A Survey of Project Risk Assessment and Estimation Models

Taher Ahmed Ghaleb, Abdulrhman Ahmed Alsri, Loay Shabaneh, Mahmood Niazi

***Abstract*—Risk is a potential event that leads to loss or harm in software projects. Risks may be classified into negative or positive; where negative risks specifically lead to loss or harm, while positive risks represent a new opportunity in the project. To handle these kinds of risks, risk assessment models and techniques have been introduced. In this paper, we review the most popular and applicable risk assessment models available in the literature. We come up with a taxonomy in which those models can be categorized as: (1) Artificial Intelligence (AI) based, (2) Classical (or Non-AI based), and (3) other Hybrid models. We propose evaluation criteria which have been used to compare these models. After analyzing evaluation results, we recommend suitable models which can be used to avoid project risks.**

Index Terms—assessment models, risk assessment, risk estimation, risk identification, risk management

I. Introduction

Risks in software projects are variables or changes that lack the stability of those projects [1]. In general, risks have 9 critical elements: product requirement, software complexity, project staff, software Usability, targeted reliability, estimation method, monitoring method, development process adopted and tools [5].

Researchers in the literature proposed various risk management strategies that should be implemented to tackle those risks. Risk management consists of two main steps: risk assessment and risk control. In this paper, we focus on the risk assessment which itself, as shown in Fig. 1, has three subsidiary units: risk identification, risk analysis and risk classification and prioritization. Also, we present some models that estimate and rank risks and their uncertainty.

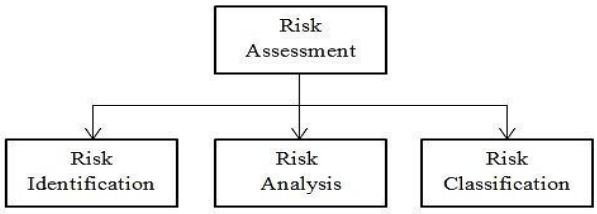

Fig 1 Risk Assessment Units

Manuscript received March 6, 2014; revised April 1, 2014. This work is supported by the Deanship of Scientific Research at King Fahd University of Petroleum and Minerals, Kingdom of Saudi Arabia (KSA), under Research Grant 11-INF2152-04.

T. A. Ghaleb is with the Information and Computer Science Department, King Fahd University of Petroleum and Minerals, Dhahran, KSA (e-mail: g201106210@kfupm.edu.sa).

A. A. Alsri is with the Information and Computer Science Department, King Fahd University of Petroleum and Minerals, Dhahran, KSA (e-mail: g201102450@kfupm.edu.sa).

L. Shabaneh is with the Information and Computer Science Department, King Fahd University of Petroleum and Minerals, Dhahran, KSA (e-mail: g201104290@kfupm.edu.sa).

M. Niazi is with the Information and Computer Science Department, King Fahd University of Petroleum and Minerals, Dhahran, KSA and Riphah International University Islamabad, Pakistan (e-mail: mkniazi@kfupm.edu.sa).

To this end, several models were proposed to assess the risks which may occur in software projects. These models employed different techniques for applying qualitative and quantitative risk assessment to risk factors of the software applications.

In this paper, we study models that have been proposed in the literature and present them in comparative taxonomy. We organize the rest of the paper as follows: section II reviews the works done in the literature. After that, we define some evaluation attributes in section III. Then, in section IV, we compare the studied models based on the evaluation attributes. After that, we analyze and discuss the comparative results and put some recommendations accordingly in section V. Finally, section VI, concludes the paper and suggests some future work.

II. Literature Review

A. Classical Models (Non-AI based Models)

1) Risk assessment based on questionnaire: Williams *et al*. [2] proposed a model for assessing risks which is a Taxonomy Questionnaire. They prepared a set of questions, and then they provide some answers to those questions. They compute the risk levels of each risk element to predict the possible outcomes of software projects accurately.

Later on, in 2000, Foo and Murganantham [5] proposed a risk assessment model, named SRAM (software risk assessment model), for software projects with the use of a predefined questionnaire. By considering nine risk factors, they developed a comprehensive questionnaire which contains a set of questions for each risk factor. The questionnaire was given to experts who are specialized in the risk assessment field. Finally, their model was tested based on historical data and it was observed that their model was able to predict the outcomes of software projects.

