Journal of Scientific Research and Development 2 (13): 42-51, 2015
Available online at www.jsrad.org
ISSN 1115-7569

SME software risk assessment: a conceptual framework

Shuib Basri *, Abdullahi Mohamud Sharif, Ahmad Sobri Hashim

Department of Computer and Information Sciences (CIS), University Technology PETRONAS, Tronoh Perak, Malaysia

Abstract: Software development projects are by their nature complex projects. Risks or threats, on the other hand, become important factors in the successful completion of the development of software projects as many projects have failed due to improper identification and management of software risks. Assessing soft-ware risks, different approach were proposed in the literature. However these approaches lack detail in their processes and they mainly focused on large projects and on specific application areas in software projects. This paper is focused on software risk assessment in the context of SME companies. The objective of this paper is to identify the influencing factors that the analyzed approaches have. Also, this paper, proposes an effective and integrated risk assessment conceptual framework for SMEs software development projects.

Key words: Software risk assessment; Small and medium enterprises; Software development projects; Risk management

1. Introduction

In today's world, software development is becoming more complex and a strong strategy is required to successfully manage software projects. Software development projects are "a complex undertaking by two or more persons within the boundaries of time, budget, and staff resources that produces new or enhanced computer code that adds significant business value to a new or existing business process" (Wysocki, 2013). This complexity and multi-dependent characteristics make managing software development a challenging task (Singh, 2012). Moreover, the complex activities undertaken in software development can result in several vulnerabilities which can affect every portion of the project (Boban et al., 2003).

In addition, software projects face a significant amount of uncertainty that is usually manifested as possible risk materialization (Sharif, 2009). Risk is "a potential event or occurrence, which, if it occurs, has a positive or negative impact on the project's objectives" (Jordan, 2013). Assessing risk requires firstly identifying risks using different risk identification techniques, secondly analyzing the identified risks, and then finally prioritizing them in order to be able to manage and control them.

Various software risk assessment approaches are proposed in the literature that provides quantitative assessment of soft-ware project risks. These approaches include models and methods but few of these models and methods developed a visual representation of their proposed solution.

In this paper, emphasis is laid on assessing software development project risks for small and medium enterprises (SME) in the context of Malaysian software development companies. An exploration of the factors influencing on the use of risk assessment techniques and tools aims to provide an understanding of software risk assessment implementation in Malaysian software development companies from multiple perspectives. The paper also proposes an effective approach of software risk assessment that simplifies the process of identifying, analyzing, and prioritizing software project risks. The proposed effective and integrated risk assessment framework works as a guideline for a better risk assessment process.

In the coming section II, the paper summarizes a brief benchmark of existing models and methods in the software risk assessment approaches. After that, a detailed discussion of the proposed conceptual framework is presented in section IV. Finally, conclusion and future directions are presented in section V.

2. Benchmark of existing approach

To evaluate the risk involved in carrying out a project, researchers have developed different approaches including models or methods by focusing on various kinds of software projects and by using different applications (Deursen and Kuipers, 2003), (Foo and Muruganantham, 2000), (Gupta and Sadiq, 2008), (Iranmanesh et al., 2009), (Jianyi et al., 2008), (Sanusi and Mustafa, 2008), (Mustafa et al., 2010), (Luqi and Bhattacharya, 2000), (Qinghua, 2009), (Sadiq and Ahmad, 2010), (Choetkiertikul and Sunetnanta, 2010), (Tang and Wang, 2010), (Sharif and Rozan, 2010), (Choetkiertikul and Sunetnanta,

* Corresponding Author.

2012). These approaches include studies that visualized their approach into a tool and those who only proposed a model or a method. A comprehensive review of software risk assessment approaches is presented in articles written by (Sharif and Basri, 2011) and (Sharif and Basri, 2011).

In addition, two different perspectives were employed to analyze the existing approaches. Firstly, the strengths and weaknesses of the visual tool of the proposed approaches were fully analyzed. The second perspective entails assessment and further analysis of the risk assessment issues of the existing approaches based on the software risk assessment processes. Both of these analyses which presented in (Sharif and Basri, 2011) and (Sharif and Basri, 2011) are combined to generate factors that influence these approaches in order to provide effective software risk assessment that can support current practices.

The analyzed current software risk assessment approaches which provide visual representation of their approach are shown in Table 1.

