Journal of Optimization in Industrial Engineering
Vol.13, Issue 2, Summer & Autumn 2020, 267- 278
DOI: 10.22094/JOIE.2020.1883892.1700

Research Article

Developing a Risk Management Model for Banking Software Development Projects Based on Fuzzy Inference System

Tooraj Karimi [a,*] , Mohammad Reza Fathi[a], Yalda Yahyazade[a]

[a] *Faculty of Management and accounting, University of Tehran college of Farabi, Iran, Tehran*

Received 30 November 2019; Revised 30 June 2020; Accepted 30 June 2020

Abstract

Risk management is one of the most influential parts of project management that has a major impact on the success or failure of projects. Due to the increasing use of information technology (IT) systems in all fields and the high failure rate of IT projects in software development and production, it is essential to effectively manage these projects is essential. Therefore, this study is aimed to design a risk management model that seeks to manage the risk of software development projects based on the key criteria of project time, cost, quality and scope. This is presented after making an extensive review of the literature and asking questions from experts in the field. In this regard, after identifying the risks and defining them based on the dimensions and indicators of software development projects, 22 features were identified to evaluate banking software projects. The data were collected for three consecutive years in the country's largest software development eco-system. According to Rough modelling, the most important variables affecting the cost, time, quality and scope of projects were identified and the amount of risk that a project may have in each of these dimensions was shown. Since traditional scales cannot provide the accurate estimation of project risk assessment under uncertainty, the indexes were fuzzy. Finally, the fuzzy expert system was designed by MATLAB software that showed the total risk of each project. To create a graphical user interface, the MATLAB software GUIDE was used. The system can predict the risks of each project before each project begins and helps project managers be prepared to deal with these risks and consider ways to prevent the project from failing. The results showed that quality and time risks were more important than cost and scope risks and had a greater impact on total project deviation.

Keywords: Project Risk Management; Software Development; Expert Systems; Rough Set Theory; Fuzzy Logi; Fuzzy Inference System.

1.Introduction

In today's ever-changing and complex world, projects and processes face immense risks that could cause major disruption in their progress, if not causing the entire project's failure, unless proper precautions and measures are taken into account in order to react to these risks. Risk management is a necessary measure for achieving goals in projects; hence, it is essential to pay special and precise attention to risk management (Akbarpour & Seyedesfahani 2010). Based on the project management body of knowledge, the risk is defined as any uncertain occurrence that would impact at least one of a project's goals (PMI 2013). The software development process is prone to many risks, which is evident from the high rates of failure in such projects. Two of the major goals of a software development project is profit and meeting the deadlines. The presence of any risk in the project would result in additional costs and delay in its progress. However, a project's success is not only related to time or cost, factors such as quality, performance, customers' satisfaction and many others are also important indicators for a project's success. A vast majority of previous models have been focused on costs and very few have considered quality and time, which results in the point that a small number of these models could apply to large projects (Zhang & Fan 2014). In this project, a model is designed that considers factors such as time, quality and scope of a project in addition to cost in risk assessments for software development projects. One of the challenges faced by the experts and supervisors of software development industry is lack of an intelligent predictive system for projects risk assessment. Among different systematic and analytic approaches, expert system (ES) has been acknowledged as an effective knowledge-based technique with a variety of applications in industry and services, e.g. failure prediction, performance evaluation and classification, disorder explanation, accident and fault diagnosis, process control and risk assessment (Ford, 1985). ES can play different roles, depending on the development extent of the knowledge base and the technology. Since ES is dependent on deduction, it must be possible to explain its reasoning to the solution in order to examine how it is argued. Given different uncertainties in software development projects, it is necessary to design a fuzzy model based on the fuzzy sets theory (Dokas et al., 2009). The fuzzy inference system provides a schematic process for converting a knowledge base into a nonlinear mapping. For this reason, knowledge-based systems (fuzzy systems) are used in engineering applications and decision making. The fuzzy inference system is a mapping of the input-output space that is implemented using membership functions and fuzzy rules (Nourian et al., 2019). Fuzzy sets and fuzzy logic are powerful mathematical tools for modeling uncertain industrial, human and natural systems. Fuzzy models facilitate decision making by means of approximate reasoning and linguistic terms. They play an important role when applied to the complex phenomena which are not easily

*Corresponding author Email address: Tkarimi@ut.ac.ir

described by traditional mathematics. In fact, this approach helps decision makers extract and keep expert knowledge in the system. It should be noted that the aforementioned ability is regarded as the main advantage of the proposed approach compared to studies, in which fuzzy logic is integrated with MCDM methods. In addition, the proposed approach provides an opportunity for experts to choose different operators such as t-norms, s-norms and defuzzification operators which, in turn, brings flexibility to the decision making process (Pourjavad & Shahin, 2018). The fuzzy ES is one of the branches of artificial intelligence that works as an expert with a wide range of specialized knowledge to solve problems with data uncertainty, e.g. data analysis, prediction, linear and nonlinear control, operations research, pattern recognition and financial models (Dokas et al., 2009). By extracting expert knowledge, a fuzzy ES outlines the interactions between the input and output variables affecting the system through a fuzzy rule set.