2) Risk assessment And estimation based on software metrics: In order to assess risks that might occur in a project, Hyatt and Rosenberg [3], introduced software metrics for assessing project risks. They defined specific quality attributes and goals based on their importance in developing software projects and their capability to be quantified. A core set of metrics was defined which relate to the software development process and products. Several measurements for metrics' usability and applicability were discussed. Gupta and Sadiq [4] presented SRAEM, which also provide software metrics with risk exposure for software risk assessment and estimation. In addition to the total cumulative risk, the proposed model has the ability of identifying the set of risks from each phase in the project development. The model

estimates the source of ambiguity and uncertainty using errors. Sadiq *et al*. [8] introduced SRAEP, Software Risk Assessment and Evaluation Process. SRAEP is based on a model based approach associated with the fault tree approach. These two approaches were used to identify the project risks.

3) Risk assessment models based on classification and prioritization: Qualitative and quantitative risk assessment of software projects was founded by Boehm [9]. This model can not classify the risk events based on their statistical (in)dependence. Instead, it can do so by using the decision tree to help in risk event classification based on their dependence. After Boehm's work, Fairley [10], recognize the statistical dependence of the risk events by taking into account some attributes in which risk events are related, such as size, time, etc. In 2011, Uzzafer [11] proposed a novel risk assessment model for classifying risk events of software projects qualitatively. This is based on their occurrence independence and statistical independence of their impacts. Additionally, it's capable of integration into the software cost estimation model in order to enhance its ability of generating cost estimates with the associated impacts of the project risk events.

4) Risk assessment based on estimation tools: Keshlaf and Hashim [13] developed, SoftRisk, a prototype tool for managing software risks. The design of this tool is based on a model proposed in the same paper called SERIM, Software Engineering risk model. This model focuses on technical, cost, and schedule risks. The problem with this model is that it does not consider the requirements and complexity issues. In 2010, Sadiq *et al*. [12] introduced a new architectural implementation using their proposed *esrcTool*. This tool is based on SRAEM [4] model. It is useful for two purposes: estimating the risk in the software, which what we focus on in this survey, and the other one is to estimate the cost of the software. They have applied the function point approach as an input parameter into the *esrcTool.*

B. Artificial Intelligence based Models

1) Artificial Neural Networks based(ANN): Artificial Neural Networks (ANN) approach is a machine learning technique that has learning abilities and it is helpful in solving problems with uncertain and poorly understood conditions. It can help humans in designing effective algorithms. ANN is constructed from a neuron processing element. Processing elements are connected by a network of connections where each connection is weighted and it builds the network knowledge. ANNs are used as pattern classifiers and memories. Software engineers have taken the advantage of ANN to identify risk in software development. Salvatore *et al*. [16] enhanced the existing risk management models by checking out the results of current approaches of risk management. They provide a way to compare historical risk data on the risks identified by similar projects with the risk found by each framework, based on direct queries to the stakeholders. Another study by Goonawardene *et al*. [18] which is to examine the effectiveness of using neural and fuzzy systems in the areas such as job recruiting, predicting of project success or failure and on decision making based on performance appraisal of employees. Kutlubay *et al*. [19] established a method for identifying software defects using machine learning methods.

2) Bayesian Belief Networks (BBNs): Bayesian Belief Networks represent probabilistic relationships between variables. BBNs enable reasoning under uncertainty and it is possible to articulate expert beliefs about the dependencies between different variables. They propagate the impact of evidence on the probabilities of uncertain outcomes, such as future system reliability [21]. Fenton and Neil [20] have shown that causal models using BBNs have many advantages over the classical approaches. Another study is done by Fenton and Neil [21] who proposed a model using BBNs and explained that a very complex problem is.