Table 1: Current software risk approaches

Author(s) & Year	Area	Approach based on	Application Used
Choetkiertikul & Sunetnanta (2012) (Tang and Wang, 2010)	Offshore	CMMI Quantitative Approach	IBM Jazz
Mustafa et al. (2010) (Luqi and Bhattacharya, 2000)	Generic	Probabilistic Inference Model	Unknown
Sharif & Rozan (2010) (Singh, 2012)	Project Time	None	Oracle Appex
Iranmanesh et al. (2009) (Jianyi et al., 2008)	Generic	Fuzzy logic	MATLAB
Jianyi et al. (2008) (Sanusi and Mustafa, 2008)	Generic	Analytic Hierarchy Process (AHP) and Support Vector Regression (SVR)	Delphi 7.0 and LibSVM 2.85

In the analysis of strength, weakness, and issues of current risk assessment approaches, detailed research gap analysis of current software risk assessment approaches in the literature was provided. Some weaknesses were found to be the most recurring such as the risk prioritizing process, which was either unavailable or insufficient, and risk identification techniques. Also observed is the lack of data on project users participating in the process of risk assessment such as project manager, project team, and stakeholders. In addition, there is inadequate risk identification and prioritization processes in the analyzed approaches.

In the process of identifying, analyzing, and prioritizing software risks, these models and methods lack detail in their processes. This includes the lack of all user (manager, team, and stakeholder) involvement in the project, the lack of grouping risk into their related dimensions, and inadequate risk identification and prioritization process. In addition, these approaches tend to focus on specific application areas in soft-ware projects like e-commerce and virtual projects. As well, these approaches lack the use of different risk identification techniques, and lack of analyzing risk based on the different characteristics of risk. These characteristics include analyzing the correlation, history, time-frame, and portfolio of the risk in the project and in the organization's projects overall.

Moreover, the proposed approaches in the literature are more useful to large enterprises that undertake large software projects. The existing software risk assessment models and methods are not readily useful for small and medium enterprises projects despite the fact that the small and medium projects typically carry the same or more risk as do large projects (Gray and Larson, 2008). According to (Habra et al., 2008), one of the main characteristics of small companies is that they do not carry out risk management.

Furthermore, the analyzed approaches are inclined to the risk analysis of specific software application areas like offshore projects. This constrains the applicability of the approach to general software projects. Also, these approaches do not take into consideration the diverse risk identification techniques, analyzing risks in different impact levels including team, technical performance, cost, and schedule impacts, and risk analysis based on the different components of risk. These components comprise analyzing the correlation, history, time-frame, and portfolio of the risk in the project and of the organization's projects in general.

In line with the different problems and weaknesses inherent in existing software risk assessment approaches, this review deduces that there are several disparities between the descriptions. The studies were unable to recognize similar pat-terns within these approaches. The non-correlation of between activities, artifacts, and roles/actors is attributable to the lack of detailed data on the processes. Hence, the current software risk assessment approaches offers no viable process to practitioners. This has led to demand for a software risk assessment process that is appropriate to all practitioners. Section IV entails a detailed analysis of the aforementioned influencing factors and covering all essential process elements that provide clear guidelines for the practitioners, so as to develop an improved framework of the software risk assessment processes.

3. Study methodology

In order to develop the proposed conceptual framework that suits the practitioners in the context of SME software development companies, the following Fig. 1 shows the process methods followed in order to develop the conceptual framework.

The collected software risk assessment approaches in the literature are studied and examined. The investigation focuses on specifying the strengths, weaknesses, and assessment issues that current approaches have. The consolidation of the findings of these analyzes provides the influencing factors and the gap found in the analyzed approaches. From these influencing factors, the conceptual framework is formulated and developed. In the following section, the main constructs.

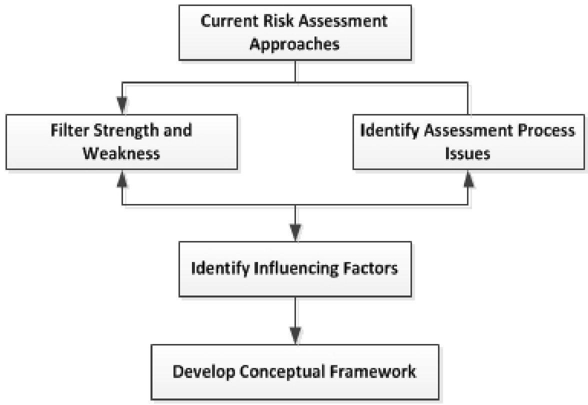

Fig. 1: Methodology

As shown in the methodology, the first phase was identifying existing approaches. The next phase was analyzing the strengths and weaknesses of the approaches and also the risk assessment issues of that these models and methods have. A comprehensive of the findings of these analysis are presented in (Sharif and Basri, 2011), (Sharif and Basri, 2011), (Sharif and Basri, 2011) and (Sharif and Basri, 2011). The next phase is to identify the influencing factors that affect the risk assessment process. These factors were identified from the weaknesses and issues that the analyzed approaches had. From these factors the relationship that each factor has to the software risk assessment process been identified which yielded the effective and integrated software risk assessment framework that is presented and detailed in next section.

