PROJECT RISK MANAGEMENT IN IRANIAN SOFTWARE PROJECTS

UDC: 005.51
005.96
Original Scientific Paper

Ali Reza AFSHARI[1], **Vladimir BRTKA**[2], **Melita ĆOĆKALO-HRONJEC**[3]

[1]Department of Industrial Engineering, Shirvan Branch, Islamic Azad University, Shirvan, Iran
E-mail: afshari@mshdiau.ac.ir
[2]University of Novi Sad, Technical faculty "Mihajlo Pupin" Zrenjanin, 23000 Zrenjanin, Đure Đakovića bb, Republic of Serbia
[3]High school "Laza Kostic", 21000 Novi Sad, Laze Lazarevića 1, Republic of Serbia

Paper received: 25.10.2018.; Paper accepted: 17.12.2018.

Many Software development organizations suffer chronic problems of cost overruns, schedule slips and projects that do not meet the originally specified functions in almost all software projects. This study is commenced by reviewing existing risk management models. Thereafter, a survey that involved software project managers in Iran was conducted to gather empirical data about the practice of risk assessment. Thirty Software Project managers participated and were interviewed. Structured questionnaires were used to capture information. Observation made from the study show that in Iran, risk assessment is poorly practiced and many projects do not yet practice systematic risk assessment. This is because 83% of software project managers implicitly assess risks and the same percentage (83%) of them uses unstructured approaches, poor risk identification and analysis techniques. It additionally shows that risks are not documented and experiences not properly utilized. This has led to recurring problems. The conclusion drawn indicates that Iran's software project managers need to start assessing risks using proper approaches. Review of existing models showed that these models are complex and may not help address the existing shortcomings. This study proposes a risk assessment framework, which helps managers to simply start assessing, documenting major risks, estimating risks using an objective approach that is based on frequently occurring risks to project likelihood of a risk occurring and subjective approach where objective approach is not applicable. It also supports qualitative risk estimation technique using prearranged risk estimation matrices and is supported by a tool that collects and stores risk data for analysis and improvement purposes.

Keywords: Project management, Software projects, Software project risk, Risk assessment, Project risk management.

INTRODUCTION

Most industries, institutions and various activities in the current century are managed using computer applications. However, many Software development organizations suffer chronic problems of cost overruns, schedule slips and projects that do not meet the originally specified functions in almost all software projects. These problems are not experienced only during the development of new and complex projects, but also during the enhancement of existing software projects. Various reasons have been put forward for the prevalence of risks in software projects. These include rapidly advancing computer technology, (Kontio, 2001). There is always constant review of computer systems where new technologies replace the existing ones which are then termed outdated and impractical for new development. These rapid changes have led to increased software development complexity bringing along new risks to the software projects. Also, the intangible nature of software means that traditional processes for managing industrial projects are not effective. It is also noted that of the many engineering disciplines, software engineering is more prone to risks than many other areas.

ISSN 2217-8147 (Online)

Available online at http://www.tfzr.uns.ac.rs/jemc

In software projects, processes and requirements evolve more, complexity of products is higher, and there is a higher number of potential risk factors than in many other disciplines, (Afshari & Kowal, 2018). This is partially due to the inherent nature of software development, in principle, all projects are more unique than those of other disciplines, growing user expectations decrease the stability of a software project, again opening a door for potential risks. This uniqueness makes it more difficult to plan, model and predict the progress of a software project. Assessing risks has a cost. This cost must be balanced against the possible loss which can be incurred if risks are not dealt with, (Wiegers, 2007). This cost includes consumption of much time. However, the benefits thereof are immense, (Selby, 2007). These benefits are - it helps in avoiding disasters, rework and overkill in software projects. It also helps the project managers generate sensible contingency buffers.

