Risk Assessment Across Life Cycle Phases for Small and Medium Software Projects

Article *in* Journal of Engineering Science and Technology · February 2020

Muhammad Bilal
Taylor's University

Abdullah Gani
University of Malaya

Nauman Bashir
COMSATS University Islamabad

Journal of Engineering Science and Technology
Vol. 15, No. 1 (2020) 572 - 588

RISK ASSESSMENT ACROSS LIFE CYCLE PHASES FOR SMALL AND MEDIUM SOFTWARE PROJECTS

MUHAMMAD BILAL[1,*], ABDULLAH GANI[2],
MISBAH LIAQAT[3], NAUMAN BASHIR[4], NADIA MALIK[5]

[1]School of Computing and IT, Taylor's University,
47500, Subang Jaya, Malaysia
[2]Faculty of Computing and Informatics, University Malaysia Sabah,
88999, Kota Kinabalu, Malaysia
[3]College of Computing and Information Technology, University of Jeddah,
23929, khulais, Saudi Arabia
[4]Department of Computer Science, COMSATS University Islamabad,
44000, Islamabad, Pakistan
[5]Department of Management Sciences, COMSATS University Islamabad,
44000, Islamabad, Pakistan
*Corresponding Author: mbilal.csit@gmail.com

Abstract

Risk is labelled as an undesirable event that is encountered by every project irrespective of industry. Most software projects fail to meet the planned targets, i.e., scope, time, cost and quality. Software projects faced a wide range of risks and all risks cannot be dealt with the same priority. Risk can be prioritized by the probability of its occurrence and its impact. Therefore, risk assessment is required to highlight and prioritize serious risks. However, a very few researches targeted risk assessment faced by small and medium software projects. This research performs a risk assessment and highlighted serious risks faced by professionals working on small and medium software projects by documenting probability and impact. The chances of success of software projects can be increased by performing a proper risk assessment. The risks are identified by exploring and reviewing the existing literature. The identified risks are grouped by life cycle phases. This research utilizes a questionnaire-based approach to record the response of 163 software professionals working in Pakistan software industry. SPSS is used for data management and for performing statistical analysis. Probability and impact of each risk are measured to highlight the potential risks. The results concluded that the severity level of the majority of risks faced by small and medium software projects in Pakistan software industry is significant and high. The success of every project matters a lot for the progress of the organizations working on a small and medium level. Therefore, this research guide professionals and organizations to consider and prioritize the risk faced while working on small and medium software projects to increase the chances of the project's success.

Keywords: Probability impact matrix, Risk assessment, Risk management, Risk prioritization, Software projects.

1. Introduction

Risk is defined as a harmful event that may occur during a project course and has adverse consequences. The topic of risk started with the beginning of projects and project management. The risk has been under discussion due to its importance and influence on projects success from decades. Risk has been encountered by every project irrespective of industry. It can be viewed from the probability of its occurrence and its impact in terms of budget loss, schedule delays, and performance issues. The measures of project success have been classified as, nature of the procedure of project management, consumer loyalty, overall industry, productivity, and so forth. By incorporating earlier studies and research discoveries of different researchers, far-reaching hypothetical structures proposed for improvement of a project risk management the chances of project success can be increased [1].

The risk may be independent (which occurrence doesn't rely on the occurrence of other risks) or dependent (which occurrence rely on the occurrence of other risks) by nature [2]. The dependent risks can arise from both inside and outside the organization. The risks that come from inside an organization and cause troubles to a project are labelled as internal risk whereas the external risks that are hard to handle come from outside the organization [3]. To overcome software projects failure software risk management has been considered an effective approach. Risk management links potential responses to the critical risks, identify causes of failure and share a common idea of the project among its stakeholders. Risk management attempts to explore, identify, assess and overcome activities and tasks risks associated with all phases of software development life cycle (SDLC) [4].

In software projects, risk can appear due to many reasons, i.e., ambiguous requirements, lack of user involvement, contract failures, people, technology and environmental issues, etc. Software projects cannot get rid of risks completely, but the likelihood of risk occurrence can be reduced along with its impact by incorporating proper risk management techniques. The risk management is an important ingredient for software projects success [5]. The software projects are more likely to fail due to the complex and unique nature. A large portion of software projects keeps running overestimated budget and schedule plans [6]. Despite the improvement in development methodologies and management strategies, the rate of software projects failure reported by several surveys remained the same 40 – 50 % since the last few decades. The chances of software project success can be improved by managing potential risks related to quadruple constraints. The importance and need for adopting software risk management processes have also been recognized by companies, i.e., Microsoft, IBM, etc. [7].

The frequently changing, misty or confounded objectives are the most common risk associated with project managers and designers. For managing software projects, firms have found that Information management along with change control would be the most appropriate to utilize [8]. With the advancements in the field of software engineering, software organizations must adopt software project management to enhance the chances of project success. Koolmanojwong and Boehm [9] have discovered that risk designs, origin, occurrence, and future effect, changes fundamentally [9]. It has been encouraged to recognize the influence of risk and the impact of self-competence on persistence for fizzling IT project [10].

Generally, the size of software projects varies, i.e., small, medium and large. Risk affected influenced all size of software projects during the software development process. Software projects faced a wide range of risks and all the risks cannot be dealt at the same level. Therefore, there appears a need for performing a risk assessment to highlight and prioritize serious risks encounter by software projects. However, a very few researches targeted risk assessment faced by small and medium software projects. Hence, more research work needed to be done for risk assessment particularly related to software risks faced by small and medium software projects during software development lifecycle. According to Bista et al. [11], effective risk assessment not only helps in the identification and analysis of critical risks but also improves the chances of software projects success. Therefore, there is a dire need to explore the gap in the risk assessment research area by considering small and medium software project [12].