3) Fuzzy Logic Based: Software project risk assessment is not always just a matter of win or lose, most of the time, it is necessary to give some degree of hazard assessment to each identified risk. That's why we need fuzzy logic to give degrees to these risks. The fuzzy logic technique is used when we want to convert linguistic variables into numeric values between 0 and 1 according to an expert inference system. Many project risk assessment and identification models used this technique to assess risks in new software projects. Li *et al*. [22], introduced a model based on the fuzzy linguistic multiple attribute decision making. In this model, a group of experts give an assessment value of each risk assessment object based on predefined risk assessment criteria and set of linguistic terms. Then a triangular fuzzy number can be given for each linguistic assessment element that will formulate an assessment matrix for each expert assessment. From those formulations, a new estimated value of the previous triangular fuzzy numbers is calculated and sorted to give priority to the risks. In [23], a new risk assessment approach is proposed using Fuzzy Inference system. This approach uses Schmidt risk factors as the basis for risk assessment system. Manalif *et al.* [24] introduced a Fuzzy expert-COCOMO model which is capable to provide the estimated effort in addition to risk assessment activities.

C. Hybrid Models

Deursen and Kuipers [7] proposed a hybrid method which integrates software metrics with questionnaires. This method focuses on primary facts which are obtained by analyzing, automatically, the source code of a software with code metrics. Hu *et al*. [15] introduced a model for project risk evaluation using Neural Network (NN), support vector machine (SVM), and also genetic algorithm approaches. Several software risk factors are gathered from 30 experts and applied as an input to their model. They experimented their model on data collected via questionnaires, and found that SVM produced better results than NN. Accordingly, their neural network model is optimized by a genetic algorithm, to have a model that outperformed SVM in performance.

III. Evaluation Criteria

In order to effectively evaluate risk assessment models, a set of criteria has been defined by us to show the objectives and impacts of these models on project management. These criteria are then applied to the different models in order to compare their actual performance. We used the three sources for the development of evaluation criteria.

ISBN: 978-988-19252-7-5 (revised on 31 December 2014) WCE 2014
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

A. Criteria adapted from related work in the literature

We have studied several researches in the literature, other than studied in the paper so far, for the sake of defining some evaluation criteria that have been used before. Table I lists some evaluation criteria which are adapted from several researches in evaluating risk assessment models. These criteria are essential in measuring risk assessment model efficiency.

B. Our proposed criteria based on our review

Throughout our reviews and studying the current risk assessment models, we have observed some characteristics which these models have in common. In addition, some models are suitable in particular situations while the others are not. Table II describes some attributes or criteria that later will be used to evaluate the risk assessment models.

C. Assigning scores to the criteria values

Textual results may sometimes confuse the project manager of deciding which model is the best. So, we convert these textual values into some numerical scores (Table III) that may be aggregated together to have a concrete result [25].

IV. Comparison

Risk assessment models have been reviewed for their ability to manage risks in various software project environments. In this section, a detailed comparative study of the best risk assessment models has been conducted based on evaluation criteria that have been proposed in Section III.

In section II, we categorized risk assessment models into three major categories (classical models, Artificial Intelligence based models, and hybrid models). To do the comparison, we firstly made a tentative comparison of nine models (three models from each category) to have an initial idea about the characteristics of each model. After analyzing the results of that tentative comparison, we select the best model in each category to make the final comparison among three models. Table IV demonstrates the evaluation criteria after applying them on the selected best models. This evaluation was done by all authors as a team based on the evaluation criteria defined in Table I and Table II.

In addition, we applied the evaluation criteria again by using the scores defined in Table III. This comparison is demonstrated in Table V. We can see from Fig. 2 that FuzzyEx COCOMO has a higher score than the two others, which means that it is the preferable risk assessment model.

V. Discussion

Certain values were given to each model after a complete study of these models. In this section, we illustrate how these values were given to those models.

