International Journal on New Computer Architectures and Their Applications (IJNCAA) 1(2): 325-335
The Society of Digital Information and Wireless Communications, 2011 (ISSN: 2220-9085)

A Study on Risk Assessment for Small and Medium Software Development Projects

Abdullahi Mohamud Sharif and Shuib Basri
Universiti Teknologi PETRONAS,
Bandar Seri Iskandar, 31750 Tronoh, Perak, Malaysia.
saakuut@kudhan.so, shuib_basri@petronas.com.my
http://www.utp.edu.my

ABSTRACT

Generally, in software development, there are large, medium, and small software projects that each of them can be affected or influenced by a risk. Therefore, it requires a distinctive assessment process of the potential risks that may cause failure or loss of the project whenever they occur. From the literature, there are actually number of risk assessment researches conducted toward software projects. However, there are at least view researches being focused on risk assessment of small and medium software projects. These particularly results in a gap to the risk assessment research area that may lead to a lot of small and medium project not having risk assessment. For that reason, the main target of the article is to give researchers an insight on the current level of risk assessment for small and medium software development projects. Finally, some future directions will be discussed hoping to insight the gap in the risk assessment field for small and medium software development projects.

KEYWORDS

Small and Medium Software Development Projects, Software Risk Assessment, Risk Management, Small and Medium Enterprise, Software Development.

1 INTRODUCTION

Risks are essential factor for the development of software projects currently and also by its effects lots of projects been unsuccessful. In [5], risk is described as *"the possibility of suffering loss that describes the impact on the project which could be in the form of poor quality of software solution, increased costs, failure, or delayed completion"*. Furthermore, all projects share some extent of risk, and the majority of Information Technology (IT) projects have *considerable* risks [6]. Risk can, however, be *reduced* [6], *stewarded* [7], and managed in accordance with tight planning and assessment.

Moreover, according to [8], risk management is divided up into risk assessment and risk control. The risk assessment is divided into three sub levels which are risk identification, risk analysis, and risk prioritization. The second part of risk management, risk control, is also broken down into risk management planning, risk resolution, and risk monitoring.

On the other hand, software development projects are divided into large, medium, and small projects which their definition is based on the number of Lines of Code (LOC), duration of the project, and number of developers of the project. In the context of software development projects, small and medium software development projects (SMSDP) are identified as projects that have 50000-100000 LOC [2], 6-12 months, and ten or fewer programmers [3]. According to [21], Software risk assessment is defined as *"A process of identifying, analyzing, and prioritizing risks"*. On other hand, Small and medium projects are expanding rapidly in the world as they are taking part in the economic growth of each country. According to [4], "*Small projects typically carry the same or more risk as do large projects.* [While] *many customers and millions of dollars are lost each year on small projects in product and service organizations*".

From that perspective of risk management and software development classification, we will focus our paper particularly on risk assessment level for small and medium software development projects. On the other hand, the main objective of this review is to present research workers an understanding of the current level of risk assessment for SMSDP. Additionally, the paper provides information about the different sorts of risk assessment models and methods that found in the literature based on the context of risk assessment for SMSDP.

In this paper, research was structured as follows: section 2 gives overview of the review process, section 3 explains current risk assessments in SMSDPs, section 4 presents comprehensive analysis, and finally section 5 summarizes the review.

2 REVIEWING PROCESS

We have taken various Internet searches for getting information about researches in the direction of SMSDP risk assessment. We broke down the investigation into two phases. In the first phase, we have explored risk assessment for SMSDPs only, and the second phase, we have searched software risk assessment without concentrating whether its toward small, medium, or large projects. We found a quite number of researches those their focal point was on this domain, but most of them toward large software projects. However, after adept research, we ended up an overall total of 12 researches for the domain of software risk assessment with regard to both aforementioned phases. Therefore, we have combined the two phase outcomes as we analyzed each of them within their components of SMSDP's focus.

Moreover, the explored researches are in the time span of the last decade. As shown in figure 1, only 3 researches were their center of attention toward software risk assessment in the first half of last decade. Despite the fact that 9 researches are in the direction of software risk assessment for the second half of last decade. That means, as its clear in the picture, the research toward software risk assessment is rising leisurely.

