Successful projects include 29%, canceled projects that cost \$55 billion annually, this is followed by challenged projects(53%) and failed projects (18%). According to Dominguez (2009) [11], the report as in Table 1 it is indicated that software projects now have a 32% success rate compared to 35% from the previous study in 2006 and 16% in the 1994 study. Furthermore, the Standish Group estimates that in 1995 American companies and government agencies spent \$81 billion for cancelled software projects [12]. According to Costa *et. al.*, (2005), any software projects are developed by small companies, the development organization may not survive a combination of project failures. Finally, Savolainen et al, (2012) reported that about 2100 sites (www) described more than 5000 reasons that software projects might fail, ranging from the poor use of technology. In addition, due to the massive resistance of end-users, billions of dollars have been wasted on failed projects, and a lot of very expensive projects had to be shelved after a short period of time [13].

Table 1. Standish CHAOS Report on Software Project Failure Updated for 2009 Year [14]

No.	2009	2006	2004	2002	2000	1998	1996
Successful	32%	35%	29%	34%	28%	26%	27%
Challenged	44%	19%	53%	15%	23%	28%	40%
Failed	24%	46%	18%	51%	49%	46%	33%

3. Principle of Risk Management

3.1. Concept of Risk

Historically, risk is defined as the probability that the real variables and the results may be different from those originally measured resulting in either positive or the negative impact [15]. Risk of failure is defined as the possibility of suffering due to harm, loss, or exposure in the life cycle of a software project [16]. Risk can be defined as imperfect knowledge where each action leads to a set of possible outcomes each with an unknown probability [17].

3.2. Risk Management

Risk management is well recognized as an important means of mitigating software failure [18]. Furthermore, risk management as any activity which is intended to help software project managers to recognize and manage the risk of cost overruns in software [19]. Risk management helps project manager and team to make better decisions, communication and to mitigate risk in project [20].

3.3. Concepts of Risk Management

Risk management is not a discrete single activity but a dynamic process, that is continuously more refined through its repetition throughout a software project's life cycle [21]. Therefore, risk management is to identify risky situations and develop strategies to mitigate the likelihood of occurrence and the negative effect of risky events [22]. Risk management is a practice of risk controlling and it consists of processes, methods, and tools for managing risks in a software project before they become problems [3]. A risk management process provides a strategic orientation for organizations of all sizes in all geographical areas with a formal process to identify measure and manage risk [17]. However detailed description is given by Dash and Dash [23], where risk management focuses on assessment of the likelihood of risk occurring, risk event drivers, risk events, the likelihood of impact and the impact drivers before the risk actually takes place. Finally, many authors defined risk management, but complex practice to measure the likelihood of impact of software risks and determine appropriate risk management techniques, especially in software development projects [24].

3.4. Elements of Risk Management

There are many approaches available for risk management [25] and the main risk management process includes scanning, verifying, risk planning, and risk education [26]. According to Holzmann and Spiegler (2010) risk management method starts with risk identification and classification of potential risk elements [27]. Clearly, there are two approaches in project risk management namely the evaluation method which focuses on the risk management project and analysis processes which aim to identify software risks [28]. Bakker *et. al.*, (2010) combined between two approaches to a software risk management.

3.5. Significance of Software Project Risk Management

The purpose of software risk management is to analyze the possibility of risks before they occur, and to identify the risk mitigating strategic that may be planned and used as needed during the software development lifecycle to mitigate software risks [29]. Today, software risk management has become a common principle and practice amongst leading software companies [30]. In the increasing effort to improve software development processes and software quality, recent studies have pointed out to an area of software project risk [31]. However, many of the software projects are risky and are often

considered runaway, because they do not meet the expectations of budgets and schedules. Therefore, an effective software risk management is extremely important to mitigate risks [32]. Indeed, risk management for software projects is now a common practice, so software managers should choose a proactive approach and technique to managing software risks in software projects [33]. In addition, risk management of software projects is highly relevant to the social and cultural context of the development activities. Thus, it is important to use software risk management to mitigate software project failure as reported by [18]. In a previous study, Oracle Corporation has proposed risk management solutions generally to enable a standardized approach for identifying, assessing and mitigating risks in enterprise project portfolios so as to reduce risks [34]. Therefore, risk management can be integrated with enterprise project portfolio management and scheduling solutions, thus providing input and visibility of risk to the entire project team. As a result, the risk management process helps to ensure that mission-critical projects come in on time and within budget.