This study investigated whether Software Project Managers in Iran assess risks in software projects. It also looked at risk assessment approaches and methodologies used in software projects risk management, means of storing risk data and their reuse and the possible reasons as to why software projects fail to meet performance, cost and schedule requirements. These reasons include major threats in software project that Software Project Managers experience in Iran. The main intention for this research was to obtain facts on the state of the art of risks assessment practice in Iran, develop a risk assessment framework and its supporting tool based on these facts also known as study findings and the existing software project management acceptable practices surveyed in the existing literature.

THEORETICAL BACKGROUND

IT projects have a dismal record of going over budget and schedule, not realizing expectations and for providing poor return on investment, (Jaques, 2004). Attempts at developing information systems are not always successful. Numerous studies have given reasons for causes of project failure and discussed a range of recognized risk factors including those concerning: project leadership and management, organizational culture and structure, inadequate resources, bad planning, negligence in management commitment and patterns of belief, user involvement and training, developer expertise, technology planning, competition in the industry might change rendering the application useless, scope and objectives setting, estimation and choice of methodology, (Bronte-Stewart, 2005). All these factors require careful analysis. The term risk is used universally in different contextual domains. For example, it is used in the financial sector to mean the possibility of incurring financial loss, and in the medical sector to mean the possibility of physiological loss to life. In the "software" world, risk is an important issue often referring to the sources of danger to software development, acquisition, procurement, or maintenance.

Risk assessment is defined as the overall process of risk identification, quantification, evaluation and acceptance. A risk assessment is concerned with measuring the risk of an operation towards the acceptance criteria set for this operation, (Zolotukhin & Gudmestad, 2002). It is apparent that a lot of time and money seems to be spent on software development and billions of dollars are wasted each year on failed projects but, one way or another, a great deal is wasted, (Bronte-Stewart, 2005). A survey for the British Computer Society in 2001 found that only around one in eight projects were successful. For development projects, the figure was even worse with less than 1% succeeding. The survey found that poor management of the requirements and scope of a project were the most common reasons for failure. Risk, communication and the change process were also badly managed, (El Emam & Koru, 2008). Another group of researchers conducted three simultaneous surveys of experienced project managers in three different settings: Hong Kong, Finland and the United States. The three panels of experts identified an initial list of 53 IT project risk factors, (Schmidt, Lyytinen, Keil, & Cule, 2001).

Risk Assessment Models

Several software project risk management approaches have been proposed in the past, most of which assess risks during all the phases of software development, by integrating risk management practices along with the software development process. As a result, in these approaches, the risk management models follow a disciplined process. This study looks at the main existing risk assessment and management approaches and models: Kontio's (2001) Riskit Methodology provides a complete conceptual framework for risk management using a goal, and stakeholder oriented approach by considering their potential utility losses. It attempts to manage risks by capturing the intentions of stakeholders in the risk management process. Riskit complements existing risk management approaches by supporting qualitative and structured analysis of risks through a graphical modelling formalism, (Kontio, 2001). Foo and Muruganathan (2000) proposed a questionnaire based approach for analyzing risks to provide their quantitative assessment. Muruganathan's approach can be used to quantify risk elements, and use them to estimate a normalized value of the overall project risk. Their model, called the Software Risk Assessment Model (SRAM), is based on the use of situational factors to predict project risks. In other words, risk assessment in this model is dependent on the nature of the project, and the situations facing it. In their model, they consider nine critical risk elements: software complexity, project staff, targeted reliability, product requirements, estimation methodology, monitoring methodology, development process adopted, usability of development software, and tools. Thereafter, they frame a list of questions for the risk assessor, by providing three choices for each of the above critical risk elements. The answers of the assessors are assessed, and sorted according to their increasing risk levels, (Misra, Kumar, & Kumar, 2006). Kuipersand Deursen (2003) risk assessment methodology involves identifying the different primary facts, and secondary facts in a project. The primary facts are obtained by analyzing the system, and interviewing different stakeholders, reviewing contract documents, project plans, requirements specifications, and design documents obtain the secondary facts. Finally, the primary facts, and the secondary facts are taken in tandem, and compared to observe whether the risks perceived from both the angles are consistent with each other. This software risk assessment methodology is different from the traditional extremes of product risk management, and process risk management. The advantage of it is that it builds on the advantages of both the above-mentioned extremes to resolve risks having conflicting viewpoints amongst stakeholders. Tiwana and Keil (2004) developed a software development risk assessment tool that project managers could use to quickly assess some of the important project risks, and their effects. This tool and the questions in it were developed because of risk management data collected from the IT managers of 60 companies. The important achievement of this tool is that it can help in quickly assessing the important risks threatening a project, instead of deploying a full-fledged, time, and budget consuming risk management methodology.

