Identification of Methodology for Analysis of the Risk Factors in Software Development Environment

GJCST Computing Classification
D.2.6, D.2.11 & D.2.1

Abdullah S. Al-Mudimigh
Basit Shahzad, , Zahid Ullah
College of Computer & Information Science, King Saud University, Saudi Arabia
mudimigh@ksu.edu.sa basit.shahzad@gmail.com, , zahid@ksu.edu.sa

***Abstract*- Software engineering has attracted the recent focus of academia and researchers by providing them means of effective software development. The effective risk management has also played a vital role is making the software development practices more reliable and organized. Ample consideration is being given to the software risk analysis and that has enabled the more reliable software management. With emergence of the need for managing the risks in software, it is essential that suitable methodology be identified for identifying the risks. This paper, this way, works as a reverse engineering approach for the identification appropriate methodology for identifying the risks in software risks.**

Keywords- Risk identification, Risk priority, Risk emergence, software failures

I INTRODUCTION

The management of risks in software is not as old as the software development itself is. The software industry, after the orientation of the need for the documentation for any software developed, came to know that certain risks are faced during the software development lifecycle. Coper Jhons [25] in his survey in 1996, provided a detailed information about the failure of the projects and concluded that the projects or either delayed or fail due to the poor risk management techniques in practice. The delayed or a failed project does not only mean a failure of that project specifically but it also means the revenues and the reputation of the development firm also comes under question, and the development team has to strive hard to re-gain its market position: both in terms of revenues and reputation. In the recent past, software risk management has gain the attention specially and much contribution has become from the academia to help in the identification [15], management [16] and prioritization [17] of the software risk factors.

Many research factors have been investigated and identified in recent past and the race of identification is still on and certainly with more passion. However, the time demands a smart work perhaps not the hard work. It is therefore can be considered that instead of being a part of the identification race; why not provide help to the world with a way to more easily identify the risk factors. For this purpose, in this paper, we will discuss already identified risk factors, and will see that to which strategy they match perfectly for the purpose of identification and this process may be repeated for many risk factors available till now to see the most appropriate technique for their identification. Concluding the paper, a comparison will be made to show the most effective technique and suggestions will be made to use that technique to gain the maximum benefits.

II FRAMEWORK OF METHODOLOGIES

In this section, we discuss the working range of each methodology, in order to establish the effectiveness of each methodology. All three methodologies are discussed for this purpose.

i. Questionnaire (Q): Questionnaires are used when an opinion is to be gathered from the public or a group of people from different localities. Questionnaires are generally not descriptive and just provide possible options to choose from.
ii. Direct Communication (DC): DC has different meanings in different circumstances, e.g. DC means communication between the risk manager and development team while at some other point DC would reflect the meanings of communication between the risk/project manager and the customer, and sometimes it may be from risk/project manger to the management of the organization itself.
iii. Experience & Knowledge (EK): Many risks can only be identified by recalling the successes and failures in the previous projects. The intuition can work as a magic and can only not help in identification of risk factors but also in the effective management of the identified risks by incurring least resources.

III IDENTIFICATION TECHNIQUES & METHODOLOGY

Identification of software risks is such a dynamic and comprehensive activity that it cannot be performed by either an individual or a specific department. Risk identification, infact is governed by the mesh of activities taking place during the software development life cycle in the entire organization. To be precise, the identification of risks in an ongoing project is not limited to the current activities only, but a huge amount of other continuing factors contribute in making the risk factors present and evident.

In order to propose some methodologies for the identification of risk factors a detailed literature survey was conducted. [22] Has considered that the management of risk registers is suitable for the identification of software risk, while [23] believes in categorizing the risks for the purpose of identification. But the categorization itself is not possible without the identification, so this approach somehow produces a deadlock in identification and categorization. Joe Hennessy at ISD-NASA (2004), in his report on ISD software risk identification has discussed the process of risk identification and has urged that before the start of risk identification process the risk management team must be armed with the list of risk already identified in the domain including technical, budget and management risks etc. Joe focuses on the discussion of each identified risk with the development team and decides an applicability of that risk in the project under consideration. Further the development team may check that either some generic risk factor that apply to the organization apply to the specific project or not? Thus contributing to develop a complete list of risk factors that is relevant to the project. Joe has precisely emphasized the need for communication between the manager, customer, and development team. The process of identification and handing of software risk factors proposed by Joe is consistent with NPR 8000.4: Risk management procedural requirements, GPG 7120.4: Risk management, GPG 8700.5: In-house development and maintenance of software projects and IEEE std 1540-2001: Standard for software life cycle processes-risk management. [17]

In another report on "taxonomy-based risk identification" by Software Engineering Institute: the method described in the report consists of developing the taxonomy-based questionnaires for identification of software risk. Taxonomy is a scheme that partitions the body of knowledge and defines the relationship among the pieces [18]. This report emphasis the need of questionnaires for identification of risk factors.

In anther report by Software Engineering Institute (2008), the author Ray C. Williams has argued that the biggest need in managing the risks is risk identification. Ray has referenced a situation in which 40 field tests were conducted with a broad range of software developers to identify that who have good communication skills and techniques to help in the process of risk identification with their own experience and by interviewing others. Ray also argues in the favor of using the "inter-organization-communication" to report any risk that is observed at any level. He suggests that the higher management must be open with the middle management and workers by sharing the risks and inviting them to share theirs.