4. Proposed conceptual framework

Consistent with the prior research aforementioned and in order to answer the objective of getting effective and integrated software risk assessment model, the paper integrated the influencing factors of the analyzed models, methods, and their visual tools to come out research constructs that take important role in the effectiveness of software risk assessment process. Fig. 2 shows the proposed theoretical research framework that combines the main research constructs. Next a detailed of these constructs and their importance in the process of getting effective and integrated risk assessment process are presented.

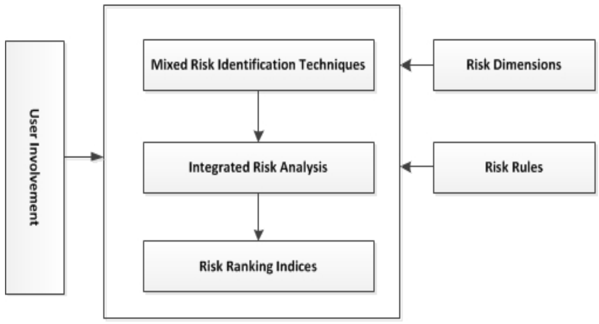

Fig. 2: Research Conceptual Framework

4.1. Mixed risk identification techniques

Risk identification is the first step in risk assessment processes. It demands the application of a diverse range of techniques which allow project managers, project team, and stakeholders to point out all likely threats that can affect the timely execution of the project within budget and of the required quality. This requires the application of variable risk identification techniques that have been reported in the literature in which their application is entirely different in relation to practitioners and researchers. For the practitioners, using variable techniques to identify risks entails time and experience, which the focus group of this research – Small and Medium Enterprises (SME), are deficient in. In contrast, the majority of researchers have developed models, methods and their visualized tools systematically or semi-systematically to identify risks in the project by integrating their models or methods using only one technique which identifies risks from one perspective.

Authors (Sharif and Rozan, 2010) and (Choetkiertikul and Sunetnanta, 2010) used project documentation data to methodically identify project risks given project documentation data is one of the major techniques for identifying project risks. Earlier on, authors (Moynihan, 1997) pointed out that using a registry in risk management, which contains prior project risks, is required for the identification of software risk. According to (Westfall, 2001), there are several risk identification techniques, which include interviewing, reporting, decomposition, assumption analysis, and critical path analysis, utilization of risk taxonomies, and using project documentation and plan data.

Interviewing/brainstorming is a technique in which undefined questions are proposed with project managers, project team, and stakeholders to identify possible risk prone areas. Voluntary reporting is another risk identification technique, where members of the project are encouraged to make suggestions, and are remunerated should they identify risk and draw the management's attention to it. Risks can also be identified by means of reporting mechanism techniques such as status reports or project reviews. Furthermore, another technique is risk identification in products of the project which are decomposed during the requirement and design phases. As author (Ould, 1990) states, “The most important thing about planning is writing down what you don’t know, because what you don’t know is what you must find out”. Thus, every TBD (“To Be Done/Determined”) is a potential risk. Hence, decomposition in the form of work breakdown structures during project planning can also aid in identifying areas of uncertainty that may be required to be reported as risks.

For another risk identification technique, Assumption Analysis, process and product assumptions must be analyzed wherein if these assumptions are verified to be false, there could be major problems. Moreover, Critical Path Analysis is used to analyze the project plan for any likelihood of schedule breakdown given that it directly affects the capability to achieve the schedule datelines. Finally, risk taxonomies are also used to identify risks by merging lists of problems that have occurred on other projects, which can be used as checklists to help make certain all possible risks have been determined. An example of risk taxonomy can be found in the Software Engineering Institute’s Taxonomy - Based Risk Identification reports (Carr et al., 1993), (Kendall et al., 2007).

Furthermore, questionnaire/checklists and Delphi techniques are also risk identification techniques that can play a significant role in project risks identification. Questionnaires are prepared with a list of existing risk factors in the field which have been reported in several studies as top or high risks in the project. The impact of these risks on current projects is analyzed by allowing the participation of all project users (project managers, project team, and project stakeholders). The prepared list of factors will aid the identification of top risks that may affect the success of a project. The contrasting feature between questionnaire and interviewing/brainstorming technique is the list of open-ended questions proposed in interviewing/brainstorming to help brainstorm project users to identify any risks that may occur in the project, unlike the questionnaire which contains directed and specific questions. On the contrary, risk identification checklist technique is closer to questionnaire risk identification technique. The questionnaire uses a Likert-scale to measure the importance level of these risk factors. Various researchers have adapted the questionnaire risk identification technique which can measure risk manually and analytically.

Delphi risk identification technique is regarded as a more versatile technique for identifying risks in contrast to other techniques but performing this technique systematically has yet to be initiated. Authors (Schmidthl et al., 2001), (Addison, 2003), (Pare, 2008) are some of the few researchers who have adapted this technique in their research. In a study by (Schmidthl et al., 2001), the process of identifying top risk factors using Delphi technique comprised three different phases. Some of these phases are also found in other techniques but the distinctiveness of this technique is the usage of panels which combines different experts to suggest risks, group them, and rank top risks by combining the expert’s dissimilar consensus and then evaluating their level of consensus.