Risk Assessment Processes

1. Risk identification is the first step in the software risk management process; risks are identified and added to the list of known risks. Identification surfaces risks before they become problems. The output of this step is a list of project specific risks that have the potential of compromising the project's success. There are many techniques for identifying risks, including interviewing and brainstorming with project personnel, customers, and vendors. Open-ended questions can help identify potential areas of risk, voluntary reporting, where any individual who identifies a risk is encouraged and rewarded for bringing that risk to management's attention. This requires the complete elimination of the "shoot the messenger" syndrome. Decomposition in the form of work breakdown structures during project planning can also help identify areas of uncertainty that may need to be recorded as risks. Risk taxonomies which entail lists of problems that have occurred on other projects and can be used as checklists to help ensure all potential risks have been considered, (Westfall, 2000).
2. Risk classification is based on taxonomy approach. The taxonomy provides a framework for organizing and studying the breadth of software development issues. Hence, it serves as the basis for eliciting and organizing the full breadth of software development risks. Tiwana and Keil (2004) revealed that another way to classify risks is to identify the area of influence. This is because in their study they noticed that project managers can identify risks that fall into their spheres and those that are out of their spheres. Therefore, they classified risks in to two spheres; the project manager's sphere and customer's (user) sphere.
3. Risk analysis is the conversion of risk data into risk decision-making information. Analysis provides the basis for the project manager to work on the "right" and most critical risks. During the risk analysis step, each risk is assessed to determine Likelihood: the probability that the risk will result in a loss and Impact: the size or cost of that loss if the risk turns into a problem. The list of risks is then prioritized based on the results of our risk analysis. Since resource limitations rarely allow

the consideration of all risks, the prioritized list of risks is used to identify risks requiring additional planning and action. Other risks are documented and tracked for possible future consideration. Based on changing conditions, additional information, the identification of new risks, or the closure of existing risks, the list of risks requiring additional planning and action may require periodic updates, (Westfall, 2000).

4. Risk planning turns risk information into decisions and actions. Planning involves developing actions to address individual risks, prioritizing risk actions, and creating an integrated risk management plan. The plan for a specific risk can take many forms.
5. Risk tracking consists of monitoring the status of risks and the actions taken to ameliorate them. Appropriate risk metrics are identified and monitored to enable the evaluation of the status of risks as well as of risk mitigation plans. Tracking serves as the “watchdog” function of management. The results of the tracking can be identification of new risks that need to be added to the risk list, Validation of known risk resolutions so risks can be removed from the risk list because they are no longer a threat to project success, Information that dictates additional planning requirements, Implementation of contingency plan. Many of the software metrics typically used to manage software projects can also be used to track risks. For example, Gantt charts, earned value measures, and budget and resource metrics can help identify and track risks involving variances between plans and actual performance. Requirements churn, defect identification rates, and defect backlogs can be used to track rework risks, risks to the quality of the delivered product, and even schedule risks, (Westfall, 2000).
6. Risk control corrects deviations from planned risk actions. Once risk metrics and triggering events have been chosen, there is nothing unique about risk control. Risk control melds into project management and relies on project management processes to control risk action plans, corrects for variations from plans, responds to triggering events, and improves risk management processes.
7. Without effective communication, no risk management approach can be viable. While communication facilitates interaction among the elements of the model, there are higher level communications to consider as well. In order to be analyzed and managed correctly, risks must be communicated to and between the appropriate organizational levels. This includes levels within the development project and organization, within the customer organization, and most especially, across that threshold between the developer, the customer, and, where different, the user. Because communication is pervasive, our approach is to address it as integral to every risk management activity and not as something performed outside of, or as a supplement to, other activities.