The rest of the paper is organized as follow: in Section 2, a detailed overview of the related work is presented. Section 3 describes the proposed methodology. In Section 4, the results are analysed and discussed in detail. Finally, Section 5 concludes this research.

2. Related Work

Previously, researchers have grouped the software project risks in six dimensions. Many studies focused on top ten risk lists of software projects have been presented in the literature. These risks give details to understand the whole situation. Many researchers studied different risk related to culture, time dimension, research method, and application area [4, 13]. Software projects of any size and type can be influenced by risk. The first key stage is the identification of risk to perfectly assess and control it. In the literature, the factors identified and discussed by different studies cannot be standardized. This is because of the unique nature of software projects as they are different with respect to various factors like time, culture, domains [14]. A number of potential risks appeared in distributed mobile application development has been classified and considered to improve effort estimation methods. Risk classification leads to better estimation and understanding impact of risks [15].

A study proposed a three-dimensional structure for comprehension of the risks confronted by data frameworks to audit risks and address huge measurements. In addition, a review of the significant risks management systems that have been utilized to deal with the different measurements of risk. Quickly changing hierarchical structures are making new risks for which, few data frameworks methods have been created. Adaptability has ended up one of the key rules for managing risks. Furthermore, the natural risk measurement has developed in essentialness and the test is to create risk management techniques for this measurement of software risk [16].

Project risk management is important to consider and measure the process of managing risks that are separated into different categories based on origins of risks, evaluation (risk examination), the advancement of risk mitigation plans, and accommodating leftover risks in projects plan. Different procedures are given in each of these stages, in spite of the fact that it is focused on that managing risk and ought to be seen innovatively and not be secured to an arrangement of guidelines [17]. Pasha et al. [18] mentioned that in software development projects, risk management improves decision making by providing a way to prioritize and rank risks.

The process of implementing standards, tools and techniques to recognize, examine and control risk source is known as “Risk Management. A Checklist approach is the most common approach in risk identification. The Checklist is all such risks encountered by other projects. May such lists of top ten approaches have existed in literature. However, there are some issues that accompanied this approach. The first issue is choosing a risk list, there are a variety of checklists available. Secondly, checklists created might be limited in scope, along with this it was also seen that how people perceive risk varies from culture to culture and organization [19]. The third issue for risk identification is that most of the stakeholders tend to identify risk out of there scope and domain, which are related to other stakeholders [20]. The performance of software projects improved by the implementation of risk management processes. Minimizing management by using certain techniques by narrowing scope does not always produce fruitful outcomes [21]. Risk mitigation strategies are adopted for critical risks identified, and the risk assessment is repeatedly performed to reduce risks to an acceptable level [22]. The on-going adjustment of factor like scope, time and cost for mitigation and avoidance of software project risks can be done using risk metrics, trade of triangles and estimation models [23].

Software projects regularly encounter problems such as cost overruns, quality deformities, rescheduling and time during execution. Project managers of different background saw distinctive software risks to be critical. The most vital risks encountered by project managers were: the absence of top administration duty to the project, vague/misconstrued scope/goals, plan defect, absence of customer involvement, project not proposed in light of the sound business case, absence of accessible skilled resources, absence of sufficient client contribution, poor risk management. From this rundown, it has been noticed that some of the above-mentioned risks were not found in earlier studies by Smith et al. [24]. In software projects, the risks are identified by managers using a checklist. In previous studies it is clear such type of list can assist managers to identify more risks, their number is not directly linked with the manager’s decision. However, it can be considered that some risks are more harmful to the project than other risks [25].

The government projects hold a very wide influence on the nation’s progress. The public-sector projects were appeared to be troubled and the performance is not much satisfactory even the project team adopted formal project management practices. The key properties of public projects were identified in order to bring improvements. 39 public sector projects were analysed from different countries and the reviews of audits were used. On the basis of finding a number of public projects properties were suggested along with a number of recommendations to keep in consideration. This is by following proposed recommendations projects, in which, can show better performance by following project management practices related to properties of the project [26]. To make a software project successful a number of challenges need to address across each phase of SDLC. The projects in the IT industry can be categorized as short-term (less than six months) and long-term (greater than six months) projects. The probability of occurrence of risk varies for the long-term and short-term projects. However, the impact of risk on long-term projects is much higher as compared to short-term projects [27].

Based on studies by Boehm [28], software projects must adopt the proper risk management win-win solution, rework avoidance and disaster avoidance. Risk management processes are important for all kind of software projects, i.e., distributed or collocated [29]. A number of improved approaches are suggested to identify, analyse and overcome risks related to distributed software development [30]. Hijazi et al. [31] examined popular SDLC models, i.e., waterfall, spiral, incremental, etc. for monitoring risk and risk management processes. The survival of any business depends on the successful completion of projects. However, the ratio of failed software projects is very high [32]. Risk management practices and strategies have been recognized for reducing and avoiding risks in software projects before the occurrence. 40 common occurring technical and managerial risks have been identified that affects the quality of software projects in Palestine [33].

Risk assessment and the development of intelligent risk management tools are needed for software projects [34]. A new conceptual model having additional conceptual factors has been proposed for software risk management [35]. The risks in software development projects can be categorized into cost, time, quality, people and process risks by examining risk sources and performing impact analysis [36]. A stochastic model has been proposed for risk assessment to analyse factors related to the productivity of the team in distributed software projects [37]. The propagation and severity of risk though software phases have been analysed to prioritize risk for effective risk management [38]. Two different approaches for risk prioritization have been presented to assist risk identification and its impact when considering the performance aspects of software projects. Objective risks and resilience risks are among the types of risks that can influence software projects performance. The objective risks negatively influence all aspects of project performance [39].

