

AN ENHANCEMENT OF FRAMEWORK SOFTWARE RISK MANAGEMENT METHODOLOGY FOR SUCCESSFUL SOFTWARE DEVELOPMENT

¹ABDELRAFE ELZAMLY, ²BURAIRAH HUSSIN

^{1,2} Information and Communication Technology, Universiti Teknikal Malaysia Melaka (UTeM)

E-mail: ¹abd_elzamly@yahoo.com, ²burairah@utem.edu.my

ABSTRACT

Despite much research and progress in the area of software project management, many software projects have a very high failure rate. This risk is not always avoidable, but it is controllable. The aim of this paper is to propose the new framework software risk management methodology for successful software project. There are 5 main phases such as identification risk, risk analysis and evaluation, risk treatment, risk controlling, risk communication and documentation for software development life cycle. Indeed, our approach focuses on identifying software risk factors, and risk management techniques and on how to manage software risk factors with statistical and mining techniques. Our framework derived data from questionnaires and historical data from software companies. Successful software project risk management methodology will greatly improve the probability of software project success.

Keywords: *Software Project Management, Risk, Software Risk Management Methodology, Software Development Lifecycle, Software Risk Factors, Risk management techniques, Statistical and mining techniques*

1. INTRODUCTION

Software development projects still fail to deliver acceptable systems on time and within budget, regardless how much effort we put for the success of software projects. Analyzing potential risks, and making decisions about what to do about potential risks, the risk management is considered the planned control of risk because of the involvement of risk management in monitoring the success of a software project. Integrating formal risk management with project management is a new phenomenon in software engineering and product management community. It requires the project managers to be involved in a project from the concept phase to the product's retirement [1]. Moreover, risk is an uncertainty that can have a negative or positive effect on meeting project objectives. Risk management is the process of identifying, analyzing and controlling risk right through the life of a project to fulfill the project objectives [2]. Clearly, the success or failure of software projects are generally evaluated based on three dimensions: budget, schedule, product functionality and quality [3].

Nevertheless, the goal of risk management at early identification and recognition of risks and then actively changes the course of actions to mitigate and reduce the risk [4]. Risk is becoming increasingly important in the process of

understanding the factors that contribute to software project success. This is a result of the size, complexity and strategic importance of many of the software project currently being developed. Today, we must think of risk is a part of software project lifecycle and is important for a software project survival [5]. On the contrary, risk management aims to read risks as improvement opportunities and provide inputs to growth plans [5]. Masticola [6], described risk management to mean any activity that is intended to help software project managers to understand and manage the risk of serious budget overruns in software projects.

According to Taylor[7], we should be applied techniques consistently throughout the software project risk management process. Risk management is a practice of controlling risk and practice consists of processes, methods, and tools for managing risks in a software project before they become problems [8]. Previous study had consider many aspect of risk management including principles and practices for risk identification, analysis, prioritization, and mitigation [9]

In reality, there are many articles were interestingly and describe risk management academically, but we need practical models to assess risk and forecast risk in software project because there are innumerable reasons that make projects in different IT application domains fail [10]. However, the development of software with



software risk management methodology is hardly ever found. Therefore, it is important to combine between software life cycle with software risk management according to techniques such as qualitative, quantitative, and mining techniques. **The objective of this study** is: To propose conceptual framework for software project risk management based on qualitative, quantitative, and mining techniques.

2. WHY SOFTWARE PROJECTS FAIL?

Billions of dollars are spent in software projects due to their success is important to companies, departments, and managers [11]. On the other hand, it is reported seven reasons why information technology projects fail? such as poor project planning, insufficient communication, lack of change, financial, and performance management, failure to align with constituents and stakeholders, ineffective involvement of executive management, lack of qualified team members in the areas of soft skills, ability to adapt, and missing methodology and tools [12]. Nevertheless, software project manager have to determine the techniques and models to identify, analysis, and manage risks in software project. The techniques insight is to estimate processes and grant the outlook of a successful software project. Therefore, software project managers need to have sufficient knowledge how to use techniques that comfortable to manage software risk factors and risk management techniques. For example, if the budget overruns, the performance is poor. Therefore, we need to increase efforts to estimate budget and schedule by techniques to pull off the success of software project early. As well as managing software project risks associate with qualitative risk analysis techniques, quantitative risk analysis techniques, and mining risk analysis techniques.

However, the software company has to solve some equations like the suitable risk management methodology practice and the ways to identify and manage risks by using techniques through software development lifecycle. More than software project manager usually responsible for the selection of appropriate risk management practices and techniques to assess risks throughout the entire software development life cycle (SDLC). Indeed, difficulty estimation risks without more efforts, recognize the process, and distinguish among three categories risk techniques. Thus, measurement is important to progress, and it is now time for software manager to learn these techniques [13]. According to [14], the best example for failure is the FBI Virtual Case File project: Primary causes

for the failure of complex IT projects as inadequate planning, unrealistic goals and objectives, objectives changing during the software project, unrealistic schedules, low user involvement, ineffective communicate and inappropriate skills. He also referred to five years of development but \$U.S. 170 million in cost had been lost. However, they referred to about 2100 sites (www) that described more than 5000 reasons that software projects fail, ranging from the poor use of technology to ineffective of communication to management in attention. In addition, due to the massive resistance of end-users, billions of dollars have been wasted on failed projects, and a lot of very expensive projects had to be shelved after a short period of time [10]. Most researchers are considering this question. Software development project has a reputation for failure even though, software has been successfully applied in a large variety of areas [15]. Consequently, the manager of the risk management process, who review data, assigns the other responsible of people and evaluates identified risks [16]. Finally, not every risk factor is fully controllable, and several risk factors exceed the authority of software managers[17].

3. SIZING FAILURE/ SUCCESS SOFTWARE PROJECTS:

Software projects are complex to manage and too many of them end in failure. Annual United States spent on software projects reached approximately \$250 billion in 1995[18]. In addition, the 2004 report showed[19]: Successful projects: 29%, canceled projects cost \$55 billion annually, challenged projects: 53%, failed projects: 18%. On the contrary, the CHAOS results showed that 35% of projects are successful, 46% are challenged, and 19% fail(first quarter research report, 2007). According to [21], the report showed that software projects now have a 32% success rate compared to 35% from the previous study in 2006 and 16% in 1994. Further than, the Standish Group estimates that in 1995 American companies and government agencies spent \$81 billion for cancelled software projects [22]. According to [23], any software projects are developed by small companies, the development organization may not survive a combination of project failures.