In yet another paper on checklist of risk identification by Mark Li (2007), different milestones of risk identification with the identification methodology has been described. Boehm (1991) identified 10 risk factors by survey of the experienced risk managers [19]. Barki et al. (1993) identified 23 risk factors by just doing the systematic literature review [20]. Heemstra and Kusters (1996) identified 36 risk items by doing the literature survey combined with experiences [21]. Moynihan (1997) identified 21 risk factors by Interviewing with 14 application developers [22]. Ropponen and Lyytinen (2000) identified six risk items by doing a survey of 83 project managers covering nearly 1100 projects[23]. Han and Huang (2007) identified six dimensions of 27 risks by an analysis of 115 software projects [8].The literature survey proposes that the risks, whatever they are, can be identified by doing the questionnaire and enhancing the inter-organization-communication, once the initial list of relevant risk factors has been identified. Although some other factors like intuition, relevance, etc can also be used for risk identification yet they are neither essential nor universal enough to be learnt as established risk identification methodologies. Therefore we consider it essential to restrict this work to three most relevant and most referenced identification methodologies, namely: Questionnaire (Q), Direct Communication (DC) and Experience & Knowledge (EK) as they have been observed to be necessary and sufficient for identification of any risk factor also these three approaches have been in wide use for the identification of risk factors in the leading academic and commercial environments throughout the world.

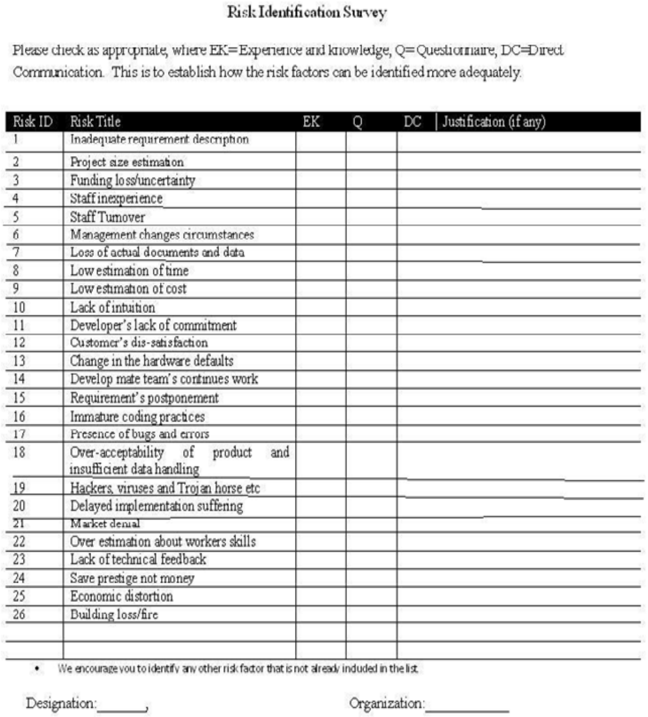
The table 1 describes different risks identified from the literature study. In the proceeding section each risk factor is considered separately, and all three methodologies (Q, DC, EK) are applied and observed that which technique can best generate /identify this risk factor. A Suitability Index (SI) is determined in this regard. The SI value of 1 shows that some risk factor e.g. x can only be generated by some specific methodology e.g. EK, which also mean that other two techniques have no contribution in the identification of that risk factor at all. The SI value 0.05 means that the identification of a risk factor e.g. y is just minutely dependent on one methodology i.e DC, and the rest 0.95 is covered by either EK or Q. likewise the SI value of 0.0 for any methodology means that this methodology has got absolutely no role to play in the identification of the risk factor under consideration. In the proceeding section we use a scale of values ranging from 0 to 1 with a multiple of 0.05. The higher the value against a specific methodology the greater the role it will possess in identification of a specific risk factor.

It is also worth mentioning that the values provided against each methodology have been derived from the survey conducted by the author and also by the author's continues experience in the domain of software risk identification and management, the author has a vast experience and contribution in this domain of knowledge [10, 11, 12, 13, 14, 15, and 16].

Having repeated this exercise for all risk factors, the total SI count is calculated for each methodology among all risk factors, and the one with highest count is declared as most suitable risk identification methodology. In order to identify the appropriate strategies for identification of risk factors, a survey was conducted on 100 individuals working in academia, management and software development across different countries in the world including Saudi Arabia, Pakistan, Sweden, Denmark and Malaysia. The survey form is shown in Table 1. For the convenience of responders, the survey has been designed in a way to require minimum effort from the respondendent.hey are just required to tick the choice(s) that they feel appropriate.