3.6. Risk Management in Software Project

Software risk management helps to analyze and address other risks with software development lifecycle to achieve success software project [35]. According to Arnuphaptrairong (2011) [36], risk management is a difficult activity where software managers need to identify and mitigate uncertain risks of the software development lifecycle. Many software projects have a very high failure rate in spite of much effort has been put for its succession [37]. Software risk management is a software engineering practice that includes processes, methods, and tools for mitigating risks in a project (Rout, Parida, & Mishra, 2013).

3.7. Element of Risk Management in Software Project

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 [39]. Thus, if the complexity and the size of the software projects are increased, managing software development risk becomes more difficult [40]. In addition, the optimization method was tested with various software project risk prediction models that have been developed [41]. 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. With this methodology software managers have a sixth sense who for economy feel there is something wrong with the software risk management approach might returned to knowledge, self-judgment according to their experiences [29]. Fakhar *et. al.*, (2013) proposed the risk management system based on three risk management steps *i.e.*, risk identification, risk reduction and risk control [42]. Furthermore, risk management approach practice 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 [31].

In addition, the principles were proposed to manage software project risks by using risk management approach that is proactive, integrated, systematic, and disciplined [25]. In addition, Boehm (1991) reported that software risk management involves two main such as risk assessment phase such as risk identification, risk analysis, and risk prioritization; risk control phase as risk planning, risk resolution, and risk monitoring [45]. Finally, these 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 do not create relation between

software risk factors in software development lifecycle and risk management techniques to mitigate risks. However, none of them uses modelling approach to mitigate failure risk in software development. This study attempted to review the modelling software risk management for successful software project. Furthermore, they proposed the elements for software risk management that has five phases such as risk planning, risk identification, risk prioritization, risk analysis, risk evaluation, risk treatment, risk controlling, and risk communication and documentation [2].

4. Quantitative Risk Models

The chi-square (χ^2) test was used to control the risks in a software project [46]. However, The new techniques which were used to manage the risks in a software project and reducing risk with software process improvement by the regression test and effect size test [47]. Besides, we improved quality of software projects of the participating companies while estimating the quality-affecting risks in IT software projects. The results show that there were 40 common risks in software projects of IT companies in Palestine. The amount of technical and non-technical difficulties was very large [48]. Furthermore, stepwise regression technique was used to manage 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 implementation phase [49]. Padayachee (2002) reported as a typical risk management is the identification of risk, usually involving checklists, questionnaires or brainstorming sessions. In addition, fuzzy approach was developed with an effective algorithm to improve the quality and effectiveness of decision-making. The proposed approach demonstrated using a real case involving an Iranian construction corporation and the method could distinguish successfully [50]. The linear stepwise discriminant analysis model was introduced to predict software risks in software analysis development process. These methods were used to measure and predict risks by using control techniques [51]. It was reported that Carr and Tah (2001) proposed a methodology in software development that covers both process and information system models that based on the software risk management framework. Islam (2009) also was proposed a Goal-driven Software Development Risk Management Model (GSRM) that supports the identification, assessment, treatment, and documentation of risks in relation to software project-specific goals. Furthermore [52], proposed an method to measure the possibility distribution of harms and earnings that can be incurred by a software development organizations. Besides, framework modeling of the process evolution, it contains techniques to identify process risks and to derive from them suggestions for software process improvement [53]. The new software risk management framework was design to determine the risk performance measure from the quantitative survey and applying a risk management strategy [54]. Carr and Tah (2001) reported that provided a systematic approach to software risk management involves as the identification of risk sources; the quantification of their effects; the development of responses to these risks to these risks; and the control of residual risks in the software project estimates. A quantitative method determines probability of occurrence and its consequences software risk. In order to give the techniques a failure rating, we usually get the data from historical data or expert opinion. However, it aimed to introduce the linear stepwise discriminant analysis model to predict software risks in software analysis development process [51]. Furthermore, the discriminant analysis (DA) techniques were proposed to classify and manage risks in software planning development process. Also, these techniques are used to test the controls to each of risks to determine and classify if they are effective in mitigating the occurrence of each risk planning factor [55]. According to [56], presented a risk management supporting tool, detailing its functionality and the user interface appearance as well as giving some design and implementation details. The tool provides in particular do automatic the risk identification from interactively answered on-line checklists and the

qualitative risk evaluation. The risk assessment model, methods and techniques are widely used to control risk in a software development [23].