According to [24], IT consultant Charette, writing in the September 2005 issue of IEEE Spectrum: Of the \$1 trillion that will be spent worldwide on technology in 2005 year, many billions will be wasted on software mistakes that are entirely preventable. Also the report explained

[25], the problem only gets worse as IT becomes anywhere. Organizations and governments spent an estimated US\$1 trillion on IT hardware, software, and services worldwide in 2005. The United States [26] spent more than \$250 billion each year on IT application development of approximately 175,000 projects. The average cost of a development project for a large company is \$2,322,000; for a medium company, it is \$1,331,000; and for a small company, it is \$434,000. However, a great many of these projects will fail. Unfortunately, there are no any official statistical reports regarding on failure, challenge and success of any project in Palestine software organizations. According to [27], a significant factor that makes Palestine attractive to potential partners outside of the region is cost. A typical senior software engineer in Palestine is charged out at between \$80-300 per day, with an average of \$172, according to this research. Palestinian companies offer a beneficial cost structure with good technical expertise which it is very good value for money.

4. CONCEPT OF RISK:

Risk is associated with all phases of the software project, such as planning, analysis, design, implementation, and maintenance. Moreover, risk is a challenging concept to define, and understand which usually return to different things and different people[28]. Also, everything has risk, whether the risk is high or low. Historically, the risk is defined as the possibility that the actual input variables and the results may vary from those originally estimated and intrinsically risk may be either positive or negative[29]. Han and Huang (2007) the likelihood of occurrence of each software risk is different and its degree of impact on project cost, schedule, and quality is different. Therefore, in order to develop a good software risk management strategy and plan, the two software risk components must be taken into consideration. According to [30], risk can be defined as the possibility of suffering loss. The most common definition is associated risk, “ the possibility of suffering harm or loss, or exposure in the life cycle software project to this ” and the same author said, definitions is the fact that a risk involves uncertainty and has an impact, both the uncertainty and the impact are capable of being quantified thereby offering a numerical definition [31]:

$$\text{Risk exposure} = \text{impact value} \times \text{probability of occurrence} \quad (1)$$

5. SOFTWARE PROJECT RISK MANAGEMENT:

Risk management is well recognized as an important means of mitigating software failure [32]. Masticola also described risk management to mean any activity which is intended to help software project managers to recognize and manage the risk of serious cost overruns in software projects [6]. According to [33], explained risk management processes (RMPs) are logically consistent and structured approaches to enumerate and understand probability software risk factors and evaluate consequences and uncertainties associated with these identified software risk factors. Clearly, the success or failure of software development projects are usually assessed in three dimensions: Budget, schedule, product functionality and quality [3]. Oracle corporation also described risk management solutions enable a consistent approach for identifying, assessing and mitigating risk throughout the entire software project lifecycle [34]. Also [16], risk management means that we change our attitude towards risks. Surely, a software project without risk management faces serious problems only after the risks came to the surface as a material fact. By contract, a software project with risk management aims at early identification and recognition of risks and then actively changes the course of actions to mitigate the risk. Finally, it is proven that quite enough issues regarding risk management in software projects remain untested and lack of practical support for the success of management software risks [35].

5.1 Concepts of risk management:

Risk management is not a discrete single activity but a dynamic process, that is continuously more refined through its repetition throughout a software project's life cycle [36]. Respectively again risk management associated with all activities, conditions and events, which can affect the organization, and its ability to achieve the organization's goals[37]. However, risk management is to identify risky situations and develop strategies to mitigate the likelihood of occurrence and the negative effect of risky events[38]. In practice, risk management involves the process of risk identification, analysis, monitoring and handling. Risk management is a practice of risk controlling and practice consists of processes, methods, and tools for managing risks in a software project before they become problems [8]. Boehm talked about value-based risk

management, including principles and practices for risk identification, analysis, prioritization, and mitigation [9]. Conversely, many authors defined risk management, but difficult in practice to measure the likelihood of impact of software risk factors and determine risk management techniques, especially in software development projects[39]. Risk management focuses on assessment of the likelihood of risk occurring, risk event drivers, risk events, the likelihood of impact and the impact drivers before the risk actually takes place [17]. According to [17] risk management consists of the processes, methodologies and tools that are used to deal with software risk factors in the Software Development Life Cycle (SDLC) process of Software Project. Also dash described risk management is defined as the activity that identifies a risk; assesses the risk and defines the policies or strategies to alleviate or lessen the risk. Also, Oracle Corporation described risk management solutions enable a standardized approach for identifying, assessing and mitigating risk throughout the software project lifecycle [34].

Surely, we need to focus on software project risk management practice in order to estimate software project risks. However, the improvement of the entire development process is desirable to obtain high software quality such as the structure of rigid specification, introducing review activity, and determining feasible development plans [40]. Also, ensured to support software project from the selection of software technology such as techniques and methods which are used to achieve its goals [41].

The Software Development Life Cycle (SDLC) is a framework that is used to recognize and develop information systems and the success of software project [17]. Hence, it is an approach to develop a software project that was characterized by a linear sequence of steps that progress from start to end. The SDLC model is one of the oldest systems development models and is still probably the most commonly applied [42]. According to [17] risk management consists of the processes, methodologies and tools that are used to deal with software risk factors in the Software Development Life Cycle (SDLC) process of Software Project. Software Development Life Cycle is the process of creating and risk management techniques is used to mitigate risk it should involve in all phases include: Planning, analysis, design, implementation, and maintenance. In our thesis, it is very important to be on familiar terms with the SDLC, in order to develop software projects. Finally, we depend on

these phases to identify software risk factors and risk management techniques in software projects.

5.2 *The Proposed New Conceptual Framework for Software Project Risk Management:*

Software risk management methodology includes five phases: risk identification, risk analysis and evaluation, risk treatment, risk controlling, risk communication and documentation these contribute to software project success. In addition [43], proposed a framework for a field investigation of risk management in the context of a particular software development organization. However, it is crucial to recognize the risk management phases, the practices which are used in a software project, the ways to estimate risk throughout risk management methodology, as well as on how the risk management will affect software project success. Indeed, our approach focused on identifying software risk factors, and risk management techniques and how to manage software risk factors with statistical and mining techniques. On the other hand, our framework derived data from questionnaires and historical data from software companies. However, to explain the concept risk management methodology and risk techniques we drew map in Figure 1.

Our framework in software project risk management methodology focused on activities that include three factors follow as:

Data source: Questionnaire, historical data, etc.

Models: Risk multiple regression modelling, risk fuzzy multiple regression modelling.