IV SURVEY DESIGN AND CONDUCTING METHODOLOGY

The risk factors from 1-26, in the table 1, have been identified by the comprehensive study of literature [12, 13, 14, 19, 20]. In most cases the source of identification was not known. However, it has been observed that most of the risk factors can be identified by using experience, direct communication between stake holders of the software or by using the questionnaire for information acquisition. The 26 most frequently used risks in software environment have been used to perform the reverse engineering to identify the methodology by which they can be identified. For this purpose, a questionnaire consisting of all 26 risk factors, with a choice of three possible answers was designed. The respondents were requested to tick the most suitable methodology that, in their opinion, can best identify that risk factor. The respondents were also free to choose both or all three methodologies to demonstrate that some specific risk factor can't be identified by using only one or two methodologies. The results of the survey are discussed in section 6 individually and in section 8 as a summary.



Risk Identification Survey

Please check as appropriate, where EK=Experience and knowledge, Q=Questionnaire, DC=Direct Communication. This is to establish how the risk factors can be identified more adequately.

Risk ID	Risk Title	EK	Q	DC	Justification (if any)
1	Inadequate requirement description				
2	Project size estimation				
3	Funding loss/uncertainty				
4	Staff inexperience				
5	Staff Turnover				
6	Management changes circumstances				
7	Loss of actual documents and data				
8	Low estimation of time				
9	Low estimation of cost				
10	Lack of intuition				
11	Developer's lack of commitment				
12	Customer's dis-satisfaction				
13	Change in the hardware defaults				
14	Develop mate team's continues work				
15	Requirement's postponement				
16	Immature coding practices				
17	Presence of bugs and errors				
18	Over-acceptability of product and insufficient data handling				
19	Hackers, viruses and Trojan horse etc				
20	Delayed implementation suffering				
21	Market denial				
22	Over estimation about workers skills				
23	Lack of technical feedback				
24	Save prestige not money				
25	Economic distortion				
26	Building loss/fire				

- We encourage you to identify any other risk factor that is not already included in the list.

Designation:______, Organization:__________

Table 1: Survey for choosing the identification methodology

V WEIGHTAGE OF THE METHODOLOGY

It has been observed that all three methodologies used for identification of risk factors in a software development environment have got certain pros and cons which are built-in with the methodology itself, e.g. the methodology 'EK' has the build-in advantage of saving time and resources and as only experienced people practice this methodology, the probability of its appropriate use is very high, and therefore proper results can be derived by using this methodology.

The 'DC' methodology is a bit slower is suggesting the response, as the project manager has to discuss the details about the project circumstances before a decision can be made, therefore the response becomes slower and some resources are to be invested to get the response. It also to be noted that the respondents of a 'DC' take this time out of their normal schedule in which they are supposed to perform some other duties as well.

The 'Q' technique suffers from the built-in disadvantage of being least explanatory and most time expensive, also the response gained from the survey is normally delayed. A survey can only be recommended if it is performed in adequate time limits required by the software development deadlines. In light of above discussion, a waitage index is suggested keeping in view the build-in pros and cons of each methodology

Method	Person hour	Schedule disturbance	waitage
EK	5	N	2.00
DC	10	Y	1.00
Q	20	Y	0.50

Table2: Weightage Index of risk identification methods

In table 2, we use a situation in which the project manager, by using his experience and knowledge suggests the solution of a problem, this single-handed effort is supposed to take 5 hours and yet not disturbing any other duty of the manager. Therefore this emerges to be the most appropriate solution and hence given the highest weightage of 2.0. If the manager determines to use services of 5 other individuals to identify the expected risk factors in software development environment, the solution will become costly and hence time consuming as well. Therefore, if the team can complete the problem in 2 hours, the organization will cost 10 person hours and the normal duties of the development team will be disturbed. As this is second most appropriate solution, the weightage index suggested for this methodology is 1.0, while weightage index for questionnaire methodology is calculated to be 0.5 as it is slowest and most resource consuming in terms of identification of any expected risk factor.

VI PROBABILISTIC IDENTIFICATION OF EACH RISK FACTOR

The approach focuses on the identification of each risk factor and it is argued that which technique, out of available three, can be most beneficial in identification of the risk factor. The approximate weightage is being given to each methodology based on its suitability, which is supported by the survey conducted in this regard and the author's intuition.

A. *Inadequate Requirement Description*

It is often known to the project manager that the customers can hardly describe the adequate amount of information about their requirements. Although the manager can identify with its experience that the requirements are incomplete and tend to change in future, yet the overall measure to which the requirements are missing and can change can only be known through by using the DC. Questionnaire, in this regard can be of a very little help, hence, the weightage of each methodology, as per survey, will be as follows:

Method	Suitability Index	WI	Suitability Weighted Index
EK	0.3	2	0.6
DC	0.6	1	0.6
Q	0.1	0.5	0.05

Table 3: Suitability Weighted Index for risk factor No.1

B. *Project Size Estimation*

Calculating the actual size of the project under consideration has been a serious question in the software cost estimation domain. It has been observed that the questionnaire can be of very little help in this regard not only because of the general irresponsiveness but also because of the natural disability of less descriptive. DC with the customer and within the development teams helps a lot in identifying the actual project size. The 'experience' plays a vital role in identification of the actual project size, and without experience other two methodologies tend to fail badly, hence, the weightage of each methodology's SI will be as follows:

Method	Suitability Index	WI	Suitability Weighted Index
EK	0.5	2	1.0
DC	0.4	1	0.4
Q	0.1	0.5	.05