However, these tests are performed using stepwise regression analysis and Durbin–Watson statistic to compare the risk management techniques to each of the software maintenance risk factors to identify if they are effective in reducing the occurrence of each software maintenance risk factor and selecting the best model [57]. Furthermore, we identified risk factors and risk management techniques that are guide software project managers to understand and mitigate risks in software development projects [58]. Moreover, we propose artifact model of the software risk management for mitigating risks. Thus, it has the five levels to mitigate risks through software project [59]. The study presented the factor analysis technique to classify and identify the risk management techniques (controls) in the software development project for mitigating risks [60]. This study aimed to test the level of prediction between two model scenarios namely- stepwise and fuzzy multiple regression analysis by using MMRE and Pred (I) [61]. Furthermore, it aimed to predict the risks in software development projects by applying multiple logistic regression. The logistic regression was used as a tool to control the software development process [62].

5. Intelligent Risk Models

There are many risk analysis techniques currently in use to evaluate and estimate software risks but it is very important to choose appropriate model to reduce software risks [63]. In the Concurrent Simultaneous Engineering Resource View (ConSERV) was developed for an intelligent knowledge based on project management techniques and can be also used as risk management system [64]. In addition, the new mining technique that was used to manage risks in a software project by the fuzzy multiple regression analysis technique. However, these mining tests were performed using fuzzy multiple regression analysis techniques to compare the risk management techniques to each of the software risk factors to determine if they are effective in mitigating the occurrence of each software risk factor [65]. This paper was aimed to present the new techniques to determine if fuzzy and stepwise regression are effective in mitigating the occurrence software risk factor in the implementation phase [66]. Method associated was described with a special fuzzy operator, namely a two-additive Choquet integral that allows modeling different effects of importance and interactions among software risks [67]. The potential of the proposed methodology is exposed through an empirical case study conducted in a Turkish software company. Dhlamini *et. al.*, (2009) demonstrated the need for risk management tools in software project since the complexity of risk management increases with the complexity of the developed system. They proposed two frameworks for the development of intelligent risk management tools; neural networks and intelligent agent based. Consequently, an approach to modeling software risk factors and simulating their effects as a means of supporting certain software development risk management activities were described. This simulator is a device designed specifically for the risk management activities of assessment, mitigation, contingency planning, and intervention [68]. 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. Also, the development of a fuzzy decision support system (FDSS) for risk assessment in e-commerce (EC) development and a risk analysis model for EC development using a fuzzy set approach was proposed and incorporated into the FDSS [63]. Additionally, to assist the project manager in risk management, they had developed an application which will identify the risks involved during software development and predict the success or failure of the project using Artificial Neural Networks [69].

However, the fuzzy multiple regression analysis modelling techniques were used to identify the risk management techniques that are effective in reducing the occurrence of each software implementation risks [70]. Furthermore, 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 [71]. In addition, we propose a new technique by which we can study the impact of different control factors and different risk factors on software projects risk. The observation from the literature is summarized that most articles talking about methodology without structure techniques that very important to manage risk in a software project. It also found rarely studies link among software development lifecycle phases, risk management methodology and three categories of techniques discussed. Through studying the literature review, most articles are focused traditional techniques to identify software risk, but we believe, software project managers must use and combine among techniques throughout the software project lifecycle according to software risk methodology practice. On the other hand, some articles in quantitative risk techniques focused on one phase like test, maintenance, code, cost or so on and ignored remain phases with new techniques to estimate risk and analysis risk; perhaps managers used more techniques to mitigate risks. Indeed, this area needs more effort from scholars and researcher in quantitative and intelligent risk models.