Methods: Risk identification relies on risk qualitative models, risk analysis relies on risk quantitative techniques and risk mining techniques, and risk controlling relies on quantitative and mining techniques, etc.

Our framework also defined the software project risk management according to software engineering methodology (SDLC), after then produced some risk models to manage risks. However, we concentrate on the techniques which will help to manage risks in the software project lifecycle. Of course, software project managers must use practical techniques, tools and approaches to manage risk. Indeed, it is complex to implement the approaches and techniques comfortably to estimate risks that essential to achieve success in a software project. Unfortunately, quantitative and mining techniques are used as more restrictions in a software risk management practice to mitigate risks. However, this methodology based on literature review, the objectives of our thesis will be achieved, followed by a survey and discussions with 76 software project managers to estimate the

software risk factors and risk management techniques that affect the software project success. The software project manager will determine the software risk factors and control factors affecting the Software Development Life Cycle phases through execution of the software projects.

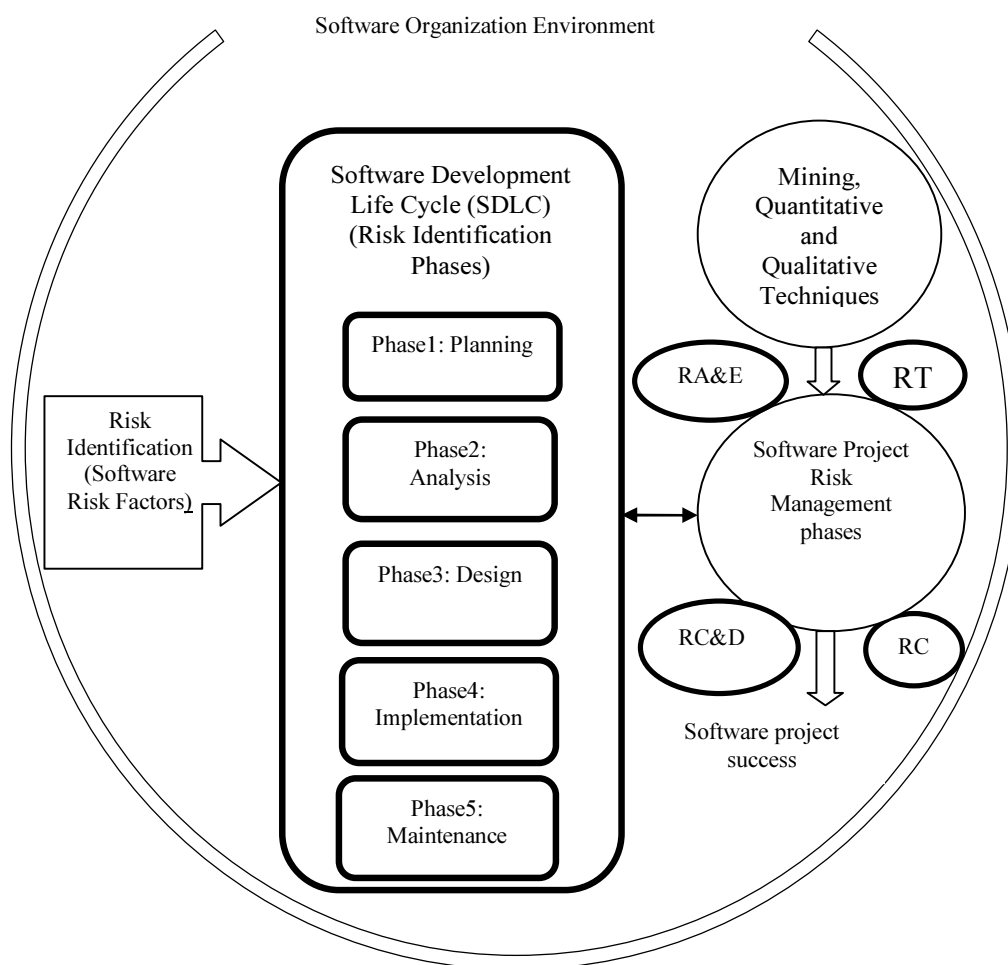


Figure 1: Conceptual Framework for Software Project Risk Management

Software risk factors affect on software development life cycle, every stage included uncertainly and certainly, risks which need to mitigate using software project risk management based on qualitative, quantitative and mining techniques to successful software project. The techniques and tools are qualitative risk management, quantitative risk management, and mining risk management based on data sources from questionnaire and historical data. Examples of

quantitative risk techniques based on statistical method are regression models, chi-square(χ^2), metrics, network analysis, cluster analysis, decision analysis, cost risk models, factor analysis, discriminant analysis, and so on. Whereas, the qualitative risk techniques based on judgment are scenario, Delphi analyses, brainstorming session, checkpoint, checklist, and worksheet that are more subjective. As for mining risk techniques, it is based on simulation analysis, fuzzy logic models,

fuzzy multiple regression, neural network models, genetic algorithm, and heuristic algorithm. The best framework explains the improvement of the software project risk management methodology and helps software project managers to identify and analysis risks at early state which associated with software project life cycle to select a fit approach and models.

However, we focus on software risk factors that take place during software project life cycle and managing risks by new statistical techniques based on proposed new framework software project risk management.

5.3 Main Software Project Risk Management Phases:

Risk management is the process which allows IT managers to balance the operational and economic costs of protective measures and achieve gains in mission capability by protecting the IT systems and data that support their organizations' missions[44]. On the contrary, they illustrated a seven-step risk management process which can be implemented to increase the likelihood of software project success. These steps include identifying risk factors, assessing risk probabilities and effects, developing strategies to mitigate identified risks, monitoring risk factors, invoking a contingency plan, managing a crisis, and recovering from a crisis[45]. Furthermore, risk management will help them to decrease software project failure and manage software project risks[46]. In addition [47], to recognize the risks and managing risk response through standard response types such as mitigation and added contingency further ensure success of software project. Besides, the chances of establishing a win-win relationship between the contractor and the subcontractor go up drastically with the tools in place such as Oracle's Primavera Risk Analysis. While, some of authors point out methodologies to guide organizations and project managers to reduce risks but software projects is still failure. Therefore, we look forward to establish methodology to help organization based on techniques and models in order to enhance our software project success and estimate risks according to this methodology.

5.3.1 The First Phase Risk Identification (RI): Concept /Approach/ Techniques

This phase involves three stages. The first stage: Risk planning includes a set of functions that are identified as continuous activities throughout the software project life cycle [8], the second stage: Risk identification associates with probability risks

in the software project and try to visualize all situations that might make things in the project go to errors, the third stage: Risk prioritization activity considers all aspects of all risk factors and then prioritizes them [48].