Table 4: Suitability Weighted Index for risk factor No.2

C. *Project Funding Loss*

Due to the inadequate handling of the project in the beginning, or for any other reason the project may not meet the milestones and consequently delivery deadlines cannot be met. An effective project manager can, by using his experience promptly, very effectively predict about the development delays and can propose extra measures in achieving the milestones. Questionnaire oriented information gathering regarding this risk factor has been extremely un-helpful. Such risk can only be identified either by experience or mainly because of the DC; hence, the weightage of each methodology in SI will be as follows:

Method	Suitability Index	WI	Suitability Weighted Index
EK	0.35	2	0.70
DC	0.6	1	0.6
Q	0.05	0.5	.025

Table 5: Suitability Weighted Index for risk factor No.3

D. *Staff Inexperience*

As expertise and experience of the individuals working in an organization are known to the management, therefore, the most effective way of finding the expertise of individuals is through the DC. Questionnaires have been found to be of least usefulness because of their descriptive nature and immaturity. The developers also may not like to provide the written proof about their deficiencies, etc. Experience also plays a vital role in the identification of any such risk factor;, the weightage of each methodology in SI will be as follows:

Method	Suitability Index	WI	Suitability Weighted Index
EK	0.35	2	0.70
DC	0.50	1	0.50
Q	0.15	0.5	.075

Table 6: Suitability Weighted Index for risk factor No.4

E. *Staff Turnover*

Staff turnover is one of the most dynamically faced challenge not only in the software development organizations but also in general as well. The change of job can hardly be evaluated by the questionnaire. DC can be of help only when the employee has shown its intentions in advance to leave the job. The experienced managers, however, can estimate and expect some staff turnover during the lifetime of the project. In estimating the staff turnover, nothing has been found more appropriate than the experience which allows the managers to plan ahead and train and attach some extra workforce with the project. , The weightage of each methodology in SI will be as follows:

Method	Suitability Index	WI	Suitability Weighted Index
EK	0.7	2	1.4
DC	0.25	1	0.25
Q	0.05	0.5	.025

Table 7: Suitability Weighted Index for risk factor No.5

F. *Management Changes Circumstances*

Change of circumstances to meet the deadlines is considered normal when the requirements are deficient in exploration at the beginning of the project. The change in requirements directly effects the time and budget allocated for the project, in order to cope with this the manager needs to change circumstances to accommodate the changes. This risk factor is quite evident and its existence can be established either by the questionnaires or by communicating to the manager and customers directly. Experience and intuition still play a vital role in the identification of any such risk factor. The weightage of each methodology in SI will be as follows:

Method	Suitability Index	WI	Suitability Weighted Index
EK	0.5	2	1.0
DC	0.25	1	0.25
Q	0.25	0.5	.125

Table 8: Suitability Weighted Index for risk factor No.6

G. *Loss Of Actual Documents And Data*

The loss of documents and data is not a common risk factor in the software development life cycle. Loss of data and document can be for any reason, including the theft, fire, loss etc. An experienced manager can have ample wisdom about the risk and maintains the data on multiple sites and servers and duplicate copies of documents are also maintained. DC has also a major role to play in the identification of this risk factor. The questionnaire methodology has been observed to be of least significance among the three; hence, the weightage of each methodology in SI will be as follows:

Method	Suitability Index	WI	Suitability Weighted Index
EK	0.60	2	1.20
DC	0.15	1	0.15
Q	0.25	0.5	.125

Table 9: Suitability Weighted Index for risk factor No.7

H. *Low Estimation Of Time*

Due to the inbuilt and perhaps genuine problem of the requirement statement by the customer, the development team remains in continuous loop for a fairly longer period to time to finalize the requirements and based on that the time and budget for the accomplishment of the project are also calculated. Questionnaire’s approach may be of slight help but takes a huge amount of time and hence stands less adequate in the race of being the fittest. DC allows the manager to communicate with the development team and the customer to manage this risk. It has been observed that an experienced manager will already know that this risk can come and he can identify such risks with the experience, hence, the weightage of each methodology in SI will be as follows:

Method	Suitability Index	WI	Suitability Weighted Index
EK	0.55	2	1.1
DC	0.35	1	0.35
Q	0.1	0.5	.05

Table 11: Suitability Weighted Index for risk factor No.9

I. *Lack Of Intuition*

Use of intuition plays a major role in smelling the out of box problems and can suggest the possible solutions. The lack of intuition may mean that a development team works more and yields less. In order to make the team productive, it is necessary that they are advised to learn from experience, use the re-usable code, be coherent with the circumstances and also keep their efforts synchronized. The lack of intuition must be identified by the higher management and when identified should be immediately in place. Although the identification of this risk factor can be done well by the experience and direct communication methodologies, yet the questionnaire methodology has been of adequate importance in the identification of this risk factor, the weightage of each methodology in SI will be as follows:

Method	Suitability Index	WI	Suitability Weighted Index
EK	0.45	2	0.90
DC	0.30	1	0.30
Q	0.25	0.5	0.125