METHODOLOGY

The commencement point for this study was to understand risk assessment and management concepts and acceptable practices found in the existing literature. The surveyed literature covered project risk assessment and management. This literature was meant to aid in the understanding of available contributions, their contexts, general current state of risk assessment in software development industry, and recommendations that have been put forward for further improvements. Then, survey that involved interviewing of software project managers in Iran was conducted using structured questionnaires to collect data. The questionnaire enabled to acquire information that could only be used to achieve the objectives of the study. The data helped to understand the kind of framework and tool that is suitable to software projects and solves existing problems in Iran. The questionnaire included twenty questions that were closed and open-ended. Almost all the closed questions comprised a number of remarks. It also contained ten brief sections which include: practice of risk assessment, methods used in software risk assessment, responsibility for risk assessment, techniques for identifying and analyzing risks, risk prioritization, involvement of stakeholders, documentation and use of risk information, perceptions on use of risk assessment tool and approach, feature for risk assessment framework and tool and problems faced by software project managers in software projects

Data collection

Determining the population presented some difficulties. This is because the study involved surveying diverse industries of software development firms and of varying sizes. The inclusion criteria were software project managers from organizations, which developed software and willing to participate in the study. The exclusion criteria were those software project managers who procured commercial off the shelf software, organizations that were selected but had relocated to other regions and those who were not willing to participate in the study. Another problem was also encountered when the sample size was being determined. This is because each of the selected industries had many organizations. Therefore, with the knowledge that this study was only meant to obtain facts or requirements to aid in the development of a risk assessment framework and tool and did not require any comparisons of variables or need power analysis to determine the sample size; organizations presumed to undertake software development from the directory were selected. The population included 29 Parastatal organizations, 9 Universities, 13 Software developers, 5 Telecommunication operators and a list of 8 Computer software and services organization. After dividing the number of companies in each industry by two to obtain a meanings sample, 15 Parastatals, 6 Universities, a whole list of 13 Software developers was considered to increase the number of software development firms for this study, and 16 computer software and service firms were selected from a list of eight columns where two firms from each column were selected. This yielded a sample size of 50.

The selected organizations were visited and one participant in each organization who was either a Software Project Manager or an IT manager was requested to participate. Participants who were willing and ready for the interview were interviewed immediately while willing but busy participants scheduled a day and time for the interview with the interviewer. Among the selected organizations, eight of these organizations did not meet the requirements for this study because they purchase commercial off-the shelf software. Nine organizations declined to participate. Three organizations were relocated. These reduced the sample size to thirty organizations. The data which was collected from questionnaires was analyzed using Microsoft excel and presented using both graphical and text format. As in all research methods, there are some uncertainties and risks involved. In this study, quite a few uncertainties and risks were identified by researcher according to this study experiences and are presented here to make the reader aware and consider them while reading this work. Each uncertainty is also connected to an explanation of how the impact and occurrence thereof was lowered. The first problem involved scarcity of review materials in the area of risks assessment and management. Choosing a sample to interview for this study presented two major problems. One, the potential population included all companies involved at any level that are small, medium and large companies with software development. It was unrealistic to survey the entire universe, so a representative sample was necessary. Therefore, to meet the objective for this study, purposive sampling technique was used. Secondly, it was difficult to determine which organizations actually produced software. It was difficult to base this on the size or location of an organization that is, an organization produced software rather than purchased commercial off-the-shelf software just because it was located in a certain area and had a given size. The sample size may look small for the case study but the researcher totally deems this sample size suitable because there were no comparisons of variables in the study. The study was trying to obtain facts about the practice of risk assessment from Iran’s software project managers or IT managers. Therefore, based on the results obtained from this study there is no apparent reason not to believe that the developed framework and tool is suitable for software development community in Iran. Another difficulty was fear. Interviewees had fear releasing their organization’s information and they had fear discussing about drawbacks that could have led to the interviewees amending the truth. In the first fear, the interviewees were assured of the safety of their information both verbally and in writing as indicated on the cover letter. In the second fear, the researcher was flexible enough and the interviewees were left to decide the location and the time for the interview. The researcher believes that this flexibility had a positive impact on the study and might have reduced this influencing factor. Most of all, the researcher provided an environment of trust and therefore confidence to believe that the interviewees were honest and trustworthy was established. The interviewees were either IT managers or Software Project Managers and were all rather busy. This could have led to answers not being as comprehensive as desired. The interviewee was able to choose time and place and the questions were presented to them in advance. In some cases where the IT manger or Project Manager was too busy, a senior software developer who had the desired experience was requested to participate.