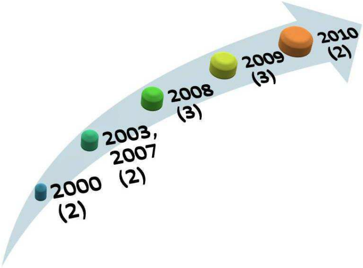

Fig. 1. SMSDP Risk Assessment Timeline

On the other hand, the founded researches were divided based on their proposed outcome into two categories:

1. Models category: are the type of researches provide a process model to assess risk.
2. Methods category: are those researches their outcome is method e.g. fuzzy logic method, etc.

Finally, the studied researches with their information of inputs, methods, and outcomes are going to be analyzed and discussed deeply in the next sections.

3 CURRENT RISK ASSESSMENTS IN SMSDP

Within this section we divide and analyze each one of the previously mentioned assessment ways for SMSDPs based on the following models and methods categories.

3.1 Models Category

There are a quite number of models in the literature, which made use of different procedures or algorithms to assess software risks in general, although some of these prototyped, a tool as a proof of concept utilization.

In this section, we review the literature of 6 models with their explanation. The explanation includes the model focus, proposes of the model, a brief description of the model, inputs of the proposed model, risk ranking approach of the model, decision analysis taking types of the model, and if the model implements a proof of concept prototype tool. The detailed information for the contribution of each model is summarized in below

Model [1]

1. *Focus:* Assessment, treatment, and monitoring automatically risks related in project time management for small and medium software development projects, such as errors in estimating time
2. *Proposes:* Risk Assessment Tool (RAT) model
3. *Description:* RAT model consists 5 interconnected phases: users, project plan input, risk rules which contains risk ranking matrix, risk conditions, and risk scenarios, risk fetching processes, and risk report. The risk assessment is taken in the early phases of the project
4. *Inputs:* Project plan (e.g. Work Breakdown Structure (WBS)) and resources
5. *Risk Ranking:* Risks are ranked based on risk rank matrix which contains risk category (1-Unknown, 2-Low, 3-Medium, 4-High, 5-Fatal), probability of occurrence, and risk impact (1-Low, 2-Medium, 3-High). The matrix produces 45 ranks for risk.
6. *Decision Taking Types:* Hybrid assessment
7. *Prototype:* Implemented a web application prototype.

Model [9]

1. *Focus:* Risk assessment and estimation of software projects
2. *Proposes:* Software Risk Assessment And Estimation Model (SRAEM)
3. *Description:* The model takes inputs to estimate efforts, cost, and risk exposures. Then the risk prioritization and ranking is

taken after applying Mission Critical Requirements Stability Risk Metrics (MCRSRM) if there is no changes in the requirements after requirement analysis
4. *Inputs:* Measurement, model, and assumption errors using the concept of Function point
5. *Risk Ranking:* The estimation and ranking risks is done by using two methods: probability by using risk exposure, and software metrics of risk management based on MCRSRM
6. *Decision Taking Types:* Quantitative assessment
7. *Prototype:* --

Model [10]

1. *Focus:* Risk assessment of software projects
2. *Proposes:* Software risk assessment model
3. *Description:* The model is based on Grey Theory using Analytic Hierarchy Process (AHP) method and entropy method. In the result of the assessment, the author suggests to study further to determine the major software risk factors
4. *Inputs:* Risk of demand analysis, project quality, project schedule, project circumstance, technology and project personnel.
5. *Risk Ranking:* In weighting of risk index, the research uses a combination of two methods: subjective method (e.g. AHP), and objective method (e.g. entropy method)
6. *Decision Taking Types:* Quantitative assessment
7. *Prototype:* --

Model [11]

1. *Focus:* Software project risk assessment especially evolutionary prototype software's
2. *Proposes:* Risk Assessment Model for Software Prototyping Projects
3. *Description:* Addresses the risk assessment issue, introducing metrics and a model that can be integrated with prototyping development processes. The proposed model which uses causal analysis to find the primitive threat factors, provides a way to structure and automate the assessment of risk.
4. *Inputs:* Requirements, personal, and complexity metrics
5. *Risk Ranking:* --
6. *Decision Taking Types:* Quantitative assessment
7. *Prototype:* --

Model [12]

1. *Focus:* Risk assessment for software projects
2. *Proposes:* Software Risk Assessment Model (SRAM)
3. *Description:* The model makes use of a comprehensive questionnaire, where a set of questions is carefully chosen with three choices of answers each. The answers are arranged in increasing order of risk.
4. *Inputs:* Complexity of software, staff, targeted reliability, product requirements, method of estimation, method of monitoring, development process adopted, usability of software, and tools used for development
5. *Risk Ranking:* Assigning different weights to the

probabilities level of risk of the project according to the impact of the associated risk elements on quality, schedule and cost respectively