Table 12: Suitability Weighted Index for risk factor 10

J. *Developer’s Lack Of Commitment*

The project starts with a positive node assuming that the work force deployed on the project is loyal, motivated, and committed but sometimes the situation may be otherwise. It is of utmost importance that the roles of each individual are discussed before they are assigned. This can help in keeping the developers committed to their work. As this risk factor is evident only after the start of the project, a good manager can identify this kind of risk before the start of the project. Such risk factors can be identified either by using the experience or by direct communication but not by questionnaire’s methodology by any means as an un-committed developer will not like admitting about its lack of commitment, hence, the weightage of each methodology in SI will be as follows:

Method	Suitability Index	WI	Suitability Weighted Index
EK	0.7	2	1.40
DC	0.25	1	0.25
Q	0.05	0.5	0.025

Table 13: Suitability Weighted Index for risk factor 11

K. *Customer’s Dis-Satisfaction*

With the emergence of agile computing and prototype models of software development the customers role as an active entity have increased gradually, over the time. Customers now have the liberty to show the consent about the development under consideration. Keeping in view, that a dis-satisfied customer may cause the funding uncertainty the manager can use the questionnaire or direct communication to get the feedback of the customer and can elaborate on that by using his experience, hence,

Method	Suitability Index	WI	Suitability Weighted Index
EK	0.35	2	0.70
DC	0.4	1	0.4
Q	0.25	0.5	0.125

Table 14: Suitability Weighted Index for risk factor 12

L. *Change In Hard-Ware Defaults*

This risk factor is more common for the products that are not developed for some specific customer but for the public. The development team must ensure that they do not waste (take) a huge amount of time to develop the product, so that the hardware defaults do not change when the product becomes available to use. This has to be done with immense speed, as the hardware defaults are changing dynamically, presently. The manager of the development team may not have the liberty to use the direct communication with the firms developing the hardware. Thus, generally, the manager has to rely either on his own experience or on the questionnaire that may contain the probabilistic questions to forecast about the future development in the hardware defaults. Even if such changes in the hardware are known, the development team may not easily adopt them with the same pace. Only the questionnaire and experience methodologies can be used hence, the weightage of each methodology in SI will be as follows:

Method	Suitability Index	WI	Suitability Weighted Index
EK	0.75	2	1.50
DC	0	1	0
Q	0.25	0.5	0.125

Table 15: Suitability Weighted Index for risk factor 13

M. *Development team continuous work*

This risk factor originates either because of the change in the requirements or because of the continuous business of the organization. It has been observed that the software developers have to work more than their fixed timings in order to meet the deadlines. It is therefore important that sufficient manpower is placed to e ensure that each employee works for not more than 4o hours a week, in normal circumstances. As the future business of the organization can hardly be forecasted, therefore the direct communication and questionnaire can help in identifying this risk factor hence, the weightage of each method in SI will be :

Method	Suitability Index	WI	Suitability Weighted Index
EK	0.05	2	0.1
DC	0.6	1	0.6
Q	0.35	0.5	0.175

Table 16: Suitability Weighted Index for risk factor 14

N. *Requirement Postponement*

Requirement gathering is difficult because of the fact that the customer may not explicitly mention what he needs. In such case, where the requirements are hard to find it is properly inadequate to postpone the gathered requirements. The idea of postponement comes only when the development team tries to make the customer happy by showing him something instead of the complete working product. Being up-to-date with the project scope and milestones the project manager can directly communicate with the development team and customer to see if some requirements can be scrubbed. An experienced manager can also prepare a questionnaire for the customer to see which requirements can be postponed, if any. Therefore, the weightage of each methodology in SI will be as follows:

Method	Suitability Index	WI	Suitability Weighted Index
EK	0.3	2	0.60
DC	0.35	1	0.35
Q	0.35	0.5	0.175

Table 17: Suitability Weighted Index for risk factor 15

O. *Immature Coding Practices*

The implementation being the core of the project requires more attention as compared to any other phase in software development. In order to ensure that the development team is doing the coding accurately, purposefully and error free, it is necessary that suitable coding practices are introduced in the organization. The employees may be trained and test for having the adequate standards of software development. If the coding standards are inadequate, the manager has to know this much earlier otherwise the failure or at least delay of the project is guaranteed. Direct communication, Questionnaire, and experience are all important to identify this risk factor respectively. Hence, the weightage of each methodology in SI will be as follows:

Method	Suitability Index	WI	Suitability Weighted Index
EK	0.35	2	0.70
DC	0.40	1	0.40
Q	0.25	0.5	0.125

Table 18: Suitability Weighted Index for risk factor 16

P. *Presence Of Bugs And Errors*

It is adequately important that the developer unit test each module and piece of code that they have developed, in order to reduce the chances of errors at later stage. More lately an error is identified, the cost to rectify will be higher. Direct communication and questionnaires can help in the identification of these risk factors while the experience can help in rectifying the identified errors. Hence, the weightage of each methodology in SI will be as follows:

Method	Suitability Index	WI	Suitability Weighted Index
EK	0.1	2	0.20
DC	0.65	1	0.65
Q	0.25	0.5	0.125