5.3.1.1 Risk planning stage:

This stage is important to identify the company actions. In order to establish it, we need to determine persons responsible for software project such as managers, stakeholders, programmers, developers, users, and so on. On the other hand, the risk planning includes plan that is an organized, and iterative approach for managing risk [8]. The main inputs to the risk planning step are [7]: The project charter, guidelines, the contract documents, the work breakdown structure (WBS), and network analysis. It enables better project selection decisions and more precise budget and schedule [34].

5.3.1.2 Risk Identification Stage:

Risk identification is the process of searching the environment, detecting risks, recognizing their attributes, and estimating their consequences [5]. Identification surfaces risks before they become problems and adversely affect a software project, in order to mitigate the likelihood of compromising the success of project[49]. Furthermore, it makes lists of the project-specific risk factors likely to compromise a success of project [50], [51]. Techniques which are determined involve checklists, network analysis decision trees, examination of decision drivers, cost models, and performance models. However, methods that are able to sustain risk identification based on qualitative techniques involve checklists of probability risks, questionnaires, interviews and brainstorming, and reviews of plans[48]. Therefore, reviewed tools are able to estimate risk-obstacles identification through checklist, questionnaires and brainstorming session with the stakeholders[52]. Miler and Górski [53], presented a process model-based approach to software project risk identification which involves explicit modeling of software processes and identifying risk by two techniques: Metrics of process structure and focus on the differences between the actual and the referential model. Risk identification[3]: Identification of risk incidents intimidating the success of project as well as their risk factors, recognize of the risk scenarios.

According to [54], risk identification appropriate countermeasures can be taken to reduce the probability of failure. It is to identify the reasons of risk according to software development project lifecycle. They used scenario models to represent risk and contingency strategies impact on software project. They expected the process to be applied in several iterations; each one discovering new risk archetypes and adding new information to previously documented archetypes.

Consequently, it involves the list of risk factors, is performed to be later categorized into appropriate branches in the classification system [55]. In this stage, it is essential to establish a risk picture clearly, identify software risk factors, risk management techniques, and thus clarify the effect of measure's risk such as qualitative techniques and quantitative techniques.

5.3.1.3 Risk Prioritization Stage:

We should categorize the software risk factors and select the best strategies based on results analysis to reduce risks after risk planning, identifying software risk factors and risk management techniques have been carried out. On the other hand, risk prioritization involves analyzing the potential effects of the risk event and impact value. The degree of risk depends on two properties: The likelihood and impact on the software project if it works [56]. We can calculate the risk exposure and their relationship based on the potential consequences and the likelihood of the risk event occurring. Hence, we can prioritizing risks as illustrated in Table1 show the degree of risk [48].

Table 1: Categorization of Degree of Risk [48], [56].

Range	Probability	Impact		
		High	Medium	Low
0.7-1.0	High	Extreme	High	Medium
0.3-0.7	Medium	High	Medium	Low
0.0-0.3	Low	Medium	Low	Minimal

There are many statistical techniques to determine the scale of risks. But, the easy way to classify and determine the risks impact is whether it is extreme, high, medium, low, or minimal. Based on the table above, quantitative analysis is when we determine a likelihood risk and consequence. Sometime, historical data and reports are not obtainable to estimate risk factors quantitative. Therefore, we need to depend on software project manager's experience and skills to determine the risk prioritization[48].

Indeed, he described impact of any risk on your project is given by the following equation[57]:

$$\text{Risk impact} = \text{likelihood} \times \text{consequence} \quad (2)$$

A risk's likelihood is classified according to a three-point scale of Low/Medium/High or a five-point scale of Very Low/Low/Medium/High/Very High. They determined a risk's impact at the range of 1 to 15 based on the Equation 1. Risks are ranked based on risk rank matrix which contains risk category (1- Unknown, 2-Low, 3-Medium, 4-High, 5-Fatal), likelihood of occurrence, and risk impact (1- Low, 2-Medium, 3-High) [58]. On the other hand, risk classification is intended to classify risk for a collective view point on a group of factors, which assist project managers to discover the group[55]. Finally, risk prioritization makes a ranked ordering of the risk factors identified and analyzed. Therefore, techniques include risk-exposure analysis, risk-reduction leverage analysis, and Delphi [50].

5.3.2 The Second Phase Risk Analysis and Evaluation (RA&E): Concept/Approach/Techniques

5.3.2.1 Risk Analysis Stage:

Risk analysis is not solely occupied with outcomes all the conditions where something could go wrong[59]. Presented risk analysis approach which represents the organizational settings of a project as a social dependency network, and identified risks originated from broken role dependency relations [32]. However, Oracle Corporation invented Primavera Risk Analysis that companies use a full lifecycle risk analytics solution which helps to integrate cost and schedule risk management. Further using the risk register feature in Primavera risk analysis allows you to make more-detailed risk assessments of activities in your projects[60].

Risk analysis contribute to analyze probability and consequences in the risk identification phase and estimate the impact, sensitivity, relationships of risks, the relationship between risk factors and risk management techniques with new techniques, analysis of risk mitigation options, analyze a certain risk mitigation strategy. Moreover, risk analysis assesses the loss likelihood and the size impact for each identified risk item and it assesses compound risks in risk-item interactions. The techniques include performance models, cost models, network analysis, statistical decision analysis, and quality-factor analysis[50].

According to [61], software project always fail. Therefore, in order to minimize the impact of risks,

risk analysis is required to be carried out. In integrated risk management (IRM), risk analysis is done along with the regular processes, as illustrated in the following list [5]: Bidding + risk analysis, Goal setting + RS, Estimation + RS, Planning + RS, Req. analysis + RS, Design + RS, Coding + RS, Test planning + RS, and Delivery + RS. Besides, risk analysis tools are designed to improve risk management by identifying possible sources of risk and evaluating them [62]. Also, we assess the risk quantitatively by questionnaire. Based on the statistical analysis of the reclaimed questionnaires we drew some conclusions about risk to contractors in software development project [54]. A risk analysis is a model for quantifying and evaluating critical event occurrence. The quantification of risk includes the estimating of the likelihood and the consequences of risk occurrence [63]. Risk analysis is process that quantifying and qualifying the degree of risk, so they pose for exposed individuals, populations, or resources and the probability that the risk will occur [64]. Finally, risk analysis solutions offer the tools for doing just this, enabling companies to identify, assess, and model risks[34].