To establish a risk assessment system, this research presents a fuzzy ES model to find an optimal model for a software development project risk management (SDPRM) with a focus on banking software. In this regard, after careful considerations for identifying and defining risks based on the project's scope and indicators, a fuzzy ES model is designed to manage risks in software development projects. Due to the large volume and variety in the model's input data based on multiple projects, fuzzy logic and functions are employed in the expert system. The expert system recommended in this paper is therefore based on fuzzy logic and fuzzy inference engine. Thus far, there has been no similar research on designing a fuzzy expert system for risk assessment in banking software development in Iran. This research proves useful to all firms working in the field of banking software development in Iran that seek to minimize risks and provide better management on their solutions.

2. Literature Review

2.1. Theoretical Literature Review

Risk management is a subsection of project management that aims to help project managers assess and better respond to risks. The main objective of risk management is to maximize the probability of success for the project which is achieved through identifying and systematically assessing the risks as well as finding solutions to avoid or remedy risks and maximize opportunities (Chapman & Ward 2011).

The project risk management process consists of two levels: assessment and response to risks. Assessing risks is broken down into two parts; identifying and analyzing risks. Many techniques have been utilized to identify risks, each of which has their unique scenario (Lee et al. 2009). At this level, the sources for risks that are halting a firm's progress towards its goals are classified and their causes are recorded based on their impacts. This is a qualitative process which aims to identify and explain risks that impact a project's goals. Identifying risks is not a fixed process and must be carried out regularly in all stages of the project. Risk identification is an iterative process. At first, the process is carried out by people in the project team or a risk management team and, later, iterations are made by the entire project team and subordinates (Vasudeva & Urvashi 2017). Main tools in risk identification are brainstorming, reviewing records, Delphi technique, checklist analysis and assumptions analysis. Risk analysis is either qualitative or quantitative. Qualitative risk analysis consists of probability assessment, impact and probability-impact matrix, risk classification, Delphi technique, brainstorming, hypothesis analysis, checklist analysis and expert's decision. On the other hand, quantitative risk analysis consists of sensitivity analysis, expected monetary value returned, decision trees based on utility theory, simulation, causal diagrams, penetration graphs, game theory, fuzzy theory, Rough analysis and analysis of error trees (Lee et al., 2009). Software development project risk management is a set of steps taken to identify and eliminate areas that are either directly causing a risk to the production and development of software, or might end up becoming a risk later on (Boehm 1989).

Project risks are uncertainties that could result in positive or negative effects on one of a project's goals such as scheduling, scope or quality (PMI 2013). Based on the standards of project management, the goals of a project could be different elements such as "scope of the project", "total time of the project", "total cost of the project" and "product quality" (Mohammadipour & Sadjadi 2016). In this research, the negative effects of risks are taken into consideration. In other words, risk means deviation from the goal. The fuzzy inference system consists of the following parts:

1. The fuzzification part which turns an integer-valued variable into a fuzzy set; During this step, membership functions are considered for each input variable in order to transform them from deterministic into fuzzy and enter the fuzzy inference system. Membership functions have multiple forms such as triangular, trapezoid, Gaussian, etc. In this research, the triangular form is studied.
2. Fuzzy rules base, which is a set of if-then instructions; There exist two main methods to determine these rules: using expert knowledge and self-organizing learning such as heuristic algorithms and neural networks. In this research, expert knowledge is used to determine the number of antecedents and outcomes and, eventually, generate fuzzy rules. In this expert system's regulations base, each rule is an "if-then" structure.
3. The fuzzy inference engine which is responsible for turning inputs into outputs through processes.

Using Fuzzy Operators: When the number of antecedents is larger than one, fuzzy operators are utilized to calculate a number which is representative of the antecedents' influence on that rule; this number is used in the output function. This number is called the "corrective number" of a rule. One of the most important relations in this section is the fuzzy inference methods of Mamdani and Larson, which use min and multiplication operators, respectively, to calculate the corrective number. In this paper, the Mamdani method is employed to calculate the corrective number.

Using Implication method: This means using the corrective number in the output function, such that the input is the corrective number and the output is a fuzzy set.

Union of Outputs: Since decision making in a fuzzy inference system is based on all rules and regulations, these rules must be combined. The union is a method that would turn all outputs into a single fuzzy set. The two most important union methods are maximization and summation, the latter of which is used in this paper.

De-fuzzification: During this step, a fuzzy output is transformed into a deterministic number. Some of the methods for de-fuzzification are centroid of area (COA), bisector of area (BOA), smallest of maximum (SOM), largest of maximum (LOM), mean of maximum (MOM), weighted average (WA) and weighted sum (WS) (Sumathi & Paneerselvam 2010). In this paper, the COA method is used. Figure 2 depicts the framework of a fuzzy inference system (Foong et al., 2009).