Table 19: Suitability Weighted Index for risk factor 17

Q. Over-Acceptability Of Product And Insufficient Data Handling

Innovations are generally appreciated in any domain. Sometimes, the product developed by a firm is overwhelmingly welcomed in the market and hence the stress on application increases both: in terms of access rate and in terms of data storage. If any such application has been publicized with a limited storage and inadequate response time the likelihood for the application crash will increase. The manager, while developing the product must also idealize about the overwhelming success of the project. The manager must try to provide as much functional facilitations as possible by not disturbing the efficiency and reliability of the system. As such situations are not common, experience of manager may not be of adequate help. Therefore this risk factor can better be identified by using the direct communication and questionnaires. The weightage of each methodology in SI will be as follows:

Method	Suitability Index	WI	Suitability Weighted Index
EK	0.05	2	0.10
DC	0.6	1	0.60
Q	0.35	0.5	0.175

Table 20: Suitability Weighted Index for risk factor 18

R. Hackers, Viruses And Trojan Horse Etc

The testing team must ensure that the system implemented is error free and must ensure that a mechanism is in place to restrict any friendly or unfriendly program to access the system without permission. The manager may also contribute to provide the updated versions of anti-viruses to ensure the maximum safety against any such event. Although this risk can more easily be identified with experience, yet the orientation of this risk factor is also possible through direct communication with development team and also by providing the questionnaires to reply accordingly. The weightage of each methodology in SI will be as follows:

Method	Suitability Index	WI	Suitability Weighted Index
EK	0.5	2	1.0
DC	0.35	1	0.35
Q	0.15	0.5	0.075

Table 21: Suitability Weighted Index for risk factor 19

S. Delayed Implementation Suffering

Software requirements are not easy to determine and determined requirements must be implemented without any delay. A delayed requirement implementation makes the job of the development team difficult and consumes extra resources. Every experienced manger is aware of the problems that can be faced because of the delayed implantation. The weightage of each methodology in SI will be as follows:

Method	Suitability Index	WI	Suitability Weighted Index
EK	0.8	2	1.60
DC	0.15	1	0.15
Q	0.05	0.5	0.025

Table 22: Suitability Weighted Index for risk factor 20

T. Market Denial

Market denial has been a serious issue in the product development. The company's management must have done adequate study about the acceptability of the product in the market before the actual work on the product starts. The complete or partial market denial after the competition of a product can suffer the business and market reputation of an organization. The organization can be in direct communication with the market or can put a survey to identify the acceptability of a specific product. The experienced manager can also use its intuition in this regard. The weightage of each methodology in SI will be as follows:

Method	Suitability Index	WI	Suitability Weighted Index
EK	0.25	2	0.50
DC	0.60	1	0.60
Q	0.15	0.5	0.075

Table 23: Suitability Weighted Index for risk factor.21

U. Over Estimation About Workers Skills

The cost to be over-optimistic is very high is software cost/time estimation. While calculating the cost and time required completing the project the analysts and managers sometime over estimate the skill of their workers and under estimate the scope of the project. This leads to a huge failure as the movement for developing the software starts and immediately the management knows about the risk of miss-calculation and realizes about the over estimation about the workers skill. Worker's skills are generally known and can be further tested either through questionnaire or by communicating directly. The weightage of each methodology in SI will be as follows:

Method	Suitability Index	WI	Suitability Weighted Index
EK	0.3	2	0.60
DC	0.45	1	0.45
Q	0.25	0.5	0.125

Table 24: Suitability Weighted Index for risk factor 22

V. *Lack Of Technical Feedback*

The requirement gathering process requires a thorough consideration and effective communication at the level of team leader/analyst and technical people at the customer side. The head of organization must not sign a contract without consulting his technical team to minimize the chance of reduction in profit. The development team must try to cover all requirements in the first iteration and not to leave any requirements unaddressed. It has been observed that more requirements identified in the beginning leads to less changes in the future. By expecting only a few changes, in the future it can be expected that the project can lead to a success. The experienced manger can identify this risk either by direct communication with customer or by putting a questionnaire. The weightage of each methodology in SI will be as follows:

Method	Suitability Index	WI	Suitability Weighted Index
EK	0.5	2	1.0
DC	0.25	1	0.25
Q	0.25	0.5	0.125

Table 25: Suitability Weighted Index for risk factor 23

W. *Save Prestige Not Money*

It has been observed that the customer opts to get its software developed from the reputed software development firms only. The reputation of firms is decided not only based on revenues but also on the basis of the goodwill and cordial relationships that they have with other groups. A failed project not only harms the revenues of the firm but also disturbs the reputation as well. Therefore, the firms try their hard not to let a project fail and even at the cost of financial losses, they would like to save their name to maintain the reputation and goodwill of the market.