5.3.2.2 Risk Evaluation Stage:

Risk evaluation requires a systematic research of random scenarios, including failure rates for the component as well as for the behavior of operator within an evolving environment[63]. Also risk evaluation—deciding on risk acceptance by evaluating the risks against an acceptance scale [3]. In addition, the purpose of risk evaluation is to determine the levels of the identified information risks, thus managers can compare them, according to the various levels of information risk, managers carry out various risk control strategies, and prevent risks from turning into project delay [65].

5.3.3 The Third Phase Risk Treatment (RT):

The nature of the risks within software project will manipulate which of these strategies are suitable to mitigate risk. There are four strategies for responding to risks [7]: Avoidance, transference, mitigation, and acceptance. Most responses fall under at least one of the following categories:

5.3.3.1 Risk Mitigation:

A risk mitigation plan aims to resolve risks as much as potential to reduce the impact of a source of the risk [5], [66]. Mitigation is to diminish the risk exposure. This can be achieved by reducing the

likelihood of the risk occurring, reducing the cost of experiencing the risk, or both [67]. Moreover, mitigating activities are those that arise from the mitigation of project risks. Your risk mitigation must make a set of activities that becomes part of your project plan; otherwise, it is wasted. You mitigate risks by reducing its likelihood, its impact, or both [56].

5.3.3.2 Risk Avoidance:

Avoidance means taking alternative steps so that the risk likelihood is reduced to zero as use a different type of process [67], [68]. Sometimes, it is difficult to shake off risks attached to goals. In result, we have to give up some plans or tasks [5], [66]. Avoidance strategies propose to prevent a negative effect occurring or really size impact software project [57]. Avoidance—if a risk is not accepted and other lower risk choices are available from various alternatives [69].

5.3.3.3 Risk Transfer:

Risk transfer is complete in practice by co-operation teaming project. Common examples of risk transfer also occur where the insurance is the most practical way of planning for risks [7]. Transference strategies include shifting the management risk to a third party or someone else [68]. In case status cannot deleted risk from software project. Transfer—when the risk is shared with others. Forms of sharing the risk with others include contractual shifting, performance incentives, insurance, warranties, bonds, etc[69].

5.3.3.4 Risk Acceptance:

Reasons, impact and consequences of the risk occurrence or exposure are analyzed and understood if risks are accepted, [5], [66], [68]. On the other hand, when a conscious decision is made to accept the consequences should the event occur [69].

5.3.3.5 Contingency plan risks:

Generally, contingency plans can be perceived as plans of action that are shelved for possible later use [70]. Meanwhile [7], [68], Contingency plans are plans to introduce risk response strategies. Realization of a risk is often recognized through the onset of a predefined risk trigger at which time predefined contingency plans in order to reduce the impact [39].

5.3.3.6 Elimination Risk:

When the exposure is unacceptably high or when the cost of elimination is not prohibitive, this is called as elimination of the risk. Usually, risk is eliminated in the case of low cost to respond [67].

5.3.4 The Forth Phase Risk Controlling (RC):

According to [57], the risk management process is concerned with planning your risk strategy approach, monitoring risks as your project progresses and dealing with those risks if they occur. Additionally, controlling—steering the risk reduction based on the actual effectiveness of the control measures and the levels of risk, deciding on launching of the contingency plans or closing a successfully mitigated risk[3]. According to [65], controlling risk is when project managers carry out optimal techniques for holding identified risks in order to reduce the probabilities of different kinds of risks, together with reducing losses caused by risk incidences. Enterprises have succeeded in managing risks if only the risk controlling is completed well. Control— when a process of continually monitoring and correcting the condition on the project is used. This process involves the development of a risk reduction plan and then tracking the plan [69].

5.3.5 The Fifth Phase Risk Communication and Documentation (RC&D):

According to [7], neither software project managers nor organizations are careful enough to documentations and archiving lessons learned them so that future projects can achieve the benefits. Every risk reports and lessons-learned have to be documented and archived until project manager can access risk file easy. Currently, some company establish online library to help us find solutions for software risk factors. Also to learning on risk—abstracting the experiences from risk identification, analysis and mitigation into a reusable knowledge; includes recording the specific highly context-dependent risks as well as the successfully applied risk control measures in a risk knowledge base [3]. This phase is very essential to risk management success, because we done documentation to determine what are wrongs and rights, why we went to wrong and the solutions and strategies those help to solve problems and mitigate risks.

6. TECHNIQUES FOR RISK MANAGEMENT:

There are many risk analysis techniques currently in use to evaluate and estimate risks but it is very important to choose appropriate techniques to reduce risks [71]. Furthermore, proposed a

research agenda for the use of knowledge-based tools from the perspective of evaluation benefits and risks[72]. In the ConSERV(Concurrent Simultaneous Engineering Resource View), a concept presently being developed as an intelligent knowledge based on project management techniques, can also use as risk management system[31]. However, a quantitative method determines probability occurrence and consequences risk. In order to give techniques a failure rate, we usually get data from historical data or expert opinion. According to [73], described method associated with a special fuzzy operator, namely a two-additive Choquet integral that allows modeling different effects of importance and interactions among risks. The potential of the proposed methodology is exposed through an empirical case study conducted in a Turkish software company.

Padayachee [43] a typical risk management programme is the identification of risk, usually involving checklists, questionnaires or brainstorming sessions. Also, based on fuzzy TOPSIS approach is developed with an effective algorithm to improve the quality and effectiveness of decision making. The proposed approach demonstrated using a real case involving an Iranian construction corporation and the method could distinguish successfully[49]. Dhlamini et al. [30] demonstrated the need for risk management tools in software project since the complexity of risk management increases with the complexity of the developed system. They proposed two frameworks for the development of intelligent risk management tools; neural networks and intelligent agent based. Consequently, described an approach to modeling software risk factors and simulating their effects as a means of supporting certain software development risk management activities. This simulator is a device designed specifically for the risk management activities of assessment, mitigation, contingency planning, and intervention[74]. Also, the development of a fuzzy decision support system (FDSS) for risk assessment in e-commerce (EC) development and a risk analysis model for EC development using a fuzzy set approach is proposed and incorporated into the FDSS[71]. According to [16], presented a risk management supporting tool, detailing its functionality and the user interface appearance as well as giving some design and implementation details. The tool provides in particular: Automatic risk identification from interactively answered on-line checklists, qualitative risk evaluation. Also, the risk assessment model, methods and techniques are