FINDINGS

This section discusses the results obtained from the study that was conducted in Iran on risk assessment. The results were obtained by interviewing software project managers or IT managers from different companies and institutions. The interviewees were asked whether they assess risk in software projects in their organizations.

Results show that most organizations do not accord much attention to risk assessment since a greater proportion (83%) implicitly assessed risks. Only 17% assessed risks, meaning they have an established structure for risk assessment. Those that implicitly assess risks, argue that risk assessment consumes much of the project time and only manage projects and deal with risks as they occur. None responded that they do not assess risk because all believe that risks in software projects are inevitable in the software project development process. Methods used in software project risk assessment included structured or systematic approach (7%), semi-structured approach (10%) and ad hoc approach (83%). project manager and customer in software risk assessment activity. Results show that 13% used brainstorming, framework, questionnaires and checklists in identifying risks while 3% had a hotline for risk assessment and monitoring. In addition, 83% of the respondents did not have any technique for risk identification because their risk handling strategy was reactive. Results show that 40% of the respondents prioritized risks, majority (60%) did not prioritize risks and they dealt with risks as they appeared. That employing structured risk assessment approach used decision support tools and risk matrix in prioritizing risks while those using the ad hoc approach consider urgency, giving technical and development risks more priority as well as brainstorming with project team members in prioritizing risks.

It was clearly demonstrated from the study that most software project managers did not involve stakeholders especially end users in software risk assessment. 17% of the interviewees who assessed risks involved project managers, project sponsors, developers, sales team, risk auditors and end users in assessment. Majority (90%) involved stakeholders only at some stages of the project. This normally happens after the impacts of a risk have been experienced. While only 10% involved stakeholders from the project inception to the end. While 17% of software project managers and IT managers documented risk data, majority of them (83%) did not. Mode of storage of risk information included database (7%) and paper (10%). Regarding the use of past risk experiences, it was established that only 13% of the respondents stated that they used past risk experiences very often and another 43% used past risk experience often. 43% said they did not use past risk experience stating that projects were not similar, and differed in complexity and stakeholders needs. Table 1 shows that the problems include major risks faced by software project manager in Iran.

Table 1 Problems faced by software project managers in Iran

Over reliance on few skilled developers	Delays in licenses
Customer delays to give report samples	Incomplete requirements
Requirements not clear	Users change requirements
High staff turnover	Frequent loss of data
Faulty equipment (network)	Late user involvement
Undefined user role	Lack of user commitment
Unrealistic schedule estimation	Delays in procurements
Over spending of allocated budget	Project team recalled in their stations
Complex projects	New technology
Lack of formal project management practices	Technical staff lack appropriate skills

Changes in user requirements, often because of limited customer or user experience in development of such projects, unclear objectives and management changes, led to delays in projects, budget overrun and unhappy developers because of the wasted effort.

Schedule overrun was reported as a risk to software projects though it is not in itself a risk, but an outcome of undesirable events in the project. Therefore, it was clearly noted from the study that most software projects fail to meet schedule objective. This was mainly due to unrealistic schedule estimation, delays in acceptance testing, delays in procurements, change in requirements and unskilled developer. While lack of formal project management practices such as assessment of risks and failure to follow the project plan often led to lack of project control hence schedule and budget overrun, late user or customer involvement led to systems being rejected, non-performing products and unsatisfied user.