It is imperative to state that a risk should always be identified before it actually starts harming the system. Once the risk has shown his presence, it doesn't remain in isolation and invites other risk factors to make a mesh and insure the project to delay if not fail at all. So in order to continue gaining business in the future, the firm may like to develop the project successfully even by going in financial deficit. The experienced manger can identify such situation by communicating with management and customers: directly or by sending questionnaires. The weightage of each methodology in SI will be as follows:

Method	Suitability Index	WI	Suitability Weighted Index
EK	0.6	2	1.20
DC	0.25	1	0.25
Q	0.15	0.5	0.075

Table 26: Suitability Weighted Index for risk factor 24

X. *Economic Distortion*

The management of software development firm must try to commit advance payment from the customer if the economic situation of the country/market is not stable. In the economic crisis, the firm must try maximizing its profit and should try to provide benefits to the employees to enable them to face the poor economic situation. Although economic distortions may be difficult to identify well in time, yet the experienced managers can have adequate vision to predict such events. The manager can communicate with top management and customers to identify the economic distortion. The weightage of each methodology in SI will be as follows:

Method	Suitability Index	WI	Suitability Weighted Index
EK	0.6	2	1.20
DC	0.25	1	0.25
Q	0.15	0.5	0.075

Table 27: Suitability Weighted Index for risk factor 25

Y. *Building Loss/Fire*

The firm must ensure that the working environment across the organization conducive and safe for the employees. Proper smoke detectors and fire alarms must be installed in the building to detect the fire and the emergency exit should be provided .The organization must also ensure that the building codes have been followed and the structure is according to the prescribed standards. The management must keenly observe the building structure and the life of building must also be known by communicating with architects and management. The weightage of each methodology in SI will be as follows:

Method	Suitability Index	WI	Suitability Weighted Index
EK	0.7	2	1.40
DC	0.25	1	0.25
Q	0.05	0.5	0.025

Table 28: Suitability Weighted Index for risk factor 26

VII RESULTS

The values of Suitability Weighted Index (SWI) presented in the section 6 of this paper are summarized here. It can be observed that the cumulative value of EK methodology is 23.50, which is highest as compared to other two values. This means that the usage of EK methodology for risk identification is most suitable in terms of identifying the risks and is cost and resource effective. As an outcome of the experimental evidence presented in the section 5-6 of this paper it is highly recommended that EK be used for the identification of risks that are probable to be present in the software development environment. The DC methodology owes the value of 9.8, which is the second best among the three. The use of questionnaire, although, may be helpful in some situations, yet it is not encouraged as a methodology for identification of all risks factors. Table 29 contains the details:

Risk No.	EK	DC	Q
1	0.6	0.6	0.05
2	1.0	0.4	0.05
3	0.7	0.6	0.025
4	0.7	0.5	0.075
5	1.4	0.25	0.025
6	1.0	0.25	0.125
7	1.2	0.15	0.125
8	1.10	0.35	0.05
9	1.10	0.35	0.05
10	0.90	0.30	0.125
11	1.4	0.25	0.025
12	0.7	0.4	0.125
13	1.50	0.0	0.125
14	0.1	0.6	0.175
15	0.6	0.35	0.175
16	0.7	0.4	0.125
17	0.20	0.65	0.125
18	0.10	0.6	0.175
19	1.0	0.35	0.075
20	1.6	0.15	0.025
21	0.5	0.60	0.075
22	0.6	0.45	0.125
23	1.0	0.25	0.125
24	1.20	0.25	.075
25	1.20	0.25	0.075
26	1.40	0.25	0.075
Total	23.50	9.55	2.40

Table 29: SWI for EK, DC and Q methodology

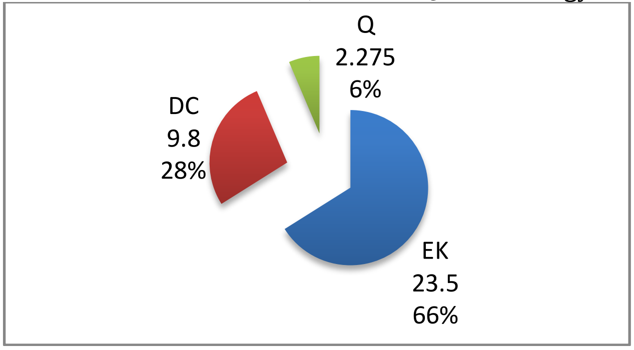


Fig. 2. SWI for EK, DC and Q methodology

From table 29, it can be observed that out of the 26 risk factors discussed in this paper, the SWI for EK methodology for 21 risk factors is highest as compared to that of the other two methodologies, while SWI for four factors is highest for Dc mythology and for one risk factor SWI for DC and EK are equal. Q methodology in none case could become the methodology of first choice as observed from table 29, although it plays a supportive role for identification of some risk factors partially.

It can therefore be concluded that the choice of methodology has a direct relation with the resources and time incurred as the consequence of that methodology. One of the reasons for the overwhelming acceptability of EK is being less expensive and highly responsive. The cost of Identification (COI) for each methodology can be identified with the formula:

$$COI=1/(\sum_{i=1}^{n} M) \qquad (1)$$

Where n=26: the no. of risk factors
M=Methodology

$$COI_{EK}=1/(\sum_{i=1}^{n} EK) \qquad (2)$$
$$=1/23.5$$
$$=0.0426$$

$$COI_{DC}=1/(\sum_{i=1}^{n} DC) \qquad (3)$$
$$=1/9.55$$
$$=0.105$$

$$COI_{Q}=1/(\sum_{i=1}^{n} Q) \qquad (4)$$
$$=1/2.40$$
$$=0.416$$

From equation 1, 2, 3 and 4 it can be observed that COI for Q methodology is highest while the COI for EK methodology is the lowest, which provides the justification for usage of EK for the purpose of risk identification.