widely used to control risk in a software development[17]. In addition, we presented a new technique by which we can study the impact of different control factors and different risk factors on software projects risk. The new technique applies the chi-square (χ^2) test to manage the risks in a software project[75]. However, we also used new techniques the regression test and effect size test proposed to manage the risks in a software project and reduce risk with software process improvement. [76]. Furthermore, we used the new stepwise regression technique to manage the risks in a software project implementation phase. These tests were performed using regression analysis in order to compare the controls to each of the risk factors to determine if they are effective in mitigating the occurrence of each risk factor and selecting best model[77]. Most article talked about methodology without techniques that very important to manage risk in software project. And

thus rarely studies linkage among software project phases, risk management methodology and three categories techniques. Also through studying, most articles focused traditional techniques to identify risk, but we believe, project managers must use and combine among techniques throughout software project lifecycle according to risk methodology practice. However, we presented the new mining technique that applies the fuzzy multiple regression analysis techniques with fuzzy concepts to manage the risks (design phase) in a software project. [78]. On the other hand, some articles in quantitative risk techniques focused on one phase like test, maintenance, code, cost or so on and ignored remain phases with new techniques to estimate risk and analysis risk; perhaps managers used more techniques to mitigate risks. Indeed, this area needs more effort from scholars and researcher to quantitative and mining risk.

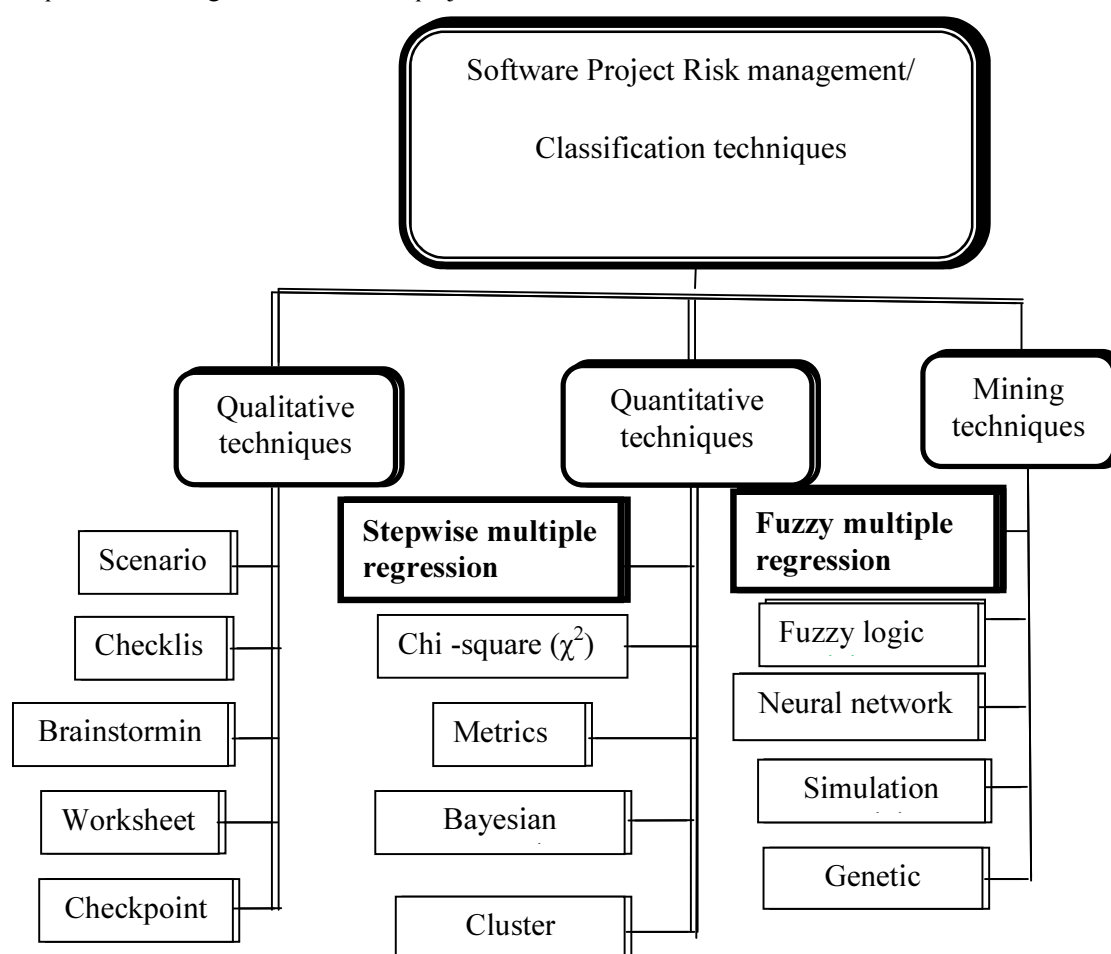


Figure 2: Classification Software Project Risk Management Techniques.

1. CONCLUSIONS

Software development lifecycle is designed to develop software. However, the management issue is not highly in the process. Thus, software managers sometimes focus into the technical decision by considering possible risk. Risk management is important to become a part of a software project. However, we proposed software risk management methodology that has five phases: Risk identification (planning, identification, prioritization), risk analysis and evaluation (risk analysis, risk evaluation), risk treatment, risk controlling, risk communication and documentation for software managers that relied on three categories techniques as risk qualitative analysis, risk quantitative analysis, and risk mining analysis. Finally, It is very important to the software project manager is known and recognizing these questions, what risk management methodology is used parallel software project methodology? Also how to use techniques influence software project success?

As future work, we will intend to apply these study results on a real-world software project to verify the effectiveness of the new techniques and approach on a software project. We can use more techniques and methods useful to manage software project risks such as neural network, genetic algorithm, Bayesian statistics, and so on.

7. ACKNOWLEDGEMENT

This work is supported by the Faculty of Information and Communication Technology, Universiti Teknikal Malaysia Melaka (UTeM), Malaysia and Al-Aqsa University, Palestine