***EsrcTool*, proposed by Sadiq *et al.* [12]:**

- We have analyzed that decision making in EsrcTool is *qualitative*, the identifying uncertainty sources is *perfect*, and it can identify the uncertainty level with a *high accuracy*. Regarding risk factors, *esrcTool* was made based on SRAEM [4], a previously proposed model, which in turn was evaluated in [28] that it fully supports *10 risk factors*.
- We followed the experiments done with their paper and found that it is *Fully* automated as well as it has an applicable tool called *esrcTool* which uses some software metrics to make the assessment process.
- As it is demonstrated in the paper, it is easy to install, configure, and operate because it has a graphical user Interface designed by Java.
- As a case study, several projects chosen from graduation projects (developed by the Master students) to test this tool on them.
- We observed that the scope of *esrcTool* model is limited to only *small projects* and it is not recommended to be used in medium-size or large projects.
- In regards to the predictability feature, we noticed that it is *not supported* because we could not infer if EsrcTool employed the estimation module from SRAEM or not.
- We give a value of *low* in prioritizing risk hazards because it is not able to do so in an accurate way.

Goonawardene *et al.* [18]:

- It is considered as a *qualitative* decision making model because it is built in an artificial neural system which can predict the project successes and failures as well as to provide the appraisal of employee performance.
- They collected a set of three risk factors in terms of cost, time and quality from *questionnaires* and considered them as an input to their model.
- The model is then tested by feeding it by some data collected from around 30 projects as special case study.
- The desired output is prioritized in a way that it suggests whether the project is successful, challenged, or failed.
- It also identifies probable sources of uncertainty as it provides some recommendation for the unsuccessful projects to be better survived.
  Finally, the model was simulated by an automated prototype and used by project managers, they compared its results with manual processes and they were satisfied with the accuracy of the model which was around 90% as prediction capability.
- It is *easy* enough to use.

FuzzyEx COCOMO, introduced by Manalif *et al.* [24]:

- It is shown from the risk assessment results (of some projects that have been used in the case studies proposed in a thesis by the same author) that their model supports both the *qualitative* and *quantitative* decision results.
- This model has an integrated tool that pre-process inputs and produce results without any personal judgment from the experts (*Fully* automated). According to the thesis, the model was developed using MATLAB and there are some screen-shots which are taken for different modules of the tool.
- According to the paper of authors, there is 31 risk rules or factors used to determine the input for the model and it was able to generate the level of priority to each risk factor.
- Our opinion, based on our understanding after reviewing the paper and the related work, this model can identify every source of uncertainty. In addition, we believe that Fuzzy logic techniques helps in identifying the uncertainty level *accurately*.

ISBN: 978-988-19252-7-5 (revised on 31 December 2014) WCE 2014
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

TABLE I
CRITERIA ADAPTED FROM RELATED WORK IN THE LITERATURE

Criteria	Description + Usefulness	Possible Values
Decision Making [26] [27]	Type of decision to be taken on risk. Qualitative assessment models produce information about risks in verbal form which displays the probability of risks to be occurred in a range form (e.g. High, medium, or low). On the other hand, quantitative assessment models results in presenting risks a quantity or number related to the cost that might be increased or the time that might be extended (e.g. there will be a risk in project implementation that will cost 20,000$ and requires 1 month more). Hybrid models combine both qualitative and quantitative assessment. **Usefulness:** This criterion helps in choosing the suitable risk assessment model based on what kind of decision support a project manager desires.	Qualitative, Quantitative, Hybrid
Identifying sources of uncertainty [27]	Uncertainty is an ambiguity or a lack of information that may lead to raise some risks in a project. **Usefulness:** Identifying uncertainty sources strongly helps in estimating and predicting risks.	Perfect, Partial, None
Automation [6]	This criterion indicates whether the technique is fully automated which means that the technique does not need human help either in pre-processing or post-processing. Also, it could be Semi-automated which means that there is some parts of risk assessment need manual human processing. The model may never been automated when it is done manually. **Usefulness:** Computerized models and techniques save users time and they usually produce more accurate output compared to manually ones.	Fully-automated, Semi-automated, Manual
Tool support [6]	Indicate whether a model has a tool support or not. This criterion has been identified because a model may be automated but has no tool. **Usefulness:** Tools are helpful to implement models and techniques easily.	Yes, No
Input Data [27]	Determines the type of the input data that will be processed by the model in order to assess risk. **Usefulness:** When selecting a risk assessment model, it is essential to select the model which is capable to deal with the data that a project has.	Questionnaire, errors Software metrics, Function point
Case Study [6]	This criterion describes whether the model has been practically evaluated by real projects. **Usefulness:** Project managers may prefer selectingthe model that has already been applied on and evaluated by real projects.	Yes, No
Number of Risk Factors [14]	Number of potential sources of the risk examined and assessed by the model. **Usefulness:** It gives an indication about how much details the risk assessment model is able to analyze the project risk.	Numeric values (1, 2,)
Predictability [17]	This criterion evaluates the model's ability of estimating and predicting potential risks of a project. **Usefulness:** For the project managers who are seeking for a model which can predict the potential risks that may occur in a project, this criterion will support them to choose.	Yes, No
identifying uncertainty level [17]	This criterion evaluates the model's ability of identifying the level of uncertainty which may lead to risks. **Usefulness:** Project managers always look for a model that solves the uncertainty problem in an accurate way.	High, Medium, Low