Complex projects often led to a wider learning curve hence delays in projects and poor design thereby production of underperforming systems. Projects become complex when the tasks to be automated demand much effort especially when new or leading edge technologies are introduced.

Organizations relied on few skilled staff. This led to schedule overrun as few staff left to work on the entire project. Coupled with lack of the desired infrastructure in the deployment environment or delays in procurements, this led to unhappy project team, unsatisfied user and delays in the project. Undefined user roles in the project limit user involvement leading to systems being rejected by users.

While lack of experience in the programming language for developing the software led to underperforming system and schedule overruns, lack of commitment by users to the project led to poor communication and delays in acceptance testing. Finally, incomplete and unclear requirements because of lack of user involvement in projects and customer understanding of the process that was to be automated were risks in software projects.

CONCLUSION

Most projects lacked formal risk assessment approaches and this is the major issue in software projects in Iran. This study reveals that, risk assessment in software projects is poorly practiced. This is because 83% of software project managers implicitly assess risks. At the same time, 83% of them use poor risk assessment approaches, which is also unstructured. This is mainly due to lack of awareness of formal risk assessment practices and ignorance. Even though some software project managers are aware of risk assessment good practices, they do not want to implement them because their attitude to risk assessment is very poor. They believe the process consumes much project's valuable time. It was also noted that Software development organizations that are ISO certified assess risks in projects. This is because ISO frameworks demands for policies and development of organizations' culture to support risk assessment and management. This study mentions two issues that were perceived to be major factors contributing to lack of formal risk assessment approaches in Iran but does not discuss them because they are not main objectives of this study. These include poor attitude on software projects risk assessment and organizations' policies and culture that do not support the risk assessment activity. Coupled with implicit and unstructured risk assessment approaches lack of risk prioritisation, effective use of lessons learned and proper allocation of resources for mitigation, late or lack user participation due to poor risk assessment strategy are used. The term late is used to explain that the project has reached an advanced stage where reversal of certain functions may affect the project greatly. These explain why there is a chronic problem of schedule and budget overrun in software projects. Also majority (93%) of Iran's software project managers have never used risk assessment tool. 80% of the software project managers expressed the need for the development of this risk assessment framework and its tool to help them quickly and easily assess risks. The conclusion drawn from the result of risk assessment study carried out in Iran is that the existing risk assessment approaches tend to be inexplicit, unstructured, and undocumented. In most organizations, there is no communication of risks in software projects. The tools are complex as this was expressed by those who use DSS risk management support work tools. Therefore, software project managers need to start assessing risks. This study proposes a risk assessment framework that supports systematic, proactive, explicit, documented and continuous risk assessment process using experience based on frequently occurring risks. It also proposes a tool that supports the framework. The tool aids in the documentation of risk information by capturing identified risk information, fast identification of risks based on a list of generic risks stored in the domain and estimation of risk likelihoods based on frequently occurring risks.

REFERENCES

Afshari, A., & Kowal, J. (2018). Decision making methods for the selection of ICT project manager. *Gospodarka Rynek Edukacja - Economy Market Education, 18*(4), 19-28. doi: 10.2139/ssrn.3118075

Bronte-Stewart, M. (2005). Developing a risk estimation model from IT project failure research. *Computing and Information Systems, 9*(3), 8.

El Emam, K., & Koru, A. G. (2008). A replicated survey of IT software project failures. *IEEE software*(5), 84-90. doi: 10.1109/MS.2008.107

Foo, S.-W., & Muruganantham, A. (2000). *Software risk assessment model*. Paper presented at the Management of Innovation and Technology, 2000. ICMIT 2000. Proceedings of the 2000 IEEE International Conference on Management of Innovation and Technology, Singapore.

Jaques, R. (2004). UK wasting 'billions' on IT projects: damning report from the Royal Academy of Engineering and the British Computer Society. *Computing IT, vnunet. com, April, 22*.