VIII CONCLUSION

As per the survey results and the results of the equation 2,3 and 4, propose that the 'EK' is the most appropriate strategy for the identification of software risks. From Table 29, and equation 2, 3, 4 it can be concluded that the performance of EK methodology is more than twice better than 'DC' and even 10 times better than 'Q' methodology in terms of usage of resources, budgeting, time and effectiveness. It can therefore be concluded that the usage of 'EK' methodology for the purpose of identification of software risk factors is highly appropriate. It is worthwhile to note that only 'EK' can help in identifying the entire risk factors single handedly without the support of any other methodology. The 'DC' methodology has performed better than 'EK' in only a few identifications. It is also recommended that the reader establish sound knowledge about the risks and the methodologies before taking a final decision. But in most cases 'EK' is the default solution. So, even a random choice of 'EK" has more probability of success as compared to any other methodology.

IX FUTURE WORK

This work can be expanded to more generalized version of the risk identification model by suggesting other techniques for the identification of the risk factors and also by introducing ore adequate methodologies to identify these risk, having introduced the methodologies, the SI will become more precise and hence will be able to produce more realistic and appropriate feedback about the choice of methodology for the identification of risk factors.

X References

1) Xiangnan Lu. , Yali Ge (2004), “Risk analysis in software development project with Owners and contractors” ”, Engineering Management Conference, 2004 IEEE International Volume 2, 18-21 Oct. 2004 Page(s):789 – 793.
2) Miller J , Gorski J (2004) “Risk identification Patterns for software projects ” Foundations of Computing and Decision Sciences Vol. 29, No. 1-2, pp. 115-131.
3) Schmidt, R., et al., Identifying software project risks (2001): An international Delphi study. Journal of Management Information Systems, 2001, 17(4): pp.5-36.
4) Powell, P.L., Klein, J.H. 1996. Risk Management for Information Systems Development. Journal of Information Technology, Vol 11, 309-319.
5) Linda Wallace and Mark Keil,“Software project risks and their affects on Outcomes”, April 2004/vol 47 No.4 Communication of ACM
6) Risk Register http://www.brighthub.com/office/project-management/articles/3247.aspx
7) Risk categorization http://www.idrisk.com/risk_consulting.asp
8) http://www.project-management-basics.com/risk-management-4.shtml risk management approach
9) http://www.businessdictionary.com/definition/risk-identification.html risk definition
10) Kamran Khan, Zaeem ahmad, Basit Shahzad, “Adressing problems and risks in software development life cycle”, 7th CIIT Workshop on Research in Computing, held at Lahore June 23, 2008, pp 123-128.
11) Basit Shahzad, Sara Safvi, “Risk mitigation and management scheme based on risk’s priority”, accepted in 7th WSEAS conference held in Hangzhou-China, April 6-8, 2008. pp 86-91
12) Basit Shahzad, Javed Iqbal, “”Software Risk Management – Prioritization of frequently occurring Risk in Software Development Phases. Using Relative Impact Risk Model”, 2nd International Conference on Information and Communication Technology (ICICT2007), December 16-17, 2007, IBA Karchi.
13) Javed Iqbal, Basit Shahzad, “Prioritization and Handling of Commonly Occurring Risks– An Analytical Approach”, 5th conference on Intelligent Systems & Networks, February 22-24, 2008(ISN-2008),Haryana, India
14) Basit Shahzad, Javed Iqbal, Zeeshan ul Haq, Saleem Raza and Salman Aslam “Distributed risk analysis using relative impact technique” , 3rd Asian Conference on Intelligent Systems and Networks, February 24-25, 2006, Haryana-India, pp 433-439.
15) Basit Shahzad, Najam us saqib, “Escape from delays and failures for software solutions”, 6th Jordanian International Electronic and Electrical Engineering Conference in Amman-Jordan, March 14-16, 2006
16) Basit Shahzad, Tanvir Afzal, “Enhanced risk analysis and relative impact factorization”, 1st IEEE international conference on information and communication technology, IBA Karachi, August 27-28, 2005 ,pp 290-295.
17) http://standards.ieee.org/catalog/olis/se.html
18) IEEE software engineering standards collection, IEEE-STD-610.12, 1990.
19) Boehm, B.W., 1991. Software risk management: principles and practices.IEEE Software 8 (1), 32–41.
20) Barki, H., Rivard, S., Talbot, J., 1993. Toward an assessment of software development risk. Journal of Management Information Systems 10 (2),203–225.
21) Heemstra, F.J., Kusters, R.J., 1996. Dealing with risks: a practical approach. Journal of Information Technology 11 (4), 333–346.
22) Moynihan, T., 1997. How experienced project managers assess risk. Software, IEEE 14 (3), 35–41.
23) Ropponen, J., Lyytinen, K., 1997. Can software risk management improve system development: an exploratory study. European Journal of Information Systems 6 (1), 41–50.
24) Han, W.-M., Huang, S.-J., 2007. An empirical analysis of risk components and performance on software projects. journal of Systems and Software 80 (1), 42–50.
25) Source: Patterns of software failures and successes, Capers Jones, 1996

Global Journals Guidelines Handbook 2010

www.GlobalJournals.org