TABLE II
OUR PROPOSED CRITERIA BASED ON OUR REVIEW

Criteria	Description + Usefulness	Possible Values
Risk Hazards Prioritization	This criterion determines the level of the model's ability of prioritizing the risk factors. **Usefulness:** Helps project managers to take more consideration as well as more planning and control over critical risks.	High, Low Medium,
Simplicity	Describes whether the model is easy to be applied and adapted or not. **Usefulness:** It encourages many project managers to use the model which is more simple, it also supports more mangers to apply it and get benefit from it.	Simple, Complex
Scope (Project Size)	Some models are suitable to be applied on specific kind of projects (e.g. small projects) while the others are general-purpose. **Usefulness:** The project managers can choose a specific model according to the size of their initiated projects.	Large, Small, Medium

TABLE III
ASSIGNING SCORES TO THE CRITERIA VALUES

Criteria	Possible Scores
Decision Making	Qualitative = 1, Quantitative = 1, Hybrid = 2
Identifying sources of uncertainty	Perfect = 2, Partial = 1, None = 0
Automation	Fully-automated = 3, Semi-automated = 2, Manual = 1
Tool support	Yes = 1, No = 0
Input Data	Questionnaire = 1, Software metrics = 1, Function point errors = 1
Case Study	Yes = 1, No = 0
Number of Risk Elements	[1..5] = 1, [6..10] = 2, [11.15] = 3, [16..20] = 4, [20...] = 5
Predictability	Yes = 1, No = 0
Risk Hazards Prioritization	High = 3, Medium = 2, Low = 1
Simplicity	Simple = 1, Complex = 0
Scope	Large projects = 6, Medium projects = 3, Small projects = 2
Identifying uncertainty level	High = 3, Medium = 2, Low = 1

ISBN: 978-988-19252-7-5 (revised on 31 December 2014)
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

TABLE IV
COMPARING THE BEST THREE BEST RISK ASSESSMENT MODELS

	Technique based	Decision Making	Identifying sources of uncertainty	Automation	Tool support	Input Data	Case Study	Number of Risk Factors	Predictability	Risk Hazards Prioritization	Identifying uncertainty level	Simplicity	Scope (project size)
esrcTool [12]	Classical "Non-AI"	Qualitative	Perfect	Fully-Automated	Yes	Function point errors + Software metrics	Yes	10	No	Low	High	Simple	Small
Goonawardene et al. [18]	Neural Networks	Qualitative	Perfect	Semi-Automated	No	Questionnaire	Yes	19	Yes	Medium	High	Simple	Small and Large
FuzzyEx COCOMO [24]	Fuzzy Logic	Qualitative + Quantitative	Perfect	Fully-Automated	Yes	Risk factors	Yes	More 21	Yes	High	High	Complex	Large

TABLE V
COMPARING THE MODELS WITH VALUE SCORES

	Technique based	Decision Making	Identifying sources of uncertainty	Automation	Tool support	Input Data	Case Study	Number of Risk Factors	Predictability	Risk Hazards Prioritization	Identifying uncertainty level	Simplicity	Scope (project size)	Total
esrcTool [12]	Classical	1	2	3	1	2	1	2	0	1	3	1	2	19
Goonawardene et al. [18]	Neural Networks	1	2	2	0	1	1	4	1	2	3	1	8	26
FuzzyEx COCOMO [24]	Fuzzy Logic	2	2	3	1	1	1	5	1	3	3	0	6	28