6. *Decision Taking Types:* Quantitative assessment
7. *Prototype:* --

Model [13]

1. *Focus:* Software project risk assessment
2. *Proposes:* Software project risk assessment model
3. *Description:* The model contains risk probability assessment model and risk impact assessment model which includes assessment of loss and comprehensive assessment of risk impact.
4. *Inputs:* Risk factor nodes
5. *Risk Ranking:* Using conditional probability distribution table (CPT) with risk semantic reduction matrix
6. *Decision Taking Types:* Hybrid assessment
7. *Prototype:* --

3.2 Methods Category

Common software project risk assessment methods are AHP, fuzzy math method, Delphi method, etc. In details, we summarized below the literature of 6 methods with the explanation. The explanation includes the model focus, proposes of the model, a brief description of the model, inputs of the proposed model, risk ranking approach of the model, decision analysis taking types of the model, and if the model implemented a proof of concept prototype. The detailed information for the contribution of each method is summarized in below.

Method [14]

1. *Focus:* Cost and quality of software projects
2. *Proposes:* Expectation-Maximization (EM) algorithm
3. *Description:* the algorithm enhances the ability in producing hidden nodes caused by variant software projects
4. *Inputs:* The probability vector of the top-level nodes
5. *Risk Ranking:* --
6. *Decision Taking Types:* Quantitative assessment
7. *Prototype:* Assessment Tool

Method [15]

1. *Focus:* Software risk assessment
2. *Proposes:* Source-based software risk assessment method
3. *Description:* The method takes into account primary facts based on workshop and secondary facts which a framework is developed.
4. *Inputs:* Secondary fact retrieval taken from organization through interviews with stakeholders, and primary fact retrieval which is analyzed from the source of the system
5. *Risk Ranking:* --
6. *Decision Taking Types:* Quantitative assessment
7. *Prototype:* --

Method [16]

1. *Focus:* General software development but its only for risk identification
2. *Proposes:* A concrete implementation method of risk identification based on the

improved Kepner-Tregoe Program

3. *Description:* Kepner-Tregoe program uses 4 analysis methods: Problem analysis (PA), Decision analysis (DA), Potential Problem analysis (PPA), and Situation analysis (SA). Each of them differs in objectives and also in application procedure respectively. Therefore, the authors' selected PPA for their risk identification as it's a kind of checklist method.
4. *Inputs:* Checking vulnerable areas of the project along the extended vulnerable areas
5. *Risk Ranking:* --
6. *Decision Taking Types:* Quantitative assessment
7. *Prototype:* --

Method [17]

1. *Focus:* Risk assessment of software projects
2. *Proposes:* Fuzzy expert system
3. *Description:* The system includes expertise to evaluate risk of software projects in all respects by using Fuzzy inference
4. *Inputs:* Corporate environment, sponsorship/ownership, relationship management, project management, scope, requirements, funding, scheduling & planning, development process, personnel & staffing, technology, and external dependencies variables
5. *Risk Ranking:* Risk matrix based on probability and severity measurements
6. *Decision Taking Types:* Quantitative assessment
7. *Prototype:* Risk assessment fuzzy expert system

Method [18]

1. *Focus:* Software project risk assessment
2. *Proposes:* Fuzzy linguistic multiple attribute decision making method
3. *Description:* The method estimates risk criteria values using linguistic terms based on triangular fuzzy number, and aggregates risk criteria values by multiple attributes decision making
4. *Inputs:* Information from experts
5. *Risk Ranking:* Risk assessment criterion is used which contains probability, loss, not controllability, and occurrence time. So the risks which have high in all criterion have high priority
6. *Decision Taking Types:* Quantitative assessment
7. *Prototype:* Case study application for historic data of completed similar projects

Method [19]

1. *Focus:* Software risk assessment
2. *Proposes:* Risk assessment method
3. *Description:* Develops software risk assessment tool using probabilistic interface model based on water fall model
4. *Inputs:* Interview-based risk assessment
5. *Risk Ranking:* Increasing order of risk by only providing 3 choices. The first choice will contribute 1 mark, 2 marks for the second choice and 3 marks for the last choice
6. *Decision Taking Types:* Quantitative assessment

7. *Prototype:* Risk Assessment Visualization Tool (RAVT) 4

4 ANALYSES OF SMSDP RISK ASSESSMENTS

4.1 Analysis Based on Assessment Parameters

In the earlier sections, we have grouped different models and methods according to 7 parameters. These parameter are focus, proposes, description, inputs, risk ranking, decision taking types, and prototype. The description parameter, which summarizes the article and decision taking type’s parameter, which will be analyzed in section 4.3, will not be analyzed in this section. Within this section we are going to analyze the aforementioned models and articles based on each parameter.