Based on the different risk identification techniques earlier reviewed, several researchers’ evaluated different models and methods developed to identify, analyze, and prioritize risks in projects. A view of these models or methods has further implemented a visual tool for assessing risks systematically. These models and methods adapted different techniques for identifying risk as shown in Table 2.

As shown in the table, researchers adopted dissimilar tech-niques systematically or semi-systematic in their risk identification process, with no commonly shared technique(s). These researchers also assessed the applicability of these different techniques in risk identification in Malaysian SME software development projects. The combination of different identification techniques will provide more robustness compared to adapting only one technique for the risk identification process.

Table 2: Risk identification techniques adapted by the reviewed models and methods

Using Risk Identification Techniques	(Tang and Wang, 2010)	(Luqi and Bhattacharya, 2000)	(Singh, 2012)	(Sanusi and Mustafa, 2008)
Risk Taxonomies	✓	✓		✓
Delphi Study				✓
Interviewing/ Brainstorming		✓	✓	
Questionnaire		✓		✓
Decomposition	✓		✓	
Critical Path Analysis			✓	

4.2. User involvement

Authors (Al-Mudimigh et al., 2010) regarded the identification of software project risks as dynamic process that cannot be executed by either an individual such as a project manager or a specific department. Therefore, combining all project users including project managers, project teams, and stakeholders in the process of risk identification is vital to deciphering the project risks that may affect the project. On the contrary, some of the developed models or methods consider only the participation of the project manager for the identification of risk while disregarding other users. For example, authors (Mustafa et al., 2010) and (Iranmanesh et al., 2009) developed a model that considers only project managers for the risk identification process while other re-searchers considered alternative project users in their model or method.

In addition, user participation is not restricted to the risk identification but also involved in the process of analyzing and prioritizing the risks in the project. In the literature, some models only allow the involvement of project users in the input process of the risks, while analyzing and prioritizing the risks are computed systemically or semi-systematically by the developed model or method. For example, as shown in Table 3 user involvement in the model and the tool developed by (Choetkiertikul and Sunetnanta, 2010), is only restricted to specific area of risk identification, while best practices based rules are used to analyze and prioritize the risks.

Table 3: Project Users Involvement

Project Users	(Tang and Wang, 2010)	(Luqi and Bhattacharya, 2000)	(Singh, 2012)	(Jianyi et al., 2008)	(Sanusi and Mustafa, 2008)
Project Team	✓		✓		
Project Manager	✓	✓	✓	✓	✓
Stakeholders			✓		

As shown in Table III, although the views of researchers are consistent on involving project managers in the risk identification process as project teams or stakeholders, not all researchers have considered their significance. Therefore, this research postulates that involving all sectors of project users in the processes of identifying, analyzing, and prioritizing the risk offers a more comprehensive and effective approach to managing project risks.

4.3. Risk rules

Following the identification of project risks using different identification techniques, the next step entails analyzing the identified risks to further determine their severity level, and ranking them afterward based on their impact on the project. This demands the adherence to procedures that analyses the identified risks based on certain rules from different perspectives in order to evaluate their severity level toward assessing the project. In the literature, there are no conventional analysis procedures or processes and every researcher adapted their own analysis to carry out the risk analysis process. However, a majority of the researchers who executed their model or method in a visualized tool used rules for risk analysis. The risk rules differ for each tool. However, the applicability of the rules in project risk analysis is a common feature among the tools. It is assumed that using rules will aid project users in analyzing the project risks without difficulty.

Authors (Sharif and Rozan, 2010) combined risk conditions with risk scenarios to generate risk rules for project risk analysis. Risk conditions are simulated conditions that can activate a risk when certain conditions are met. Whenever a condition arises, a specific simulated scenario is proffered which defines basic features of the potential risk that may occur if that condition becomes real. In addition, authors (Iranmanesh et al., 2009) developed Fuzzy logic based IF-THEN risk rules using MAT-LAB. These fuzzy logic rules are treated as condition cases where certain actions are carried out in actual conditions in order to identify and analyses the risks in the project. Alternatively, authors (Choetkiertikul and Sunetnanta, 2010) used Capacity Maturity Model Integration (CMMI) best practices to determine the quality of the project as an inverse option to analyzing project risks. All these different risk analysis approaches are evidently insufficient when it comes to real life scenarios given that every researcher makes assertion to the uniqueness and versatility of their separately proposed risk analysis technique.