The software projects that are managed according to risk management are appeared to be more successful than the projects, which do not include risk management. Various risks have been identified that can leads to an alarming situation for project performance [40]. The literature on project risk management reported that project performance can be improved by managing risk properly [41]. The two dimensions of risk assessment are "probability and "impact". Probability is related to uncertainty or chances of occurrence of any risk, whereas impact is related to the effect or consequences of risk if occurred in terms of budget. Both dimensions should be kept into consideration while performing a risk analysis to prioritize risk. A risk having a high probability of occurrence but no impact the risk is not considered significant. Similarly, a risk with significant impact and a low probability is also not considered worthy to investigate. Probability-Impact Matrix by Project Management Institute (PMI) is an example framework that considers both dimensions for risk assessment [42].

Software projects faced various uncertain risks, and risk management has to explore the cause and effect relationships due to which, the application of software risk management processes is not an easy task. Efforts have been made by researchers to identify and publish risk lists that may help project managers in identifying potential risks that their software project is expected to face. In spite of various software risk management methodologies, there is still a high failure rate of software projects. If the rate of risk factors increased, it will become more difficult to manage software project performance. The past researchers concluded that there is a need for further analysis of risk assessment [4, 43]. Moreover, a study reported that the demand for project managers in Pakistan software industry is only 2%, compared to other job roles [44].

3. Research Methodology

This research is based on a survey-based approach to record the response of professionals working in Pakistan software industry. A structured questionnaire is designed to assess risk across life cycle phases for small and medium software projects selected from the literature [4, 45-48]. The sample size is 163 professionals. 4-point likert scale is used to measure the probability of risks as [49]:

- 1 = Low (less than 10% of chance happening)
- 2 = Moderate (10-29% of chance happening)
- 3 = Significant (30-35% of chance happening)
- 4 = High (greater than 50% of chance happening)

Hughes and Cotterell [49] mentioned that similarly, 4-point likert scale is used to measure the impact of risks:

- 1 = Low (within 10% of budgeted expenditure.)
- 2 = Moderate (10 to 19% above budgeted expenditure)
- 3 = Significant (20 to 29% above budgeted expenditure)
- 4 = High (greater than 30% above budgeted expenditure)

Along with the response, the demographics of respondents are also recorded. To perform risk assessment, the recorded responses are processed by using SPSS by performing various tests, i.e., frequency distribution, reliability tests, descriptive statistics and, probability impact matrix to calculate the significance of risks. The research flow is illustrated in Fig. 1.

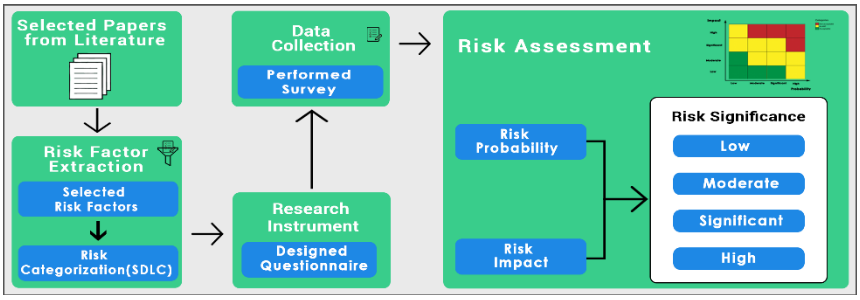

Fig. 1. Research flow.

4. Results and Discussion

The results are analysed by importing collected responses using a Google survey to SPSS. The demographics of respondents are reported in Table 1. 70.5% of respondents are male whereas 32.6% are female. 74.2% of respondents are single and 25.8% are married. The majority of respondents, 41.7% have age between 20 – 30 years, 41.1% have a salary range from RS. 40,001-60,000, 68.1% have BS qualification and 44.2% have 40 working hours per week. To test the reliability of data collected using a questionnaire, Cronbach's Alpha is calculated for all risk factors grouped according to various project life cycle phases. George and Mallery [50] reported that the 7 values of Cronbach's Alpha out of 10 are above 0.7, which indicates that the data is acceptable for further analysis and conclusions. The reliability statistics are reported in Table 2.

Table 1. Demographics of respondents.

Demographics			
		Frequency	**Percent**
Gender	Male	115	70.5
	Female	48	32.6
	Total	163	100
Marital status	Single	121	74.2
	Married	42	25.8
	Total	163	100
Age of respondent	20-25 years	47	28.8
	26-30 years	68	41.7
	31-35 years	35	21.5
	36-40 years	8	4.9
	40 and above	5	3.1
	Total	163	100
Monthly income (Rs.)	21,000-40,000	41	25.1
	40,001-60,000	67	41.1
	60,001-80,000	28	17.2
	80,001-1,00,000	14	8.6
	1,00,001 and above	13	8.0
	Total	163	100
Education level	BS	111	68.1
	MS	34	20.9
	Others	18	11.0
	Total	163	100
Number of hours worked (per week)	Less than 40 hours	55	33.7
	40 hours	72	44.2
	more than 40 hours	36	22.1
	Total	163	100

Table 2. Reliability statistics of probability and impact of risk factors.

Reliability statistics			
		Cronbach's alpha	
Project phases	***N* of Items**	**Probability**	**Impact**
Requirement and planning	17	.634	.714
Analysis and design	16	.781	.705
Coding/development	19	.704	.671
Testing	16	.641	.783
Deployment and maintenance	15	.784	.702

The risks are mapped according to calculated values probability impact values for each risk in the context of small and medium software projects. The significance of risk is represented by Eq. (1). Where P the probability of occurrence of any risk, whereas I represents the impact of risk in terms of cost. The probability and impact values for risks associated with each SDLC phase are mapped to probability impact matrix and represented in Figs. 2 to 6.