Focus: All articles are focused on risk assessment for software development projects in general. There are some of the articles specified certain scopes under project management areas or under software development methodology. Also there is an article focused on one part of risk assessment branches. On the other hand, there is an article focused on software risk assessment with additional area.

For those focused on software risk assessment with specific scope under project management are article [1] and [14]. Risk related in project time management such as errors in estimating time is focused by [1], while [14] focuses on risks related on cost and quality of software projects. On the other hand, [11] specifically focuses on risk related on evolutionary prototype software's. Moreover, article [16] focuses on one of the three branches of risk assessment, that is, risk identification. The estimation of software projects is also focused additionally in article [9].

Proposes: For articles under model's category, they all of them propose models for their risk assessment procedure. For method's category, they proposed also different methods based on different algorithms. Some of these articles used fuzzy for their proposed methods like [17] and [18], Expectation-Maximization (EM) algorithm like [14], source code based analysis like [15], and concrete implementation method of risk identification based on the improved Kepner-Tregoe Program such as in [16].

Inputs: All articles used different inputs for their risk identification, analyzation, and prioritization process. For models they used different inputs for their risk assessment model, and for methods, they created different methods based on their followed algorithm to assess risks. Table 1 details each model or methods inputs used for their risk identification.

Table 1. Risk Identification Inputs

Type	Description of Inputs
Model [1]	The risk assessment is taken in the early phases of the project and it based on project plan (e.g. Work Breakdown Structure (WBS)) and resources.
Model [9]	The models inputs are measurement errors, model errors, and assumption errors using the concept of Function point
Model [10]	The models inputs comes from risk of demand analysis, project quality, project schedule, project circumstance, technology and project personnel.
Model [11]	Metrics is used to identify the key factors of threat to provide early alerts

Type	Description of Inputs
	of risks, and they are metrics for requirements, personal, and for complexity. These metrics are followed by having characteristics of a) robustness, b) repeatability, c) simplicity in terms of the number of parameters, d) easy to calculate, and e) automatically collectable.
Model [12]	Complexity of software, staff, targeted reliability, product requirements, method of estimation, method of monitoring, development process adopted, usability of software, and tools used for development are used to identify risks
Model [13]	The inputs are risk factor nodes, that is demand risk, demand changes, technology risk, personnel communication and software performance security strategies, etc.
Method [14]	The input of the method tool is the probability vector of the top-level nodes by giving the probability of the exiting event to be 1, and the other event to be 0. The top-level nodes are new domain project, high technical complexity, lack of top management support, lack of users' participation, inexperienced project manager, inexperienced technology leader, and not enough testing.
Method [15]	Inputs come from secondary fact retrieval taken from organization through interviews with stakeholders, and primary fact retrieval which is analyzed from the source of the system
Method [16]	Checking vulnerable areas of the project along the extended vulnerable areas by using one of the four analysis methods, i.e. Potential Problem analysis (PPA) of Kepner-Tregoe Program.
Method [17]	Corporate environment, sponsorship/ownership, relationship management, project management, scope, requirements, funding, scheduling & planning, development process, personnel & staffing, technology, and external dependencies variables are method inputs
Method [18]	The inputs are Information from experts. The method estimates risk criteria values using linguistic terms based on triangular fuzzy number, and aggregates risk criteria values by multiple attributes decision making.
Method [19]	Inputs are Interview-based risk assessment where users need to answer questionnaires based on a risk taxonomy checklist.

Risk ranking: Every model or methods has declared specific ranking procedure for the risk, while some does not. For summarized information, please refer table 2 below.