Therefore, the paper has investigated the significance of each risk rule based on the Malaysian SME software development context. Additionally, these risk rules are used in the analyzing stage of the identified risk of the project. Also, as will be explained in next section, analyzing a risk requires adhering to the characteristics of the risk which determine the level of rules to be used in order to examine all characteristics of the risk. This will lead to a more comprehensive risk analysis process that analyses the risk from contrasting angles. Therefore, the researchers examined the importance of risk rules in the risk analysis process.

4.4. Integrated risk analysis

Risk analysis based on one risk rule or an integrated risk rule requires defining the parameters in which these rules are going to process the identified risk. According to (Hubbard, 2008), good risk management demands a risk analysis process that is scientifically sound and that is supported by quantitative techniques. In addition, the aim of risk analysis is to provide an insight into the risk profile of a project and to use these insights to drive the risk response process (The Project Management Institute, 2008). Therefore, in order to obtain a scientifically sound analysis, understanding the characteristics of the risk are inevitable. As shown in Fig. 2, a list of different characteristics in the literature is presented which by doing any risk analysis should be addressed. A brief definition of the characteristics that are used for the feeding input or the parameters of analyzing the risks are explained below.

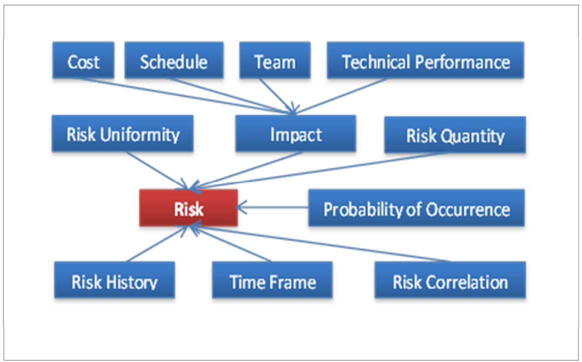

Fig. 3: Risk Characteristics

Probability of Occurrence and Risk Impact: These two parameters are the most prevalently used in the literature to compute the risk exposure which is calculated by multiplying them (Boehm, 1991). The probability of occurrence is the possibility of a threat negatively affecting the project while the impact of software risks signifies the degree or level of influence that a risk may have on the success or completion of a project. There are negative impacts and positive impacts to the project. The negative impacts are described as threats (risks) while the positive impacts are referred to as opportunities (Office of Government Commerce, 2009). This research is focused on the negative impacts (risks). In related literature, risk impact types have been defined differently. For example, the Risk Management Guide for DoD Acquisition (US Department of Defense, 2003)), stipulate five levels to evaluate the probability of occur-rence of software risks i.e., ”Remote”, ”Unlikely”,”Likely”, ”Highly likely” and “Near certainty”.

The DoD method comprises different types of impact of software risks with comprehensible assessment of criteria, and explanation for each probability scale of occurrence and impact. The impact contained in the DoD Guide includes tech-nical performance, cost, schedule and team. The assessment of the probability of occurrence and impact of software risks based on the DoD Risk Management Guide is shown in Table 4.

Table 4: DoD modified risk assessment method

Level	Probability of Occurrence	Impact			
		Technical Performance	*Cost*	*Schedule*	*Team*
1	Unlikely	Acceptable with some reduction in margins	<5%	Additional resources required. Able to meet need dates	Some
2	Likely	Acceptable with significant reduction in margins	5-7%	Minor slip in key milestone. Not able to meet need dates	Moderate
3	High Likely	Acceptable: No remaining margins	7-10%	Major slip in key milestone or critical path impacted	Major
4	Near Certainty	Unacceptable	>10%	Cannot achieve key team or major program milestone	High

In contrast, the Software Risk Evaluation (SRE) method describes the rating of the probability of occurrence on a scale of one to three, while the impact was defined by four components: cost, schedule, support and technical performance (Williams et al., 1999). Apparently, the SRE impact components did not take account for the “team” component, which has become a significant factor in modern software projects (Jiang et al., 2000). Also, authors (Karolak, 1995) proposed the Software Engineering Risk Management (SERIM) which utilizes three of the risk elements: cost, schedule and technical performance to identify any correlation between software risks. However, similar to the SRE Method, SERIM also excluded the “team” issue. A comparative analysis of the different impact for the aforementioned software risk assessment methods is shown in Table 5.

Risk Uniformity: Involves evaluating the impact of a sin-gle risk on several activities. It is applicable in activities being affected or may be affected by similar risk are categorized into a specific activity group in order to analyze the effect these activities have on similar mutual risks. According to (Creemers et al., 2010), low risk uniformity results in an average of 60 activity groups in a project network. The average number of activities in an activity group equals 2.