The colour gradient shows the level of significance for each risk. The green colour region shows the risk, which has low significance, the yellow colour is for risks having moderate significance; amber colour shows the risk that has significant significance, whereas the red colour shows the risk that has a high significance. The Significance values for each individual risk are reported in Table 3.

$$S\ (significance) = P\ (probability) \times I\ (impact) \tag{1}$$

Figure 2 illustrates the pictorial mapping of requirement and planning risk. The significance for 17 risks associated with this phase is calculated and the results reported that no risk appeared to have low significance, 8 risks are moderate, and 4 are significant, whereas 5 risks are reported to have a high significance. Figure 3 represents a probability impact matrix for risks associated with the analysis and design phase, 3 risks appeared moderate, 8 as significant and 5 as high.

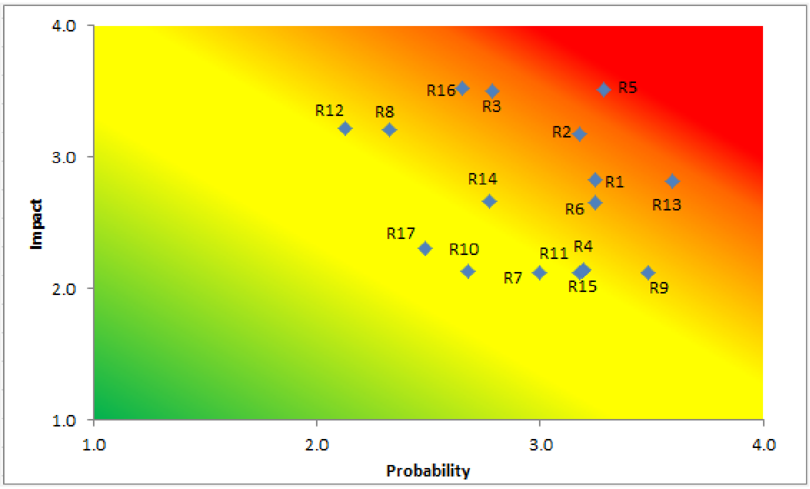

Fig. 2. Probability impact matrix of requirement and planning risks.

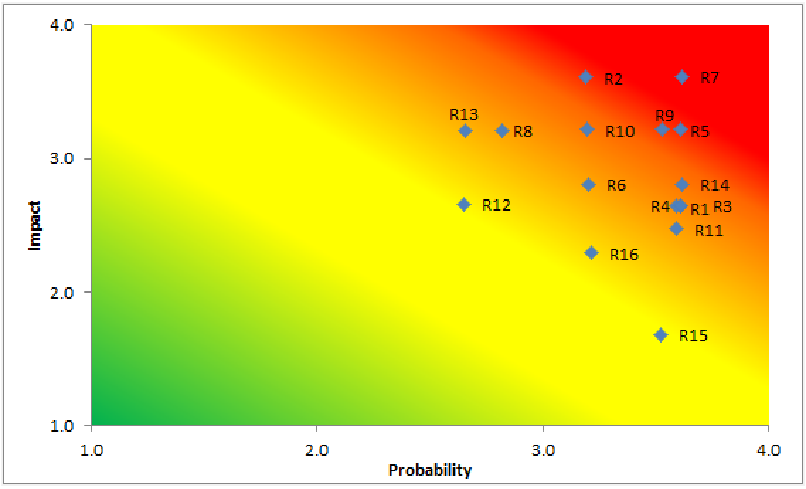

Fig. 3. Probability impact matrix of analysis and design risks.

The probability impact matric for risks related to coding/development phase is illustrated in Fig. 4. The R10 (too many syntax errors) has low significance, 3 risks are moderate, 9 are significant and 6 are high. The risks related to testing phase are mapped according to probability impact values are illustrated by pictorial mapping of each risk in Fig. 5, 5 risks are moderate, 8 risks are significant and 3 risks have a high significance. Figure 6 illustrates the risk mapping for the deployment and maintenance phase. The 3 risks ($R1$, $R7$, $R12$) are moderate, 6 risks are significant and 6 risks have a high significance.

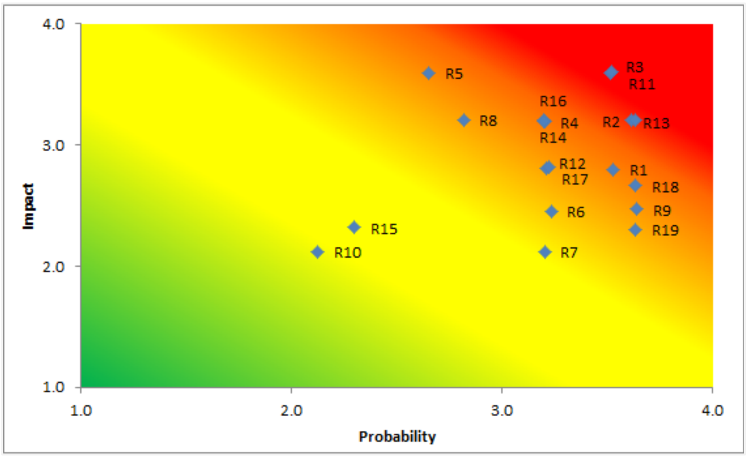

Fig. 4. Probability impact matrix of coding/development risks.