Previous studies had shown that risk mitigation in software projects classified into three categories—namely, qualitative, quantitative, and intelligent approaches. Firstly, quantitative risk is based on statistical methods that deal with accurate measurement about risk or leading to a quantitative inputs, that helped to form a regression model to understand how software project risk factors influence project success [72]. Furthermore, qualitative risk techniques lead to subjective opinions expressed or self-judgment by software manager using techniques namely scenario analysis, Delphi analysis, brainstorming session, and other subjective approach to mitigate risks. Lastly, intelligent 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 logic models, fuzzy multiple regression analysis, neural network models, genetic algorithm, and heuristic algorithm. However, the prediction techniques as stepwise multiple regression analysis and fuzzy multiple regression analysis used to mitigate and model between software risks in software development lifecycle and risk management techniques in risk management approach. In addition, there is no software clearly to compute the fuzzy regression analysis and combine between linear and nonlinear technique [73]. Unfortunately, there is no software included mining and statistical techniques to mitigate risks in a software project [73]. Besides, we can use other techniques to manage and mitigate software project risks, such as neural network, genetic algorithm, Bayesian statistics, and other artificial intelligence techniques [74]. Finally, many authors defined risk management, but complex practice to measure the likelihood, of impact of software risks and determine appropriate risk management techniques, especially in software development projects [24].

Table 2: Analysis of the Quantitative and intelligent models for software risk management

No.	Paper	Type model	Techniques
1	Khanfar <i>et. al.</i> , [46]	Quantitative	Chi square and Spearman Correlation
2	Elzamly and Hussin [47]	Quantitative	Regression Analysis (Simultaneous selection procedure)

3	Elzamly and Hussin [49]	Quantitative	Stepwise Regression Analysis Techniques
4	Elzamly et al. [55], [75]	Quantitative	Discriminant Analysis Techniques
5	Elzamly and Hussin [58], [66], [73]	Quantitative and intelligent	Stepwise And Fuzzy Multiple Regression Analysis Techniques
6	N. V and K. Iyakutti [76]	Quantitative	Bayesian Statistics
7	Padayachee [54]	Quantitative	Approach
8	Carr and Tah [77]	Quantitative	Hierarchical Risk Breakdown Structure
9	Islam [78]	Quantitative	Systematic and straight-forward methods and techniques (Bayesian Belief Network)
11	Miler and Gorski [53]	Quantitative	Checklists and Group Effort
12	Miler and Gorski [56]	Quantitative	Risk Management Supporting Tool
13	Costa et al. [79]	Quantitative	RISICARE Tool
14	Elzamly and Hussin [59]	Quantitative	Approach
15	Elzamly and Hussin [60]	Quantitative	Factor Analysis Techniques
16	Ngai and Wat [63]	Intelligent	Fuzzy Set Approach
17	Conroy and Soltan [64]	Intelligent	Concurrent Simultaneous Engineering Resource View (ConSERV) as intelligent knowledge based
18	Büyüközkan and Ruan [67]	Intelligent	Special Fuzzy Operator
19	Dhlamini et al. [16]	Intelligent	Proposed Neural Network Approach and Intelligent Agents
20	Carr and Tah [80]	Intelligent	Fuzzy approach
21	Gandhi et al. [69]	Intelligent	Artificial Neural Networks
22	Christiansen et al. [62]	Quantitative	Multiple Logistic Regression
23	Jiang [81]	Intelligent	Artificial Neural Networks

6. Conclusions

Through studying the literature review, most articles are focused traditional techniques and models to identify software risk, but we believe, software project managers must use and combine among techniques throughout the software project lifecycle according to software risk methodology practice. On the other hand, some articles in quantitative or intelligent risk models focused on one phase like test, analysis, planning, design, maintenance, code, cost or so on and ignored remain phases with new techniques to estimate risk and analysis risk; perhaps software managers used more techniques to mitigate risks. Indeed, this area needs more effort from scholars and researchers in

quantitative and intelligent risk models. As future work, we can use more models useful to combine quantitative and intelligent risk models such as neural network, genetic algorithm and other artificial intelligence techniques to mitigate risks of cloud computing in banking organizations.

Acknowledgments

This work is supported by Welfare Association and Bank of Palestine (Zamalah Program) and the authors would like to thank Faculty of Information and Communication Technology, Universiti Teknikal Malaysia Melaka (UTeM) and Al-Aqsa University, Palestine.