Table 2. Risk Ranking Schemas

Type	Description of Risk Ranking
Model [1]	Risks are ranked based on risk rank matrix which contains risk category (1-Unknown, 2-Low, 3-Medium, 4-High, 5-Fatal), probability of occurrence, and risk impact (1-Low, 2-Medium, 3-High). The matrix produces 45 ranks for risk.
Model [9]	The estimation and ranking risks is done by using two methods, first method is based on probability by using risk exposure, and second method is software metrics of risk management which is based on Mission Critical Requirements Stability Risk Metrics (MCRSRM).
Model [10]	In weighting of risk index, the research uses a combination of two methods: subjective method (e.g. AHP), and objective method (e.g. entropy method)
Model [12]	Assigning different weights to the probabilities level of risk of the project according to the impact of the associated risk elements on quality, schedule and cost respectively
Model [13]	Using conditional probability distribution table (CPT) with risk semantic reduction matrix
Metho	Risk matrix based on probability and

Type	Description of Risk Ranking
d [17]	severity measurements
Method [18]	Risk assessment criterion is used which contains probability, loss, not controllability, and occurrence time. So the risks which have high in all criterion have high priority
Method [19]	Increasing order of risk by only providing 3 choices. The first choice will contribute 1 mark, 2 marks for the second choice and 3 marks for the last choice

Prototype: For model category, only one article has developed proof of concept prototype for their risk assessment model. Article [1] provides web application prototype using Oracle Application Express (Apex) 3.2 as web tool and Oracle Database 11g as a database tool.

On the other hand, articles [14], [17], and [19] have developed tools or expert systems for their risk assessment methods in the methods category. While [18] takes case study application for historic data of completed similar projects.

4.2 Level of Risk Awareness

There is few researches taken toward small and medium software development project (SMSDP) risk assessment, but most of them are based on a specific aspect of risks, for example, assessing risk in time management of the project [1], or assessing quality risks of the project [14].

In the previously mentioned methods and models, the majority of risk assessment for software development projects derive from software projects in general without referring whether its small, medium, or large project. As shown in figure 2, level of risk assessment awareness for large and medium software projects in large enterprises have enough assessment by using different commercial tools and framework. While small software projects does not have enough risk awareness. The more the software project size increases the more risk awareness is taken by the enterprises, and the more the software project size decreases the less risk awareness is applied.

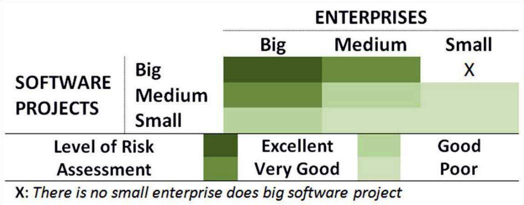

Fig. 2. Software vs. Enterprise Risk Assessment

4.3 Risk Assessment Decision Taking Types

Taking decision on a risk is based on qualitative assessment, quantitative assessment, or hybrid assessment results. Qualitative assessment means the information are in verbal form rather than in a number or quantity form as in the case of quantitative analysis. Hybrid analysis is combination of both quantitative and qualitative analysis. On the other hand, a survey done by [20] for 10 risk assessment methods, only one method is used qualitative assessment, and another one for hybrid assessment, while the remaining used quantitative assessment.

Based on the aforementioned models and methods in the literature, the decision taking types of them is illustrated in table 3. The main summaries that can be made from the table is that the most common type of information that the software risk assessment use is quantitative and in only two cases are used hybrid assessment.

Table 3. Decision Taking Types

Decision Taking Types	Model/Method	Total
Quantitative Assessment	[9], [10], [11], [12], [14], [15], [16], [17], [18], [13]	10
Qualitative Assessment	—	0
Hybrid Assessment	[1], [13]	2
Total		**12**

4.4 Some of the Limitations

The different models and methods mentioned above have some limitations including:

1. The parameters and inputs that each model or method takes are not all of them available in SMSDPs
2. While SMSDPs are rapid development projects and they run from cost, they do not have time to fill all the conditions that methods or models defines

5 CONCLUSION AND FUTURE DIRECTIONS

We have discussed and analyzed the existing software risk assessment in the literature for the last decades. A total of 12 articles were studied in this paper based on two categories: models and methods. With each category, we examined the articles according into 7 parameters. As we also discussed these parameters in each based on their different models and methods.

On the other hand, we spotlighted the gap of SMSDP risk assessment in the research field, while we are encouraging other researchers to make their focal point in the direction of SMSDP risk assessment. By the way, solving the abovementioned problems needs different directions. Firstly, this field needs deep research to find the needs and requirement of SMSDPs. Doing brain storming researches are not only enough to fill the gap of the SMSDP needs and requirements, therefore researchers should also focus on the real SMSDP projects to know exactly what those projects requires. Secondly, a part of finding the needs and requirements of SMSDPs, researchers should find also and categories risk factors for SMSDPs locally and globally. This will help to know risk factor of different projects globally. Thirdly, finding factors and requirements of SMSDPs will make easy for other researchers to prepare methods, models, or frameworks that provide suitable approaches for risk assessment of SMSDPs.