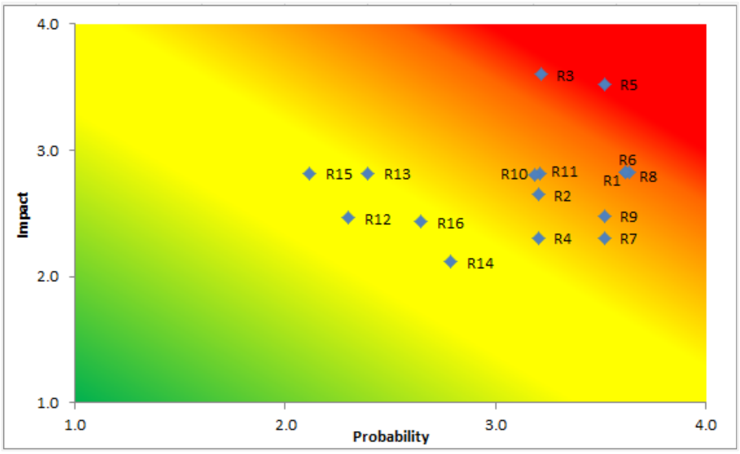

Fig. 5. Probability impact matrix of testing risks.

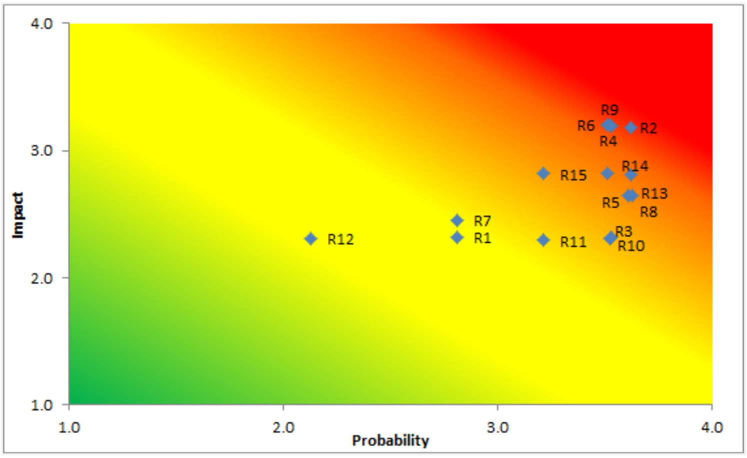

Fig. 6. Probability impact matrix of deployment and maintenance risks.

The descriptive statistics and a significance level of each risk for requirement and planning, analysis and design, coding/development, testing and deployment, and maintenance are reported in Table 3.

Table 3 risk for each phase is assigned a label (L), the risks are mapped to the probability impact matrix (Figs. 1 to 6) according to these labels. Significance for each risk is reported using Eq. (1). Moreover, Table 3 also gives means values and standard deviation (SD) for probability and impact for each risk.

Table 3. Descriptive statistics and significance of risk factors.

Phases	L	Risk factors	Probability		Impact		Significance
			Mean	SD	Mean	SD	
Requirements and planning	$R1$	Lack of user involvement	3.245	.7123	2.822	.6278	Significant
	$R2$	Unrealistic schedules and budgets	3.178	.7529	3.172	.7166	High
	$R3$	Unrealistic scope and objectives/goals	2.785	.6454	3.503	.6022	High
	$R4$	Insufficient/ inappropriate staffing	3.196	.7442	2.135	.5828	Moderate
	$R5$	Poor/inadequate planning	3.282	.6984	3.515	.5917	High
	$R6$	Change in organizational management during the software project	3.245	.7209	2.650	.8053	Significant
	$R7$	Ineffective communication software project system	3.000	.4714	2.123	.5856	Moderate
	$R8$	Absence of historical data (templates)	2.325	.8380	3.209	.7491	Significant
	$R9$	Unclear/incorrect system requirements	3.485	.6222	2.123	.5749	Significant
	$R10$	Delay in documentation	2.675	.7926	2.129	.5895	Moderate
	$R11$	Lack of IT management	3.178	.7529	2.123	.5856	Moderate
	$R12$	Artificial deadlines	2.123	.5960	3.215	.7515	Moderate
	$R13$	Inadequate training team members	3.595	.6539	2.816	.6210	High
	$R14$	Project milestones not clearly defined	2.773	.6507	2.663	.7952	Moderate
	$R15$	Lack of senior management commitment and technical leadership	3.190	.7499	2.129	.5789	Moderate
	$R16$	Ignoring the non-functional requirements	2.650	.7899	3.521	.5915	High
	$R17$	Non-verifiable requirements	2.485	.6700	2.301	.8472	Moderate
Analysis and design	$R1$	Failure to incomplete or missing detailed requirements analysis	3.607	.6425	2.644	.8065	Significant
	$R2$	Major requirements change after software project plan phase	3.190	.7499	3.607	.6520	High
	$R3$	Changing software project specifications	3.589	.6642	2.650	.8053	Significant
	$R4$	Inadequate value analysis to measure progress	3.607	.6520	2.650	.8053	Significant
	$R5$	Introduction of new technology	3.607	.6520	3.215	.7515	High
	$R6$	Designing wrong user interface	3.202	.7549	2.810	.6241	Significant
	$R7$	Insufficient procedures to ensure security, integrity, and availability of the database	3.613	.6414	3.613	.6414	High
	$R8$	Lack of integrity/consistency	2.822	.6178	3.209	.7491	Significant
	$R9$	Absence of quality architectural and design documents	3.528	.5912	3.215	.7515	High
	$R10$	Redefining the business rules	3.196	.7606	3.215	.7515	Significant