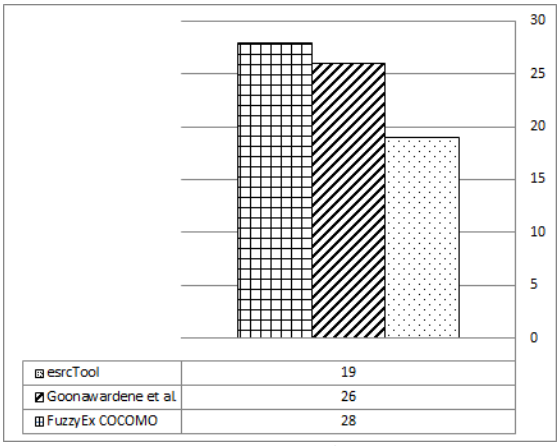

Fig 2 A graphical representation of the comparison results

- Based on the information provided in the paper, the author emphasized estimates for the total project risk and cost contingency effort estimation.
- The scope of this model is *large* companies and organizations. If applied on small projects, it will not produce better results. We claim this, because we found that the author mentioned in the related thesis that three datasets have been used to evaluate the model (*NASA'93 poject, 6 Industrial projects, and 12 projects of Turkish software development company*) as *case studies*.
- Based on our analysis, the model proposed is very complicated and needs lots of training.

ISBN: 978-988-19252-7-5 (revised on 31 December 2014) WCE 2014
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

VI. Conclusion

A comprehensive study of project risk assessment models has been conducted in order to introduce work that contributes to the state-of-the-art risk assessment. We review the most beneficial and applicable models that have been introduced in the literature. In this paper, we present a taxonomy in which those models were categorized as different groups. We defined several evaluation attributes in which these models were compared. After analyzing the evaluation results of three models, we recommend the FuzzyEx COCOMO model is the most suitable model to be applied to software projects in order to assess the potential risks that might occur.

Acknowledgment

T. A. Ghaleb thanks his home institution, Taiz University, which grants him a donation during his Master program at King Fahd University of Petroleum and Minerals.

References

[1] S. S. M. Fauzi, N. Ramli, M. Nasir, *Assessing Software Risk Management practices in a small scale project*, In Information Technology, ITSim, International Symposium, Volume 4, pages 1-5, IEEE, August 2008.

[2] R. C. Williams, G. J. Pandelios, S. G. Behrens, *Software Risk Evaluation (SRE) Method description (Version-2.0)*, Technical report December-1999.

[3] L. Hyatt, L. Rosenberg, *A Software Quality Model Metrics For Risk Assessment*, European Space Agency Software Assurance Symposium, 1996.

[4] D. Gupta, M. Sadiq, *Software Risk Assessment and Estimation Model*, International Conference on Computer Science and Information Technology, 2008. ICCSIT '08, pages 963-967, August 2008 - September 2008.

[5] S. W. Foo, A. Muruganantham, *Software risk assessment model*, In Proceedings of the 2000 IEEE International Conference on Management of Innovation and Technology, Volume2, pages 536-544, IEEE, 2000.

[6] I. A. Qureshi, A. Nadeem, *GUI Testing Techniques: A Survey*, International Journal of Future Computer and Communication, Vol. 2, No. 2, April 2013.

[7] T. Deursen, A. V. Kuipers, *Source-Based Software Risk Assessment*, In: ICSM 2003: Proceedings of the International Conference on Software Maintenance. IEEE Computer Society, Los Alamitos (2003).

[8] M. Sadiq, M. K. I. Rahmani, M. W. Ahmad, S. Jung, *Software Risk Assessment and. Evaluation Process (SRAEP) using model based approach*, 2010 International Conference on Networking and Information Technology (ICNIT) , pages 171-177, June 2010.

[9] W. B. Boehm, *Software Risk Management: Principles and Practices*, IEEE Software, pages 33-40, January 1991.

[10] R. Fairley, *Risk Management for Software Projects*, IEEE Software, volume 11, issue 3, pages 57 67, May 1994.