Finally, the review taken in this paper is hopefully could give the overall benefits to all researchers in the field of risk assessment for software development projects.

6 ACKNOWLEDGMENTS

The authors would like to thank Universiti Teknologi PETRONAS (UTP) for their cooperation toward submitting and presenting this paper.

7 REFERENCES

1. Sharif, A.M., Rozan, M.Z.A.: Design and Implementation of Project Time Management Risk Assessment Tool for SME Projects using Oracle Application

Express. World Academy of Science, Engineering, and Technology (WASET) 65, 1221-1226 (2010)

2. Dennis, A., Wixom, B.H., Tegarden, D.P.: System Analysis and Design with UML: An Object-Oriented Approach. John Wiley and Sons, Inc., Chichester (2005)
3. Johnson, D.L.: Risk Management and the Small Software Project. LOGOS International, Inc. (2006)
4. Gray, C.F., Larson, E.W.: Project Management: The Managerial Process. McGraw-Hill Irwin, New York (2008)
5. Boban, M., Sertic, P.Z., Strategic, H.: for Successful Software Development Risk Management. Management 8, 77-91 (2003)
6. Brandon, D.: Project Management for Modern Information Systems, vol. 417(6). Idea Group Inc., USA (2006)
7. Barkley, B.T.: Project Risk Management. McGraw-Hill, New York (2004)
8. Boehm, B.W.: Software Risk Management: Principles and Practices. IEEE Software 8(1), 32?41 (1991)
9. Gupta, D., Sadiq, M.: Software Risk Assessment and Estimation Model. In: International Conference on Computer Science and Information Technology, ICCSIT 2008, pp. 963-967 (2008)
10. Qinghua, P.: A Model of Risk Assessment of Software Project Based on Grey Theory. In: 4th International Conference on Computer Science Education, ICCSE 2009, pp. 538-541 (2009)
11. Nogueira, J., Luqi, Bhattacharya, S.: A Risk Assessment Model for Software Prototyping Projects. In: Proceedings of 11th International Workshop on Rapid System Prototyping, RSP 2000, pp. 28-33 (2000)
12. Foo, S.-W., Muruganantham, A.: Software risk assessment model. Management of Innovation and Technology. In: Proceedings of the 2000 IEEE International Conference on ICMIT 2000, vol. 2, pp. 536-544 (2000)
13. Tang, A.-g., Wang, R.-l.: Software Project Risk Assessment Model Based on Fuzzy Theory. In: International Conference on Computer and Communication Technologies in Agriculture Engineering, pp. 328-330 (2010)
14. Yong, H., Juhua, C., Huang, J., Liu, M., Xie, K.: Analyzing Software System Quality Risk Using Bayesian Belief Network. In: IEEE International Conference on Granular Computing, GRC 2007, p. 93 (2007)
15. van Deursen, A.: Source-Based Software Risk Assessment. Software Maintenance. In: Proceedings of the International Conference on ICSM 2003, pp. 385-388. IEEE Computer Society, Los Alamitos (2003)
16. Nagashima, T., Nakamura, K., Shirakawa, K., Komiya, S.: A Proposal of Risk Identification Based on the Improved Kepner-Tregoe Program and its Evaluation. International Journal of Systems Applications, Engineering and Development 4(2), 245-257 (2008)
17. Iranmanesh, S.H., Khodadadi, S.B., Taheri, S.: Risk Assessment of Software Projects Using Fuzzy Interface System, pp. 1149-1154. IEEE, Los Alamitos (2009)
18. Li, Y., Li, N.: Software Project Risk Assessment Based on Fuzzy Linguistic Multiple Attribute Decision Making. In: Proceedings of IEEE International Conference on Grey Systems and Intelligent Services, November 10-12, pp. 1163-1166 (2009)
19. Sanusi, N.M., Mustafa, N.: A visualization tool for risk assessment in software development. In: International Symposium on Information Technology, ITSim 2008, vol. 4, pp. 1-4 (2008)
20. Georgieva, K., Farooq, A., Dumke, R.R.: Analysis of the Risk Assessment Methods -A survey, pp. 76-86. Springer, Heidelberg (2009)
21. Abdullahi M. S., Shuib B., Software Risk Assessment: A Review on Small and Medium Software Projects. In: Second International Conference on Software Engineering and Computer Systems, ICSECS 2011, Springer Berlin Heidelberg, vol. 180, Part 1, pp. 214-224 (2011)