Phases	*L*	Risk factors	Probability		Impact		Significance
			Mean	SD	Mean	SD	
	*R*11	Failure to redesign and design (blueprints) software processes	3.589	.6642	2.472	.6696	Significant
	*R*12	Misalignment of a software project with local practices and processes	2.650	.8053	2.656	.8042	Moderate
	*R*13	Assumption on the number of interfaces	2.656	.7964	3.209	.7573	Significant
	*R*14	Extensive specification	3.613	.6414	2.810	.6142	High
	*R*15	Many feasible solutions available	3.521	.5915	1.681	.7261	Moderate
	*R*16	Difficulties in allocating functions to components	3.215	.7515	2.294	.8457	Moderate
Coding/development	*R*1	Inadequacy of source code comments	3.528	.5912	2.798	.6202	Significant
	*R*2	Developer software gold-plating	3.613	.6414	3.202	.7549	High
	*R*3	Platform tools are not independent	3.515	.5917	3.595	.6633	High
	*R*4	Engineering standard not defended	3.202	.7300	3.190	.7581	Significant
	*R*5	Unlicensed software	2.650	.8130	3.595	.6633	Significant
	*R*6	Programming for the future	3.233	.7334	2.454	.6592	Moderate
	*R*7	Insufficient reuse of existing technical objects	3.202	.7549	2.123	.5856	Moderate
	*R*8	Obsolescence of technology	2.816	.6210	3.209	.7573	Significant
	*R*9	Development environment	3.638	.6169	2.472	.6696	Significant
	*R*10	Too many syntax errors.	2.123	.5856	2.123	.5856	Low
	*R*11	Developing the wrong software functions and properties	3.521	.5915	3.607	.6520	High
	*R*12	Inadequate knowledge about tools and programming techniques	3.209	.7573	2.810	.6142	Significant
	*R*13	Personnel shortfalls	3.632	.6182	3.209	.7573	High
	*R*14	Developing the wrong user interface	3.196	.7606	3.202	.7549	High
	*R*15	Programmers cannot work independently	2.294	.8310	2.319	.8513	Moderate
	*R*16	Programming language does not support architectural design	3.196	.7606	3.196	.7524	Significant
	*R*17	Complex, ambiguous, inconsistent code	3.221	.7456	2.816	.6210	Significant
	*R*18	Developing components from scratch	3.632	.6182	2.663	.7952	High
	*R*19	Large amount of repetitive code	3.632	.6182	2.307	.8413	Significant
Testing	*R*1	Lack of complete automated testing tool	3.632	.6182	2.822	.6278	High
	*R*2	Test case design and Unit-level testing turns out very difficult	3.202	.7300	2.650	.8053	Significant
	*R*3	Unqualified testing team	3.215	.7432	3.601	.6624	High
	*R*4	Variation in number of test cases	3.202	.7384	2.307	.8413	Significant
	*R*5	High fault rate in newly designed components	3.515	.5917	3.521	.5915	High
	*R*6	Inadequate test cases and generate test data	3.613	.6414	2.822	.6178	Significant
	*R*7	Testing is monotonous, boring and repetitive	3.515	.5917	2.307	.8413	Significant
	*R*8	Not all faults are discovered in unit testing	3.613	.6414	2.822	.6178	Significant
	*R*9	Poor documentation of test cases	3.515	.5917	2.479	.6698	Significant
	*R*10	Poor regression testing	3.209	.7408	2.816	.6210	Significant

Phases	L	Risk factors	Probability		Impact		Significance
			Mean	SD	Mean	SD	
	*R*11	Integrate wrong version of components	3.184	.7637	2.810	.6142	Significant
	*R*12	Difficulties in localizing errors	2.301	.8472	2.466	.6600	Moderate
	*R*13	Difficulties in repairing errors	2.393	.8273	2.816	.6210	Moderate
	*R*14	Non-readable code and data design	2.785	.5957	2.117	.5708	Moderate
	*R*15	Difficulties in ordering components' integration	2.117	.5599	2.816	.6210	Moderate
	*R*16	Lack of traceability, confidentiality, correctness, and inspection of the software project planning	2.644	.8065	2.436	.6670	Moderate
Deployment and maintenance	*R*1	Failure to gain user commitment	2.810	.6142	2.319	.8513	Moderate
	*R*2	Failure to utilize a phased delivery approach	3.620	.6306	3.184	.7555	High
	*R*3	Too little attention to breaking development and implementation into manageable steps	3.521	.5915	2.307	.8413	Significant
	*R*4	Real-time performance shortfalls	3.509	.6021	3.196	.7606	High
	*R*5	Lack of adherence to programming standards	3.626	.6294	2.650	.8053	Significant
	*R*6	Lack of resources and reference facilities	3.521	.5915	3.196	.7606	High
	*R*7	Lack of top management commitment and support and involvement	2.810	.6241	2.454	.6685	Moderate
	*R*8	Shortfalls in externally furnished components, COTS	3.607	.6614	2.650	.8053	Significant
	*R*9	Acquisition and contracting process mismatches	3.528	.5807	3.190	.7581	High
	*R*10	User documentation missing or incomplete	3.528	.5807	2.313	.8353	Significant
	*R*11	Variation in configuration component	3.215	.7179	2.294	.8457	Significant
	*R*12	Connectivity issues	2.129	.5789	2.307	.8413	Moderate
	*R*13	Integration is required between many different technologies	3.620	.6403	2.810	.6142	High
	*R*14	Acquisition and contracting process mismatches	3.509	.5918	2.816	.6210	High
	*R*15	Budget not enough for maintenance activities	3.215	.7515	2.816	.6210	Significant

The summary of risk assessment is presented in Table 4. Total of 83 risks are explored and the overall summary shows that 1 risk is low, 22 are moderate, 35 are significant and 25 have a high significance. The majority of risks associated with small and medium software projects in Pakistan software industry appeared to be significant and high.

Table 4. Summary of risk assessment.