[11] M. Uzzafer, *A Novel Risk Assessment Model for Software Projects*, 2011 International Conference on Computer and Management (CAMAN), pages 1-5, 19-21, May 2011.

[12] M. Sadiq, A. Rahman, S. Ahmad, M. Asim, J. Ahmad, *esrcTool: a tool to estimate the software risk and cost*, In: IEEE second international conference on computer research and development, pages 886890, 2010.

[13] A. A. Keshlaf, K. Hashim, *A model and prototype tool to manage software risks*, in the 1st Asia pacific conference on software quality, pages 297305, IEEE 2000.

[14] N. Bajpai, *Business research methods*, Pearson Education India, 2011

[15] Y. Hu, J. Huang, J. Chen, M. Liu, K. Xie, *Software project risk management modeling with neural network and support vector machine approaches*. In Natural Computation, 2007, ICNC, Third International Conference, Volumne no. 3, pages 358-362, IEEE, August 2007.

[16] S. A. Sarci, G. C., V. R. Basili A *STATISTICAL NEURAL NETWORK FRAMEWORK FOR RISK MANAGEMENT PROCESS From the Proposal to its Preliminary Validation for Efficiency*, ICSOFT (SE), pages 168-177, INSTICC Press, 2007.

[17] A. Klinke, O. Renn, , *A New Approach to Risk Evaluation and Management: Risk-Based, Precaution-Based, and Discourse-Based Strategies1*, Risk analysis, 22(6), 1071-1094, 2002.

[18] N. Goonawardene, S. Subashini, N. Boralessa, L. Premaratne, *A Neural Network Based Model for Project Risk and Talent Management*, ISNN (2), Lecture Notes in Computer Science, Volume no. 6064, pages 532-539, Springer, 2010.

[19] O. kutlubay, A. Bener, *A Machine Learning Based Model For Software Defect Prediction*, working paper, Bogazici University, Computer Engineering Department, 2005.

[20] Fenton and Neil, *Software Metrics and Risk*, Second Earopean Measurmenet Conference, 1999.

[21] Fenton and Neil, *A Critique of Sftware Defect rediction Models*, IEEE Transaction on Software Engineering, volume no. 25, NO. 5, 1999.

[22] Y. Li, N. Li, *Software project risk assessment based on fuzzy linguistic multiple attribute decision making*, IEEE International Conference on Grey Systems and Intelligent Services, pages 1163-1166, November 2009.

[23] Iranmanesh, S. Hossein, S. B. Khodadadi, S. Taheri, *Risk Assessment of Software Projects Using Fuzzy Inference System*, Computers and Industrial Engineering, CIE, International Conference on IEEE, 2009.

[24] E. Manalif, L.F. Capretz and D. Ho, *Software Project Risk Assessment and Effort Contingency Model Based on COCOMO Cost Factors*, Journal of Computations & Modeling (JCM), Volume 3, Issue 1, pp. 113-132, International Scientific Press Ltd., March 2013.

[25] M Niazi, An instrument for measuring the maturity of requirements engineering process, The 6th International Conference on Product Focused Software Process Improvement, (pp 574-585), LNCS. 2005.

[26] A. M. Sharif, S. Basri, A Study on Risk Assessment for Small and Medium Software Development Projects, International Journal of New Computer Architectures and Their Applications(IJNCAA), 1(2), 325, ISO 690, 2011.

[27] K. Georgieva, A. Farooq, R. R. Dumke, *Analysis of the Risk Assessment Methods: A Survey*, In Software Process and Product Measurement, Springer Berlin Heidelberg, (pp. 76-86), ISO 690, 2009.

[28] Y. Bazaz, S. Gupta, O. PrakashRishi, L. Sharma, *Comparative study of risk assessment models corresponding to risk elements*. In Advances in Engineering, Science and Management (ICAESM), 2012 International Conference on (pp. 61-66). IEEE. 2012.

The paper has been revised on December 30, 2014. The modification includes fixing one reference with the correct citation, and removing another uncited reference. In addition, values of the last three columns in Table V have been fixed correctly.