Kontio, J. (2001). *Software engineering risk management: a method, improvement framework, and empirical evaluation*. Helsinki: University of Technology.

Kuipers, T., & Deursen, A. V.(2003). *Source-based software risk assessment*. Paper presented at the International Conference on Software Maintenance, Amsterdam, The Netherlands.

Misra, S. C., Kumar, V., & Kumar, U. (2006). *Different techniques for risk management in software engineering: A review*. Paper presented at the Conference of the Administrative Sciences Association of Canada (ASAC), Banff, Alberta.

Schmidt, R., Lyytinen, K., Keil, M., & Cule, P. (2001). Identifying software project risks: An international Delphi study. *Journal of Management Information Systems, 17*(4), 5-36. doi: 10.1080/07421222.2001.11045662

Selby, R. W. (2007). *Software Engineering: Barry W. Boehm's Lifetime Contributions to Software Development, Managment and Research*. Hoboken, New Jersey: Wiley Press.

Tiwana, A., & Keil, M. (2004). The one-minute risk assessment tool. *Communications of the ACM - Bioinformatics, 47*(11), 73-77. doi: 10.1145/1029496.1029497

Westfall, L. (2000). *Software risk management*. Paper presented at the Annual Quality Congress Proceedings-American Society for Quality Control, Milwaukee, Wisconsin.

Wiegers, K. (2007). *Practical project initiation: a handbook with tools*. Microsoft Press.

Zolotukhin, A. B., & Gudmestad, O. T. (2002). Application of fuzzy sets theory in qualitative and quantitative risk assessment. *International Journal of Offshore and Polar Engineering, 12*(4).

UPRAVLJANJE PROJEKTOM RIZICIMA U IRANSKIM SOFTVERSKIM PROJEKTIMA

Mnoge organizacije za razvoj softvera pate od hroničnih problema prekoračenja troškova, rasporeda i projekata koji ne ispunjavaju izvorno specificirane funkcije u gotovo svim softverskim projektima. Ovo istraživanje započeto je pregledom postojećih modela upravljanja rizicima. Nakon toga, sprovedena je anketa koja je uključivala menadžere softverskih projekata u Iranu, kako bi prikupila empirijske podatke o praksi procene rizika. Trideset menadžera softverskih projekta učestvovalo je u istraživanju. Strukturirani upitnici korišćeni su za prikupljanje informacija. Rezultati koji proizilaze iz studije pokazuju da je u Iranu slaba procena rizika, odnosno da se kod mnogih projekata još uvijek ne sprovodi sistematska procenu rizika. To je zato što 83% menadžera softverskih projekata implicitno procjenjuje rizike, a isti procenat (83%) koristi nestrukturiran pristupo, lošu identifikaciju rizika i tehnike analize. To dodatno pokazuje da rizici nisu dokumentovani i iskustva nisu pravilno iskorišćena, što dovodi do problema koji se ponavljaju. Izvedeni zaključak ukazuje da menadžeri softverskih projekata u Iranu moraju početi procenjivati rizike koristeći odgovarajuće pristupe. Pregled postojećih modela ukazuje je da su ti modeli složeni i da ne mogu pomoći u rešavanju postojećih nedostataka. Ova studija predlaže okvir za procenu rizika, koji pomaže menadžerima da jednostavno počnu sagledavati i dokumentovati glavne rizike, procenjivati rizike pomoću objektivnog pristupa, koji se temelji na često prisutnim rizicima i verovatnoći nastanka rizika, kao i subjektivnom pristupu u situacijama kada objektivni pristup nije primjenjiv. Takođe podržava kvalitativne metode procene rizika pomoću unaprijed utvrđenih matrica procene rizika i podržan je alatom kojim se prikupljaju i čuvaju podaci o rizicima za svrhu analize i poboljšanja.

Ključne reči: Management projektima, Softverski projekti, Rizici softverskih projekata, Analiza rizika, Rizik projekta.