REFERENCES

- [1] D. McNair, "Controlling risk," *Magazine Ubiquity*, vol. 2001, no. January, ACM New York, NY, USA, 01-Jan-2001.
- [2] K. Schwalbe, *Information Technology Project Management*, Sixth. course technology, cengage learning, 2010, p. 490.
- [3] J. Miler, "A Method of Software Project Risk Identification and Analysis," Gdansk University of Technology, 2005.
- [4] J. Miler and J. Górski, "Supporting Team Risk Management in Software Procurement and Development Projects," in *4th National Conference on Software Engineering*, 2002.
- [5] C. R. Pandian, *Applied software risk management : a guide for software project managers*. Auerbach Publications is an imprint of the Taylor & Francis Group, an informa business, 2007, p. 246.
- [6] S. Masticola, "A Simple Estimate of the Cost of Software Project Failures and the Breakeven Effectiveness of Project Risk Management," *2007 First Int. Work. Econ. Softw. Comput.*, May 2007.
- [7] J. Taylor, *Managing Information Technology Projects: Applying Project Management Strategies to Software, Hardware, and Integration Initiatives*. AMACOM © 2004, 2004, p. 274.
- [8] J. Sodhi and P. Sodhi, *IT Project Management Handbook*. Management Concepts, 2001, p. 264.
- [9] B. Boehm, "Value-based software engineering," *ACM SIGSOFT Softw. Eng. Notes*, vol. 28, no. 2, p. 3, Mar. 2003.
- [10] W. Al-ahmad, K. Al-fagih, K. Khanfar, K. Alsamara, S. Abuleil, and H. ABU-salem, "A Taxonomy of an IT Project Failure: Root Causes," *Manage. Rev.*, vol. 5, no. 1, pp. 93–104, 2009.
- [11] N. Gorla and S.-C. Lin, "Determinants of software quality: A survey of information systems project managers," *Inf. Softw. Technol.*, vol. 52, no. 6, pp. 602–610, Jun. 2010.
- [12] J. Gulla, "Seven Reasons Why Information Technology Projects Fail," 2011.
- [13] C. Jones, *Applied Software Measurement Global Analysis of Productivity and Quality*, Third Edit., no. Third Edition. McGraw-Hill Companies., 2008, p. 662.
- [14] A. Taimour, "Why IT Projects Fail," 2005.
- [15] P. Savolainen, J. Ahonen, and I. Richardson, "Software development project success and failure from the supplier 's perspective: A systematic literature review," *Int. J. Proj. Manag.*, p. 12, 2011.
- [16] J. Miler and J. Górski, "Implementing risk management in software projects," in *3rd National Conference on Software Engineering*, 2001.
- [17] R. Dash and R. Dash, "Risk Assessment Techniques for Software Development," *Eur. J. Sci. Res.*, vol. 42, no. 4, pp. 629–636, 2010.
- [18] M. Keil, P. Cule, K. Lyytinen, and R. Schmidt, "A framework for Identifying Software Project Risks," *Commun. ACM*, vol. 41, no. 11, 1998.
- [19] "Third Quarter Research Report," 2004.
- [20] "First Quarter Research Report," 2007.

- [21] J. Dominguez, "The CHAOS Report 2009 on IT Project Failure," 2009.
- [22] T. Clancy, "The standish group report," 1995.
- [23] H. Costa, M. Barros, and G. Travassos, "A risk based economical approach for evaluating software project portfolios," *ACM SIGSOFT Softw. Eng. Notes*, vol. 30, no. 4, p. 1, Jul. 2005.
- [24] I. S. Magazine, "Why Software Fails," *IEEE Spectrum Magazine*, 2009.
- [25] R. Charette, "Why Software Fails," 2006.
- [26] T. Clancy, "The standish group report ©," 1995.
- [27] N. White, *ICT BUSINESS DEVELOPMENT "Market Mapping of the Palestinian ICT Sector the Opportunities for Partnerships in the Region,"* 2nd Editio. Mercy Corps Knowledge Based Transformation Programs, 2010, p. 86.
- [28] D. Remenyi and A. Heafield, "Business process re-engineering: some aspects of how to evaluate and manage the risk exposure," *Int. J. Proj. Manag.*, vol. 14, no. 6, pp. 349–357, Dec. 1996.
- [29] A. Z. B. Kamaruddin, "Development of an Early Software Project Risk Assessment Application Using Case-Based Reasoning," *Universiti Teknologi MARA*, 2006.
- [30] J. Dhlamini, I. Nhamu, and A. Kachepa, "Intelligent Risk Management Tools for Software Development," *Risk Manag.*, pp. 33–40, 2009.
- [31] G. Conroy and H. Soltan, "ConSERV, a project specific risk management concept," *Int. J. Proj. Manag.*, vol. 16, no. 6, pp. 353–366, Dec. 1998.
- [32] W. Ma, L. Liu, W. Feng, Y. Shan, and F. Peng, "Analyzing project risks within a cultural and organizational setting," in *2009 ICSE Workshop on Leadership and Management in Software Architecture*, 2009, pp. 6–14.
- [33] V. Tummala and B. John, "Applying a Risk Management Process (RMP) to manage cost risk for an EHV transmission line project," *Int. J. Proj. Manag.*, vol. 17, no. 4, pp. 223–235, Aug. 1999.
- [34] Oracle, "A Standardized Approach to Risk Improves Project Outcomes and Profitability," 2010.
- [35] L. Sarigiannidis, "Software Development Project Risk Management: A New Conceptual Framework," *J. Softw. Eng. Appl.*, vol. 04, no. 05, pp. 293–305, 2011.
- [36] T. Merna and F. Al-Thani, *Corporate Risk Management and Optimal Hedging Disclosure*, Second. John Wiley and Sons, 2008.
- [37] T. Aven, *Risk Analysis: Assessing Uncertainties Beyond Expected Values and Probabilities*. John Wiley & Sons, Inc., 2008, p. 204.
- [38] M. Fan, N.-P. Lin, and C. Sheu, "Choosing a project risk-handling strategy: An analytical model," *Int. J. Prod. Econ.*, vol. 112, no. 2, pp. 700–713, Apr. 2008.
- [39] P. Bannerman, "Risk and risk management in software projects: A reassessment," *J. Syst. Softw.*, vol. 81, no. 12, pp. 2118–2133, Dec. 2008.
- [40] O. Mizuno, E. Shigematsu, Y. Takagi, and T. Kikuno, "On Estimating Testing Effort Needed to Assure FieldQuality in Software Development," *Proc. 13 th Int. Symp. Softw. Reliab. Eng. (ISSRE'02)*, p. 139, 2002.
- [41] A. Dias-Neto and G. Travassos, "Evaluation of {model-based} testing techniques selection approaches: An external replication," in *2009 3rd International Symposium on Empirical Software Engineering and Measurement*, 2009, pp. 269–278.
- [42] Z. Begum, M. S. A. Khan, M. Hafiz, M. S. Islam, and M. Shoyaib, "Software Development Standard and Software Engineering Practice: A Case Study of Bangladesh," *J. Bangladesh Acad. Sci.*, vol. 32, no. 2, pp. 131–139, May 2008.
- [43] K. Padayachee, "An Interpretive Study of Software Risk Management Perspectives," in *annual research conference of the South African institute of computer scientists and information technologists on Enablement through technology*, 2002, pp. 118 –127.
- [44] Veracode, "IT Risk Management : Guide to Software Risk Assessments and Audits," 2008.
- [45] V. Holzmann and I. Spiegler, "Developing risk breakdown structure for information technology organizations," *Int. J. Proj. Manag.*, Jun. 2010.
- [46] T. Arnuphaptrairong, "Top Ten Lists of Software Project Risks : Evidence from the Literature Survey," in *proceedings of the Intrnational multi conference of Engineers and Computer Scientists 2011 (IMECS 2011)*, 2011, vol. I, p. 6.