Project phases	*N* of risks	Low	Moderate	Significant	High
Requirement and planning	17	0	8	4	5
Analysis and design	16	0	3	8	5
Coding/development	19	1	3	9	6
Testing	16	0	5	8	3
Deployment and maintenance	15	0	3	6	6
Total	83	1	22	35	25

5. Conclusions

The process of software development is full of diverse risks from beginning till end. Efforts have been made by researchers to identify and publish risk lists that may help project managers in identifying potential risks that a software project is expected to face. In spite of various software risk management methodologies, there is still a high failure rate of software projects. The risk assessment for small and medium software projects is neglected. This study tried to perform the risk assessment for small and medium software projects in Pakistan software industry by grouping risks identified by previous studies across lifecycle phases, i.e., requirement and planning, analysis and design, coding/development, testing and deployment, and maintenance. A survey-based approach is adopted and a structured questionnaire is used as an instrument to record the response of 163 professionals working on small and medium software projects in Pakistan software industry. The results are analysed using SPSS and risk is highlighted by assigning the significance level according to the probability and impact values calculated. In future work, we will investigate the causes of risks along with introducing and evaluating various reduction techniques for each risk.

Nomenclatures

I	Impact of risk in terms of cost
P	Probability of risk occurrence
S	Significance of risk

Abbreviations

PMI	Project Management Institute
SDLC	Software Development Life Cycle

References

1. Aldhfayan, F.S. (2008). Analysis of the role of standardized project management on project performance. Retrieved August 05, 2019, from http://www.strategicstandards.com/files/2008/ProjectManagement08.pdf.
2. Pressman, R.S. (2005). *Software engineering: a practitioner's approach* (7th ed.). New York, United States of America: McGraw-Hill Higher Education.
3. Uzzafer, M. (2010). A risk classification scheme for software projects. *International Journal of Software Engineering and its Applications*, 7(1).
4. Arnuphaptrairong, T. (2011). Top ten lists of software project risks: Evidence from the literature survey. *Proceedings of the International MultiConference of Engineers and Computer Scientists (IMECS)*. Hong Kong, 1-6.
5. Chowdhury, A.A.M.; and Arefeen, S. (2011). Software risk management: Importance and practices. *International Journal of Computer and Information Technology (IJCIT)*, 2(1), 49-54.
6. Tavares, B.G.; da Silva, C.E.S.; and de Souza, A.D. (2017). Risk management analysis in Scrum software projects. *International Transactions in Operational Research*, 26(5), 1884-1905.

7. Avdoshin, S.M.; and Pesotskaya, E.Y. (2011). Software risk management. *Proceedings of 7th Central and Eastern European Software Engineering Conference (CEE-SECR)*. Moscow, Russia, 1-6.
8. Neves, S.M.; da Silva, C.E.S.; Salomon, V.A.P.; da Silva, A.F.; and Sotomonte, B.E.P. (2014). Risk management in software projects through knowledge management techniques: Cases in Brazilian incubated technology-based firms. *International Journal of Project Management*, 32(1), 125-138.
9. Koolmanojwong, S.; and Boehm, B. (2013). A look at software engineering risks in a team project course. *Proceedings of 26th International Conference on Software Engineering Education and Training (CSEE&T)*. San Francisco, California, United States of America, 21-30.
10. Jani, A. (2011). Escalation of commitment in troubled IT projects: Influence of project risk factors and self-efficacy on the perception of risk and the commitment to a failing project. *International Journal of Project Management*, 29(7), 934-945.
11. Bista, R.; Karki, S.; and Dongol, D. (2017). A new approach for software risk estimation. *Proceedings of 11th International Conference on Software, Knowledge, Information Management and Applications (SKIMA)*. Malabe, Sri-Lanka, 1-8.
12. Sharif, A.M.; and Basri, S. (2011). A study on risk assessment for small and medium software development projects. *International Journal of New Computer Architectures and Their Applications (IJNCAA)*, 1(2), 325-335.
13. Moran A. (2014). *Agile risk management*. Springer briefs in computer science. Cham, Switzerland: Springer International Publishing.
14. Sharif, A.M.; and Basri, S. (2014). Risk assessment factors for SME software development companies in Malaysia. *Proceedings of International Conference on Computer and Information Sciences (ICCOINS)*. Kuala Lumpur, Malaysia, 1-5.
15. Aslam, W.; Ijaz, F.; Lali, M.I.; and Mehmood, W. (2017). Risk aware and quality enriched effort estimation for mobile applications in distributed agile software development. *Journal of Information Science and Engineering*, 33(6), 1481-1500.
16. Sherer, S.A. (1995). The three dimensions of software risk: Technical, organizational, and environmental. *Proceedings of the Twenty-Eighth Annual Hawaii International Conference on System Sciences*. Wailea, Hawaii, United States of America, 369-378.
17. Perry, J.G. (1986). Risk management-an approach for project managers. *International Journal of Project Management*, 4(4), 211-216.
18. Pasha, M.; Qaiser, G.; and Pasha, U. (2018). A critical analysis of software risk management techniques in large scale systems. *IEEE Access*, 6, 12412-12424.
19. Keil, M.; Tiwana, A.; and Bush, A. (2002). Reconciling user and project manager perceptions of IT project risk: A Delphi study 1. *Information Systems Journal*, 12(2), 103-119.