References

- [1] W. Han and S. Huang, "An empirical analysis of risk components and performance on software projects," *J. Syst. Softw.*, vol. 80, no. 1, (2007), pp. 42–50, Jan.
- [2] A. Elzamy and B. Hussin, "An Enhancement of Framework Software Risk Management Methodology for Successful Software Development," *J. Theor. Appl. Inf. Technol.*, vol. 62, no. 2, (2014), pp. 410–423,
- [3] J. Sodhi and P. Sodhi, *IT Project Management Handbook. Management Concepts*, (2001).
- [4] J. Miler, "A Method of Software Project Risk Identification and Analysis," *Gdansk University of Technology*, (2005).
- [5] R. K. Bhujang and S. V., "Risk Impact Analysis across the Phases of Software Development," *Lect. Notes Softw. Eng.*, vol. 2, no. 3, (2014), pp. 282–287.
- [6] A. Kanane, "Challenges related to the adoption of Scrum," (2014).
- [7] T. Dyba and T. Dingsoyr, "Empirical studies of agile software development: A systematic review," *Inf. Softw. Technol.*, vol. 50, no. 9–10, pp. 833–859, Aug. (2008).
- [8] N. Gorla and S. Lin, "Determinants of software quality: A survey of information systems project managers," *Inf. Softw. Technol.*, vol. 52, no. 6, pp. 602–610, Jun. (2010).
- [9] M. Keil, P. Cule, K. Lyytinen, and R. Schmidt, "A framework for Identifying Software Project Risks," *Commun. ACM*, vol. 41, no. 11, pp. 76–83, (1998).
- [10] "Third Quarter Research Report," (2004).
- [11] J. Dominguez, "The CHAOS Report 2009 on IT Project Failure," (2009).
- [12] T. Clancy, "The standish group report ©," (1995).
- [13] W. Al-ahmad, K. Al-fagih, K. Khanfar, K. Alsamara, S. Abuleil, and H. Abu-salem, "A Taxonomy of an IT Project Failure: Root Causes," *Manage. Rev.*, vol. 5, no. 1, pp. 93–104, (2009).
- [14] IEEE, "Why Software Fails," *IEEE Spectrum Magazine*, (2009).
- [15] A. Kamaruddin, "Development of an Early Software Project Risk Assessment Application Using Case-Based Reasoning," *Universiti Teknologi MARA*, (2006).
- [16] J. Dhlamini, I. Nhamu, and A. Kachepa, "Intelligent Risk Management Tools for Software Development," in *Proceeding SACLA '09 Proceedings of the 2009 Annual Conference of the Southern African Computer Lecturers' Association*, (2009), pp. 33–40.
- [17] U. B. Varanasi and B. D. C. N. Prasad, "A Framework for Auditing the Evaluation of Uncertainty for Compliance in Information Systems," *Lect. Notes Softw. Eng.*, vol. 3, no. 1, pp. 40–43, (2015).
- [18] W. Ma, L. Liu, W. Feng, Y. Shan, and F. Peng, "Analyzing project risks within a cultural and organizational setting," in *ICSE Workshop on Leadership and Management in Software Architecture*, (2009), pp. 6–14.
- [19] S. Masticola, "A Simple Estimate of the Cost of Software Project Failures and the Breakeven Effectiveness of Project Risk Management," in *First International Workshop on the Economics of Software and Computation(ESC'07)*, (2007), p. 6.
- [20] A. Talet, R. Mat-zin, and M. Houari, "Risk Management and Information Technology Projects," *Int. J. Digit. Inf. Wirel. Commun.*, vol. 4, no. 1, pp. 1–9, (2014).
- [21] T. Merna and F. Al-thani, *Corporate Risk Management*, Second. John Wiley & Sons, Ltd, (2008).
- [22] M. Fan, N. Lin, and C. Sheu, "Choosing a project risk-handling strategy: An analytical model," *Int. J. Prod. Econ.*, vol. 112, no. 2, pp. 700–713, Apr. (2008).
- [23] R. Dash and R. Dash, "Risk Assessment Techniques for Software Development," *Eur. J. Sci. Res.*, vol. 42, no. 4, pp. 629–636, (2010).
- [24] P. Bannerman, "Risk and risk management in software projects: A reassessment," *J. Syst. Softw.*, vol. 81, no. 12, pp. 2118–2133, Dec. (2008).
- [25] SQAS, "Software Risk Management A Practical Guide.," (2000).
- [26] S. Alhawari, F. Thabtah, L. Karadsheh, and W. Hadi, "A Risk Management Model for Project Execution," in *The 9th IIBIMA Conference on Information Management in Modern Organizations*, (2008), pp. 887–893.