Table 5: Various impacts of the risk assessment (Adapted from (US Department of Defense, 2003))

Impact	SRE	SERIM	DoD Guide
Cost	✓	✓	✓
Schedule	✓	✓	✓
Technical performance	✓	✓	✓
Team			✓
Support	✓		

On the other hand, a high risk uniformity setting corresponds to an average of 20 activity groups in a project network whereas the average number of activities in an activity group equals 6. Categorizing activities based on similar risks, and analyzing them plays a significant part in identifying and analyzing the most risk prone activities in the project rather than emphasizing only top risks of the project. This generates a register of top influencing risks in the project in addition to top affected activities in the project.

Risk Quantity: During the risk identification process, several dissimilar risks are identified using various risk identification techniques. Therefore, risk quantity indicates the number of risks that are identified in the process of risk identification. A project which has a high number of risks deserves more interest than a project with relatively fewer identified risks. According to (Creemers et al., 2010), a low risk quantity setting points to a project where activities are affected by 25 risks; while the incidence of 50 risks in a project suggests risk quantity is high. These settings are reliant on management decisions given that there is no consensus on the number of risks to be considered high or low in a project, rather it differs based on company or country.

Risk Time Frame: Risks inherent in one phase of a project may have lower occurrence and impact in other phases of the project. High impact risks in the planning phase may have differing impact on other phases of the project. Therefore, risk time frame has been defined as "when the risk needs to be addressed (i.e., risk associated with activities in the near future would have a higher priority than similar risks in later activities)" (Westfall, 2001).

Risk Correlation: The interrelationship between identified project risks requires an assessment to determine if the impact of this variable will complicate risk conditions and thus increase losses. Therefore, risk correlation studies confirm if the varying risks in any given activity group are correlatable (Creemers et al., 2010).

Risk History: The risks of a project may arise from the organizational environment or caused by external factors. These risks may occur in different projects concurrently or transferred over from prior projects. As risk correlation in-volves the evaluation of the interrelationship of current project risks, risk history entails correlating previous risks or risks of other existing projects to the current project risks.

Based on the above risk characteristics, a summary of the possibility of each characteristic are outlined in Table 6.

Table 6: Risk Characteristics Matrix (Adapted from (Williams et al., 1999))

Risk Uniformity	High								Low							
Risk Quantity	H				L				H				L			
Risk Probability	H		L		H		L		H		L		H		L	
Risk Impact	H	L	H	L	H	L	H	L	H	L	H	L	H	L	H	L

Each risk impact, whether high or low, is derived from the modified DoD risk assessment method as shown in Table 7.

Table 7: Project risk impact matrix

Risk Impact	High		Low	
Impact Level	4	3	2	1
Technical Performance Impact	Unacceptable	Acceptable: No remaining margins	Acceptable with significant reduction in margins	Acceptable with some reduction in margins
Cost Impact	>10%	7-10%	5-7%	<5%
Schedule Impact	Cannot achieve key team or major program milestone	Major slip in key milestone or critical path impacted	Minor slip in key milestone. Not able to meet need dates	Additional resources required. Able to meet need dates
Team	High	Major	Moderate	Some

Based on these impact levels, explained risk characteristics or parameters that are mentioned in the literature are analyzed in this research as to their applicability in the context of Malaysian SME software development companies.

4.5. Risk ranking indices

Prioritizing project risks is the concluding step in the risk assessment process. However, there are no conventional techniques or procedures for prioritizing project risks during project risk assessment. Various researchers developed their own ranking procedure that was most suited to their proposed model or method. In addition to ranking procedures, basic ranking processes are based on two different approaches: activity-based ranking and risk-driven ranking approaches. According to (Creemers et al., 2010), risk-driven approach evaluates the impact of root risks on the uncertainty of the activities and on the project completion time. On the contrary, an activity-based approach assesses only the uncertainty of the activities without analyzing the root risks that cause this uncertainty.

Authors (Creemers et al., 2010) created two novel ranking indices based on Monte Carlo simulation for construction projects. One of these ranking indices is based on activity-based ranking indices while the other is based on risk-driven approach. Modifying these ranking indices will help software projects exploit project risks ranking based on activities and risks. In this research, applicability of both activities in the context of Malaysian SME software development was assessed by analyzing the significance of each one in the ranking process or the importance of combining both ranking approaches in the risk ranking process.

However, the proposed risk assessment models or methods for software projects reported in literature only apply risk-driven approach in their risk ranking. Nonetheless, using activity-based approach to rank project activities with severe effects from different risks is also important in project risk analysis. Therefore, this research studies the importance level of both ranking approaches in the process of risk analysis.

4.5. Risk dimensions

Risks arise from different sources, thus it is important to analyze them based on their source. There are various approaches to categorizing risks and researchers in the literature used different dimensions to group risks in software projects. Authors (Wallace et al., 2004) investigated all risk dimensions in the literature and examined them based on relevant references and established 6 dimensions which most researchers apply in their risk grouping system. These risk dimensions include user, team, requirement, project complex, planning and control and organizational environment.