20. Schmidt, R.; Lyytinen, K.; Keil, M.; and Cule, P. (2001). Identifying software project risks: An international Delphi study. *Journal of Management Information Systems*, 17(4), 5-36.
21. Ropponen, J.; and Lyytinen, K. (1997). Can software risk management improve system development: An exploratory study. *European Journal of Information Systems*, 6(1), 41-50.
22. Esche, M.; Toro, F.G.; and Thiel, F. (2017). Representation of attacker motivation in software risk assessment using attack probability trees. *Proceedings of Federated Conference on Computer Science and Information Systems (FedCSIS)*. Prague, Czech Republic, 763-771.
23. Ghane, K. (2017). Quantitative planning and risk management of Agile Software Development. *Proceedings of Technology & Engineering Management Conference (TEMSCON)*. Santa Clara, California, United States, of America, 109-112.
24. Smith, D.; Eastcroft, M.; Mahmood, N.; and Rode, H. (2006). Risk factors affecting software projects in South Africa. *South African Journal of Business Management*, 37(2), 55-65.
25. Li, L. (2013). The impact of risk checklists on project manager's risk perception and decision-making process. *Proceedings of the Southern Association for Information Systems Conference.* Savannah, Georgia, United States of America, 106-110.
26. Patanakul, P.; Kwak, Y.H.; Zwikael, O.; and Liu, M. (2016). What impacts the performance of large-scale government projects? *International Journal of Project Management*, 34(3), 452-466.
27. Bhujang, R.K.; and Suma, V. (2014). Risk impact analysis across the phases of software development. *Lecture Notes on Software Engineering*, 2(3), 282-287.
28. Boehm, B. (1989). *Software risk management*. United States of America, Washington, D.C.: IEEE Computer Society Press.
29. Prikladnicki, R.; and Yamaguti, M.H. (2004). Risk management in global software development: A position paper. *Proceedings of 26th International Conference on Software Engineering*. Edinburgh, United Kingdom, 18-20.
30. Keshlaf, A.A.; and Riddle, S. (2010). Risk management for web and distributed software development projects. *Proceedings of Fifth International Conference on Internet Monitoring and Protection (ICIMP)*. Barcelona, Spain, 22-28.
31. Hijazi, H.; Khdour, T.; and Alarabeyyat, A. (2012). A review of risk management in different software development methodologies. *International Journal of Computer Applications*, 45(7), 8-12.
32. Kanane, A. (2014). *Challenges related to the adoption of Scrum. Case study of a financial IT company*. Master Thesis. Department of Informatics, UMEA University, Umea, Sweden.
33. Elzamly, A.; and Hussin, B. (2011). Estimating quality-affecting risks in software projects. *International Management Review*, 7(2), 66-83.
34. Dhlamini, J.; Nhamu, I.; and Kaihepa, A. (2009). Intelligent risk management tools for software development. *Proceedings of the Annual Conference of the*

Southern African Computer Lecturers' Association. Eastern Cape, South Africa, 33-40.

35. Sarigiannidis, L.; and Chatzoglou, P.D. (2011). Software development project risk management: A new conceptual framework. *Journal of Software Engineering and Applications*, 4(5), 293-305.
36. Ogunsanmi, O.E. (2016). Risk classification model for design and build projects. *Covenant Journal of Research in the Built Environment*, 3(1), 54-76.
37. Bhujang, R.K.; and Suma, V. (2014). Risk impact analysis across the phases of software development. *Lecture Notes on Software Engineering*, 2(3), 282-287.
38. Bhujang, R.K.; and Dean, S.V. (2018). Propagation of risk across the phases of software development. *Proceedings of 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC) I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*. Palladam, India, 508-512.
39. Han, W.-M. (2014). Validating differential relationships between risk categories and project performance as perceived by managers. *Empirical Software Engineering*, 19(6), 1956-1966.
40. Jun, L.; Qiuzhen, W.; and Qingguo, M. (2011). The effects of project uncertainty and risk management on IS development project performance: A vendor perspective. *International Journal of Project Management*, 29(7), 923-933.
41. Junior, R.R.; and de Carvalho, M.M. (2013). Understanding the impact of project risk management on project performance: An empirical study. *Journal of Technology Management and Innovation*, 8, Special Issue ALTEC, 64-78.
42. Hillson, D.A.; and Hulett, D.T. (2004). Assessing risk probability: Alternative approaches. *Proceedings of PMI Global Congress*. Prague, Czech Republic, 1-7.
43. Hoodat, H.; and Rashidi, H. (2009). Classification and analysis of risks in software engineering. *World Academy of Science, Engineering and Technology*, 3(8), 2044-2050.
44. Bilal, M.; Malik, N.; Khalid, M.; and Lali, M.I. (2017). Exploring industrial demand trend's in Pakistan software industry using online job portal data. *University of Sindh Journal of Information and Communication Technology*, 1(1), 17-24.
45. Elzamly, A.; Hussin, B.; and Salleh, N.M. (2016). Top fifty software risk factors and the best thirty risk management techniques in software development lifecycle for successful software projects. *International Journal of Hybrid Information Technology*, 9(6), 11-32.
46. Bhujang, R.K.; and Suma, V. (2014). Risk impact analysis across the phases of software development. *Lecture Notes on Software Engineering*, 2(3), 282-287.
47. Asif, M.; Ahmed, J.; and Hannan, A. (2014). Software risk factors: A survey and software risk mitigation intelligent decision network using rule based technique. *Proceedings of the International MultiConference of Engineers and Computer Scientists (IMECS)*. Hong Kong, 1-7.

48. Hijazi, H.; Alqrainy, S.; Muaidi, H.; and Khdour, T. (2014). Identifying causality relation between software projects risk factors. *International Journal of Software Engineering and Its Applications*, 8(2), 51-58.
49. Hughes, B.; and Cotterell, M. (2009). *Software project management* (5th ed.). New York: Tata McGraw-Hill Education.
50. George, D.; and Mallery, P. (2011). *Using SPSS for windows step by step: A simple guide and reference* (11th ed.). Boston, United States of America: London: Alynn and Bacon.