- [27] V. Holzmann and I. Spiegler, "Developing risk breakdown structure for information technology organizations," *Int. J. Proj. Manag.*, vol. 29, no. 5, pp. 537–546, Jun. (2010).
- [28] K. Bakker, A. Boonstra, and H. Wortmann, "Does risk management contribute to IT project success? A meta-analysis of empirical evidence," *Int. J. Proj. Manag.*, vol. 28, no. 5, pp. 493–503, Jul. (2010).
- [29] C. R. Pandian, *Applied software risk management: A guide for software project managers*. Auerbach Publications is an imprint of the Taylor & Francis Group, (2007).
- [30] J. Kontio, "Risk Management in Software Development: A Technology Overview and the Riskit Method," in *ICSE '99 Proceedings of the 21st international conference on Software engineering*, (1999), pp. 679–680.
- [31] P. Bannerman, "Managing Structure-Related Software Project Risk: A New Role for Project Governance," in *21st Australian Software Engineering Conference*, (2010), pp. 129–138.
- [32] R. Kumar, "Managing risks in IT projects: an options perspective," *Inf. Manag.*, vol. 40, no. 1, pp. 63–74, Oct. (2002).
- [33] K. Blackmore and K. Nesbitt, "Identifying Risks for Cross-Disciplinary Higher Degree Research Students," in *ReproductionProc. Tenth Australasian Computing Education Conference (ACE2008)*, Wollongong, Australia, (2008), pp. 43–52.
- [34] Oracle, "A Standardized Approach to Risk Improves Project Outcomes and Profitability," (2010).
- [35] J. Persson, L. Mathiassen, J. Boeg, T. Madsen, and F. Steinson, "Managing Risks in Distributed Software Projects: An Integrative Framework," *IEEE Trans. Eng. Manag.*, vol. 56, no. 3, pp. 508–532, Aug. (2009).
- [36] T. Arnuphaptrairong, "Top Ten Lists of Software Project Risks: Evidence from the Literature Survey," in *proceedings of the International multi conference of Engineers and Computer Scientists 2011 (IMECS 2011)*, (2011), p. 6.
- [37] A. Elzamly and B. Hussin, "Identifying and Managing Software Project Risks with Proposed Fuzzy Regression Analysis Techniques : Maintenance Phase," in *2014 Conference on Management and Engineering (CME2014)*, (2014), no. Cme, pp. 1868–1881.
- [38] N. Rout, N. Parida, and S. Mishra, "Software Risk Management and Mitigation Model," *Int. J. Appl. or Innov. Eng. Manag.*, vol. 2, no. 7, pp. 545–549, (2013).
- [39] 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).
- [40] H. Hoodat and H. Rashidi, "Classification and Analysis of Risks in Software Engineering," *Eng. Technol.*, vol. 56, no. 32, pp. 446–452, (2009).
- [41] F. Reyes, N. Cerpa, A. Candia, and M. Bardeen, "The optimization of success probability for software projects using genetic algorithms," *J. Syst. Softw.*, vol. 84, no. 5, pp. 775–785, May (2011).
- [42] M. Fakhar, M. Abbas, and M. Waris, "Risk Management System for ERP Software Project," in *Science and Information Conference 2013*, (2013), pp. 223–228.
- [43] T. Ernawati, Suhardi, and D. Nugroho, "IT Risk Management Framework Based on ISO 31000:2009," in *International Conference on System Engineering and Technology (ICSET)*, (2012), pp. 1–8.
- [44] B. Boehm and P. Bose, "A Collaborative Spiral Software Process Model Based on Theory W," in *Proceedings, 3 rd International Conference on the Software Process, Applying the Software Process*, IEEE, (1994), pp. 59–68.
- [45] B. Boehm, "Software Risk Management: Principles and Practices," *IEEE Softw.*, vol. 8, no. 1, pp. 32–40, (1991).
- [46] K. Khanfar, A. Elzamly, W. Al-Ahmad, E. El-Qawasmeh, K. Alsamara, and S. Abuleil, "Managing Software Project Risks with the Chi-Square Technique," *Int. Manag. Rev.*, vol. 4, no. 2, pp. 18–29, (2008).
- [47] 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).
- [48] 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).
- [49] 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).
- [50] A. Azari, N. Mousavi, and S. Mousavi, "Risk assessment model selection in construction industry," *Expert Syst. Appl.*, vol. 38, no. 8, pp. 9105–9111, Dec. (2011).
- [51] A. Elzamly, B. Hussin, S. Naser, and M. Doheir, "Predicting Software Analysis Process Risks Using Linear Stepwise Discriminant Analysis : Statistical Methods," *Int. J. Adv. Inf. Sci. Technol.*, vol. 2015, no. June, pp. 108–115, (2015).
- [52] H. Costa, M. Barros, and G. Travassos, "A Risk Based Economical Approach for Evaluating Software Project Portfolios," in *Proceeding EDSER '05 Proceedings of the seventh international*