In this research, these risk groups are assessed in line with their applicability in the Malaysian SME software development companies by analyzing the significance of each risk group in the process of grouping identified risks in software projects. In addition, grouping risks also plays a vital part in the analysis process of project risks. Therefore, the paper examines their importance level in the project risk analysis process. Moreover, the importance of these dimensions in using mixed risk identification techniques is also examined and analyzed.

5. Conclusion

In this paper, the influencing factors that take important roles in the processes of having integrated and effective risk assessment approach in the context of SME software enterprises are discussed in detail. Also the proposed conceptual framework and it's constructs are presented and explained. Finally, as our ongoing research, currently the study focuses on validating the proposed conceptual framework from practitioners in Malaysian SME companies.

References

A. Jordan, Risk Management for Project Driven Organizations: A Strategic Guide to Portfolio, Program and PMO Success. J. Ross Publishing, Incorporated, 2013.

A. M. Sharif and M. Z. A. Rozan, "Design and Implementation of Project Time Management Risk Assessment Tool for SME Projects using Oracle Application Express," World Academy of Science, Engi-neering, and Technology (WASET), vol. 41, pp. 126–131, 2010.

A. M. Sharif and S. Basri, "A Study on Risk Assessment for Small and Medium Software Development Projects," International Journal of New Computer Architectures and their Applications (IJNCAA), pp. 325–335, 2011.

A. M. Sharif and S. Basri, "Assessing issues for the software risk assessment tools that are based on models or methods," 2014.

A. M. Sharif and S. Basri,, "Software Risk Assessment: A Review on Small and Medium Software Projects," The Second International Conference on Software Enggineering and Computer Systems (ICSECS 2011), pp. 214–224, 2011.

A. M. Sharif, S. Basri, and H. O. Ali, "Strength and Weakness of Software Risk Assessment Tools," International Journal of Software Engineering & Its Applications, vol. 8, no. 3, pp. 389–398, 2014.

A. S. Al-Mudimigh, B. Shahzad, and Z. Ullah, "Identification of Methodology for Analysis of the Risk Factors in Software Development Environment," Global Journal of Computer Science and Technology, vol. 10, no. 1, pp. 53–62, 2010.

A. Trendowicz, Software Cost Estimation, Benchmarking, and Risk Assessment, W. R. C. II, R. E. Achatz, and H. Krcmar, Eds. Springer Berlin Heidelberg, 2013.

A. van Deursen and T. Kuipers, "Source-Based Software Risk As-sessment," in Proceedings of International Conference on Software Maintenance. ICSM2003. Los Alamitos, CA, USA: IEEE Computer Society, 2003, pp. 385–388.

A.-g. Tang and R.-l. Wang, "Software Project Risk Assessment Model Based on Fuzzy Theory," International Conference on Computer and Communication Technologies in Agriculture Engineering, pp. 328–330, 2010.

B. Singh, K. D. Sharma, and S. Chandra, "A New Model for Software Risk Management," Int.J.Computer Technology & Applications, vol. 3, no. 3, pp. 953–956, 2012.

B. W. Boehm, "Software Risk Management: Principles and Practices," IEEE Software, vol. 8, no. 1, pp. 32–41, 1991.

C. F. Gray and E. W. Larson, Project Management: The Managerial Process, 4th ed. USA: McGraw-Hill Irwin, 2008.

D. Gupta and M. Sadiq, "Software Risk Assessment and Estimation Model," in International Conference on Computer Science and Information Technology, 2008. ICCSIT '08., 2008, pp. 963–967.

D. Hubbard, The failure of risk management: Why it's broken and how to fix it. New York: Wiley, 2008.

D. W. Karolak, Software Engineering Risk Management: A Just-in-Time Approach. San Francisco, USA: Wiley-IEEE Computer Society Press, 1995.

G. Jianyi, Z. Li, L. Xusheng, H. Yuejuan, and Y. Zhengtao, “Implement-ing A Quantitative-Based Methodology for Project Risk Assessment DSS,” the 27th Chinese Control Conference, pp. 730–734, 2008.

G. Pare, C. Sicotte, M. Jaana, and D. Girouard, “Prioritizing Clinical In-formation System Project Risk Factors: A Delphi Study,” in Proceeding of the 41 st Hawaii Conference on System Sciences, 2008, pp. 1–10.

J. C. Nogueira, Luqi, and S. Bhattacharya, “A risk assessment model for software prototyping projects,” in 11th International Workshop on Rapid System Prototyping, 2000. RSP 2000., 2000, pp. 28–33.

J. Jiang, G. Klein, and T. Means, “Project risk impact on software development team performance,” Project Management Journal, vol. 31, no. 4, pp. 19–26, 2000.

L. Wallace, M. Keil, and A. Rai, “How Software Project Risk Affects Project Performance: An Investigation of the Dimensions of Risk and an Exploratory Model,” Decision Sciences 35, vol. 2, pp. 289–321, 2004.