- [47] Oracle white paper, "Successfully Managing Contract Risk by Forming Win-Win Relationships," 2009.
- [48] P. Jalote, *Software Project Management in Practice*. Addison Wesley, 2002, p. 288.
- [49] A. Azari, N. Mousavi, and S. F. Mousavi, "Risk assessment model selection in construction industry," *Expert Syst. Appl.*, no. December, Dec. 2010.
- [50] B. Boehm, "Software Risk Management: Principles and Practices," *Management*, no. January, 1991.
- [51] B. Boehm and R. Ross, "Theory-W Software Project Management: Principles and Examples," vol. 15, no. 7, pp. 902–916, 1989.
- [52] S. Islam, "Software Development Risk Management Model – A Goal Driven Approach," *Development*, pp. 5–8, 2009.
- [53] J. Miler and J. Górski, "Risk-driven Software Process Improvement - a Case Study," in *11th European Software Process Improvement Conference EuroSPI'2004*, 2004, no. 4, pp. 1–7.
- [54] X. Lu and Q. Ma, "Risk analysis in software development project with owners and contractors," *IEEE Int. Eng. Manag. Conf.*, vol. 4, pp. 789–793, 2004.
- [55] T. Wu, J. Blackhurst, and V. Chidambaram, "A model for inbound supply risk analysis," *Comput. Ind.*, vol. 57, pp. 350–365, Jan. 2006.
- [56] J. Hallows, *Information Systems Project Management: How to Deliver Function and Value in Information Technology Projects*, Second. AMACOM, 2005, p. 304.
- [57] C. Dawson, *Projects in Computing and Information Systems A Student's Guide*, First. England: Pearson Education Limited, 2005.
- [58] A. Sharif and S. Basri, "A Study on Risk Assessment for Small and Medium Software Development Projects," *Int. J. New Comput. Archit. their Appl.*, pp. 325–335, 2011.
- [59] J. Bennett, G. Bohoris, E. Aspinwall, and R. Hall, "Risk analysis techniques and their application to software development," *Eur. J. Oper. Res.*, vol. 95, no. 3, pp. 467–475, Dec. 1996.
- [60] Oracle, "More Realistic Estimating: Separating Risks and Opportunities from Uncertainty," 2009.
- [61] M. Holcombe, *Running an Agile Software Development Project*. John Wiley & Sons, Inc., 2008, p. 306.
- [62] S. Du, M. Keil, L. Mathiassen, Y. Shen, and a Tiwana, "Attention-shaping tools, expertise, and perceived control in IT project risk assessment," *Decis. Support Syst.*, vol. 43, no. 1, pp. 269–283, Feb. 2007.
- [63] P. Webern, G. Medina-Oliva, C. Simon, and B. Iung, "Overview on Bayesian networks applications for dependability, risk analysis and maintenance areas," *Eng. Appl. Artif. Intell.*, Jul. 2010.
- [64] B. Berenbach, D. J. Paulish, J. Kazmeier, and A. Rudorfer, *Software & Systems Requirements Engineering: In Practice*. McGraw-Hill Companies., 2009, p. 321.
- [65] J. Deng and Y. Bian, "Constructing a Risk Management Mechanism Model of ERP Project Implementation," in *2008 International Conference on Information Management, Innovation Management and Industrial Engineering*, 2008, pp. 72–77.
- [66] A. Zafra-Cabeza, M. Ridao, and E. Camacho, "An algorithm for optimal scheduling and risk assessment of projects," *Control Eng. Pract.*, vol. 12, no. 10, pp. 1329–1338, Oct. 2004.
- [67] John Horch, *Practical Guide to Software Quality Management*. ARTECH HOUSE, INC., 2003, p. 286.
- [68] S. H. Kan, *Metrics and Models in Software Quality Engineering*, Second. Addison Wesley, 2002, p. 560.
- [69] J. Walewski, "International Project Risk Assessment: Methods, Procedures, and Critical Factors," 2003.
- [70] E. Bennatan, *Catastrophe Disentanglement: Getting Software Projects Back on Track*. Addison Wesley Professional, 2006, p. 288.
- [71] E. W. T. Ngai and F. K. T. Wat, "Fuzzy decision support system for risk analysis in e-commerce development," *Decis. Support Syst.*, vol. 40, no. 2, pp. 235–255, Aug. 2005.
- [72] L. Cooper, "A research agenda to reduce risk in new product development through knowledge management: a practitioner perspective," *J. Eng. Technol. Manag.*, vol. 20, no. 1–2, pp. 117–140, Jun. 2003.
- [73] G. Büyüközkan and D. Ruan, "Choquet integral based aggregation approach to software development risk assessment," *Inf.*



- Sci. (Ny)*, vol. 180, no. 3, pp. 441–451, Feb. 2010.
- [74] D. Houston, G. Mackulak, and J. Collofello, “Stochastic simulation of risk factor potential effects for software development risk management,” *J. Syst. Softw.*, vol. 59, no. 3, pp. 247–257, Dec. 2001.
- [75] K. Khanfar, A. Elzamly, W. Al-Ahmad, E. El-Qawasmeh, K. Alsamara, and S. Abuleil, “Managing Software Project Risks with the Chi-Square Technique,” *Manage. Rev.*, vol. 4, no. 2, pp. 18–29, 2008.
- [76] A. Elzamly and B. Hussin, “Managing Software Project Risks with Proposed Regression Model Techniques and Effect Size Technique,” *Int. Rev. Comput. Softw.*, vol. 6, no. 2 March, pp. 250–263, 2011.
- [77] A. Elzamly and B. Hussin, “Managing Software Project Risks (Implementation Phase) with Proposed Stepwise Regression Analysis Techniques,” *Int. J. Inf. Technol.*, vol. 1, no. 4, 2013.
- [78] A. Elzamly and B. Hussin, “Managing Software Project Risks (Design Phase) with Proposed Fuzzy Regression Analysis Techniques with Fuzzy Concepts.pdf,” *Int. Rev. Comput. Softw.*, vol. 8, no. 11, 2013.