- workshop on Economics-driven software engineering research, **(2005)**, no. 1, pp. 1–5.
- [53] J. Miler and J. Górski, “Risk-driven Software Process Improvement - a Case Study,” in 11th European Software Process Improvement Conference EuroSPI’2004, **(2004)**, no. 4, pp. 1–7.
- [54] K. Padayachee, “An Interpretive Study of Software Risk Management Perspectives,” in annual research conference of the South African institute of computer scientists and information technologists on Enablement through technology, **(2002)**, pp. 118–127.
- [55] A. Elzamly, B. Hussin, S. S. A. Naser, and M. Doheir, “Classification of Software Risks with Discriminant Analysis Techniques in Software planning Development Process,” *Int. J. Adv. Sci. Technol.*, vol. 81, no. 2015, pp. 35–48, **(2015)**.
- [56] J. Miler and J. Górski, “Implementing risk management in software projects,” in 3rd National Conference on Software Engineering, **(2001)**, pp. 1–10.
- [57] A. Elzamly and B. Hussin, “Mitigating Software Maintenance Project Risks with Stepwise Regression Analysis Techniques,” *J. Mod. Math. Front.*, vol. 3, no. 2, pp. 34–44, **(2014)**.
- [58] A. Elzamly and B. Hussin, “Evaluation of Quantitative and Mining Techniques for Reducing Software Maintenance Risks,” *Appl. Math. Sci.*, vol. 8, no. 111, pp. 5533–5542, **(2014)**.
- [59] A. Elzamly, B. Hussin, and N. Salleh, “Methodologies and techniques in software risk management approach for mitigating risks: A review,” *Asian J. Math. Comput. Res.*, vol. 2, no. 4, pp. 184–198, **(2015)**.
- [60] A. Elzamly and B. Hussin, “Classification and identification of risk management techniques for mitigating risks with factor analysis technique in software risk management,” *Rev. Comput. Eng. Res.*, vol. 2, no. 1, pp. 22–38, **(2015)**.
- [61] A. Elzamly and B. Hussin, “Estimating Stepwise and Fuzzy Regression Analysis for Modelling Software Design Project Risks,” *Asian J. Math. Comput. Res.*, vol. 3, no. 4, pp. 234–241, **(2015)**.
- [62] T. Christiansen, P. Wuttidittachotti, S. Prakanchoen, and S. A. Vallipakorn, “PREDICTION OF RISK FACTORS OF SOFTWARE DEVELOPMENT,” *ARPJ. Eng. Appl. Sci.*, vol. 10, no. 3, pp. 1324–1331, **(2015)**.
- [63] E. Ngai and F. Wat, “Fuzzy decision support system for risk analysis in E-commerce development,” *Decis. Support Syst.*, vol. 40, no. 2, pp. 235–255, Aug. **(2005)**.
- [64] G. Conroy and H. Soltan, “ConSERV, a project specific risk management concept,” *Int. J. Proj. Manag.*, vol. 16, no. 6, pp. 353–366, Dec. **(1998)**.
- [65] 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)**.
- [66] A. Elzamly and B. Hussin, “A Comparison of Fuzzy and Stepwise Multiple Regression Analysis Techniques for Managing Software Project Risks : Implementation Phase,” *Int. Manag. Rev.*, vol. 10, no. 1, pp. 43–54, **(2014)**.
- [67] G. Büyükoçkan and D. Ruan, “Choquet integral based aggregation approach to software development risk assessment,” *Inf. Sci. (Ny)*, vol. 180, no. 3, pp. 441–451, Feb. **(2010)**.