L. Westfall, “Software Risk Management,” in Annual Quality Congress Proceedings-American Society for Quality Control. ASQ, 2001, pp. 32– 39.

M. Boban, Z. Pozgaj, and H. Seric, “Strategic for Successful Software Development Risk Management,” Management, vol. 8, pp. 77–91, 2003A. M. Sharif, “Project Time Management Risk Assessment Tool for Software Development Project,” Master’s thesis, University Technology Malaysia (UTM), 2009.

M. Choetkiertikul and T. Sunetnanta, “A Risk Assessment Model for Offshoring Using CMMI Quantitative Approach,” 2010 fifth Interna-tional Conference on Software Engineering Advances, pp. 331–336, 2010.

M. Choetkiertikul and T. Sunetnanta, “A Risk Assessment Tool Using a CMMI Quantitative Approach,” IACSIT International Journal of Engineering and Technolog, vol. 4, no. 4, pp. 352–353, 2012.

M. J. Carr, S. L. Konda, I. Monarch, F. C. Ulrich, and C. F. Walker, “Taxonomy-Based Risk Identification,” Software Engineering Institute. Technical Report: CMU/SEI-93-TR-6, ESC-TR-93-183, Tech. Rep., 1993.

M. Ould, Strategies For Software Engineering: The Management of Risk and Quality. Chichester, England: John Wiley & Sons, 1990.

M. Sadiq and M. W. Ahmad, “Software Risk Assessment and Evaluation Process (SRAEP) using Model Based Approach,” 2010 International Conference on Networking and Information Technology, pp. 171–177, 2010.

N. Habra, S. Alexandre, J.-M. Desharnais, C. Y. Laporte, and A. Re-nault, “Initiating software process improvement in very small en-terprises: Experience with a light assessment tool,” Information and software technology, vol. 50, no. 7, pp. 763–771, 2008.

N. M. Sanusi and N. Mustafa, “A visualization tool for risk assessment in software development,” in International Symposium on Information Technology, 2008. ITSim 2008., vol. 4, 2008, pp. 1–4.

N. Mustafa, N. M. Sanusi, and I. E. A.Jalil, “Software Risk Assess-ment Visualization Tool Design Using Probabilistic Inference Models,” International Symposium in Information Technology (ITSim), vol. 1, pp. 1–3, 2010.

Office of Government Commerce, Managing Successful Projects with PRINCE2. The Stationery Office, 2009.

P. Qinghua, “A model of risk assessment of software project based on grey theory,” in 4th International Conference on Computer Science Education, ICCSE ’09, 2009, pp. 538–541.

R. C. Williams, G. J. Pandelios, and S. G. Behrens, SRE Method Description & SRE Team Members Notebook, 2nd ed. Software Engineering Institute, Carnegie Mellon University, 1999.

R. P. Kendall, D. E. Post, J. C. Carver, D. B. Henderson, and D. A. Fisher, A Proposed Taxonomy for Software Development Risks for High-Performance Computing (HPC) Scientific/Engineering Applications. Software Engineering Institute, 2007.

R. Schmidth, K. Lyytinen, M. Keil, and P. Cule, “Identifying Software Project Risks: An International Delphi Study,” Journal of Management of Information Systems, vol. Vol. 17, no. No. 4, pp. 5–36, 2001.

R. Wysocki, Effective Software Project Management. Wiley India Pvt. Limited, 2006.

S. Creemers, E. Demeulemeester, and S. V. de Vonder, “A new approach for quantitative risk analysis,” Annals of Operations Research, pp. 1–26, 2010.

S. H. Iranmanesh, S. B. Khodadadi, and S. Taheri, “Risk Assessment of Software Projects Using Fuzzy Interface System,” International Conference on Computers & Industrial Engineering, 2009. CIE 2009., pp. 1149–1154, 2009.

S.-W. Foo and A. Muruganantham, “Software risk assessment model,” in Proceedings of the 2000 IEEE International Conference on Manage-ment of Innovation and Technology, ICMIT, vol. 2, 2000, pp. 536–544.

T. Addison, “E-Commerce Project Development Risks: Evidence from a Delphi survey,” International Journal of Information Management, vol. 23, no. 1, pp. 25–40, 2003.

T. Moynihan, “How experienced project managers assess risk,” IEEE Software, vol. 14, no. 3, pp. 35–41, 1997.

The Project Management Institute, Guide to the project management body of knowledge (PMBOK guide). Newton Square: The Project Management Institute, 2008.

US Department of Defense, Risk management guide for DOD ac-quisition, 5th ed. Defense Acquisition University, Defense Systems Management College., 2003.

W.-M. Han and S.-J. Huang, “An empirical analysis of risk components and performance on software projects,” The Journal of Systems and Software, vol. 80, pp. 42–50, 2007.