- [68] D. Houston, G. Mackulak, and J. Collofello, “Stochastic simulation of risk factor potential effects for software development risk management,” *J. Syst. Softw.*, vol. 59, no. 3, pp. 247–257, Dec. **(2001)**.
- [69] A. Gandhi, A. Naik, K. Thakkar, and M. Gahirwal, “Risk Management in Software Development using Artificial Neural Networks,” *Int. J. Comput. Appl.*, vol. 93, no. 19, pp. 1–7, **(2014)**.
- [70] A. Elzamly and B. Hussin, “Modelling and mitigating Software Implementation Project Risks with Proposed Mining Technique,” *Inf. Eng.*, vol. 3, no. 2014, pp. 39–48, **(2014)**.
- [71] A. Elzamly and B. Hussin, “Managing Software Project Risks (Analysis Phase) with Proposed Fuzzy Regression Analysis Modelling Techniques with Fuzzy Concepts,” *J. Comput. Inf. Technol.*, vol. 22, no. 2, pp. 131–144, **(2014)**.
- [72] V. Bhoola, S. Hiremath, and D. Mallik, “Identifying and Evaluation of Risks in Software Projects in Indian IT Industries,” *Int. J. Res. Bus. Technol.*, vol. 4, no. 1, pp. 346–353, **(2014)**.
- [73] A. Elzamly and B. Hussin, “A Comparison of Stepwise And Fuzzy Multiple Regression Analysis Techniques for Managing Software Project Risks : Analysis Phase,” *J. Comput. Sci.*, vol. 10, no. 10, pp. 1725–1742, **(2014)**.
- [74] A. Elzamly and B. Hussin, “Modelling and Evaluating Software Project Risks with Quantitative Analysis Techniques in Planning Software Development,” *J. Comput. Inf. Technol.*, vol. 23, no. 2, pp. 123–139, **(2015)**.
- [75] A. Elzamly, B. Hussin, S. S. A. Naser, and M. Doheir, “Predicting Software Analysis Process Risks Using Linear Stepwise Discriminant Analysis : Statistical Methods,” *Int. J. Adv. Inf. Sci. Technol.*, vol. 38, no. 38, pp. 108–115, **(2015)**.
- [76] N. V and K. Iyakutti, “Bayesian Statistics Software Approach For Risk Control on Complex Software Projects*,” *Int. J. Eng. Res. Technol.*, vol. 3, no. 12, pp. 1062–1068, **(2014)**.
- [77] J. Tah and V. Carr, “Towards a framework for project risk knowledge management in the construction supply chain,” *Adv. Eng. Softw.*, vol. 32, no. 10–11, pp. 835–846, Oct. **(2001)**.
- [78] S. Islam, “Software Development Risk Management Model- a goal-driven approach,” *Technische Universität München Institut für Informatik*, **(2009)**.

- [79] H. Costa, M. Barros, and G. Travassos, "A risk based economical approach for evaluating software project portfolios," *ACM SIGSOFT Softw. Eng. Notes*, vol. 30, no. 4, p. 1, Jul. **(2005)**.
- [80] V. Carr and J. Tah, "A fuzzy approach to construction project risk assessment and analysis: construction project risk management system," *Adv. Eng. Softw.*, vol. 32, no. 10–11, pp. 847–857, Oct. **(2001)**.
- [81] H. Jiang, "The Application of Artificial Neural Networks in Risk Assessment on High-Tech Project Investment," in *2009 International Conference on Business Intelligence and Financial Engineering*, **(2009)**, pp. 17–20.