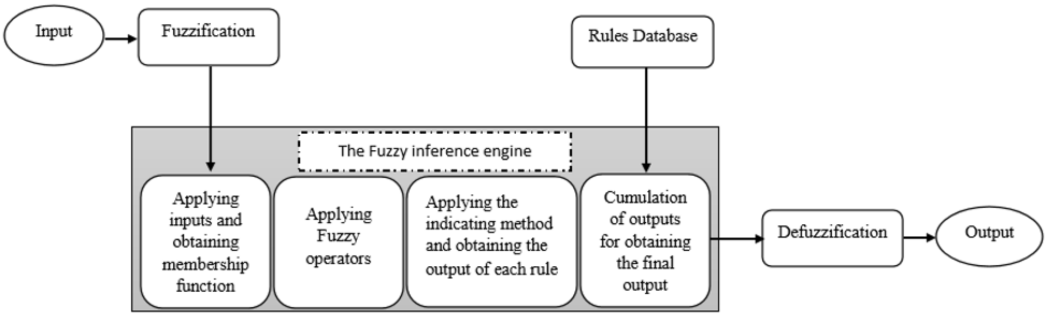

Fig 1. The fuzzy inference system

2.2.Experimental literature review

Ropponen and Boonjing (2000) took into account the risks surrounding the software business from different points of views to study the positive effects of risk management on software development. They identified 6 types of risks associated with software development through a survey on 80 project managers: Scheduling, system operations, subsidiary contracts, customers' demands, operational risks, resources usage risks and human recourses management. They studied how risk management could control these risks and which environmental factors such as methods of development for software or experience of managers could impact risks.

Pimchangthong and Boonjing (2017) expanded those previous methods for risk management that had impacted information technology (IT) projects successfully. For this study, data were gathered from 200 project managers as well as managers and IT technicians of successful IT firms and analyzed using methods such as t-test, one-way ANOVA and linear variable analytics in a sample with 0.05 error. Results showed differences in firms and that their sizes affected their success in IT projects.

Song and Jiang (2016) identified and analyzed 10 of the most frequent and impactful risks associated with software development such as End customer risks, hardware limitations such as low quality or quantity or availability, sufficient test scenarios, no knowledge of necessities, insufficient test times, unrealistic scheduling, shortage in manpower as well as unfamiliarity of developers with conditions and new hardware. Chen et al. (2014) thought that many risk factors directly influence software development and risk management and finding risk factors are major parts of the software development process. Some of the risks influencing software development in their study are: corporate environment, user-related risks, obligatory processes' risks, project complexity related risks, project control risks and inefficient team. Sonchan and Ramingwong (2014) reviewed previous literature and used the Delphi technique for analyzing twenty of the highest risk factors in software projects. Some of the risks were obligatory process (not knowing customer demand, inability to satisfy demand), one's test (not having the technical competence), test and combinations (low performance of software), development process (inefficient team, unsuitable development process), process system (problems with new technologies, insufficient infrastructure), management (unrealistic planning, optimistic resource planning, no executive involvement), work environment (communication gap, disconnect between team members), resources (financial situation of employees, unrealistic budget, scarcity of resources) and program environment (user resistance, rule-breaking).

Arnuphaptrairong (2011), in their literature review, studied risks of software development projects from Paré et al. (2008), Han and Huang (2007), Wallace and Keil (2004), Addison (2003), Schmidt et al. (2001), and Boehm (1991) and, then, introduced them in 6 dimensions: users (14), obligations (17), complexity (4), planning and control (27), team member (9), and company environment (9). This showed that the most risk-prone areas were planning and project management, not fully understanding obligations and disconnection from users of the software. Chatterjee et al. (2019) studied software risks in its first stages to produce more reliable software because the necessary measures for reliability optimization with regard to time and cost have been set by developers. In this study, algorithms were used based on rules to produce the fuzzy rules for predicting the initial risks of the software's life cycle. The model uses fuzzy logic to combat uncertainty and 26 software projects' data were used to analyze the performance and accuracy of this model.

Aqlan and Lam (2015) provided a guideline for supply chain risk assessment which included three parts: estimates, bow-tie analysis, and fuzzy inference systems. Some risks were then identified for supply chains. Bow-tie analysis was a diagram which showed the relationship between the causes of risks and effects of them and, eventually, provided the total probability of risks and impact. Finally, with the use of fuzzy inference systems and taking into account management and risk prediction parameters, total risk was calculated.

The problems of data non-availability or uncertainty are prevalent in most of the modelling and decision making areas of engineering. Jamshidi et al. (2013) used fuzzy logic to model uncertainty existing in the risk evaluation problem in a pipeline. Thus, they applied one of the most popular pipelines the causes of risks and their correlations. This model depicts the relative danger criterion as a quantitative value which is the estimation of the cumulative probable dangers in a flight. In this flight risk assessment model, hiring processes are systematic, experiences and knowledge are gathered in an expert system, and assessment is autonomous. Mohagheghi et al. proposed a project cash flow assessment technique, according to project scheduling in different stages of project. Interval-valued fuzzy sets (IVFSs) were taken into account to cope with the uncertainty of activity durations and cost. Due to the advantages of fuzzy models, different types of them have been used in many fields (Ghasemi et al., 2015; Faezy Razy, 2015; Hossain et al., 2016; Makui et al., 2016). Muriana and Vizzini (2017) proposed a model based on quality in addition to time and cost in the project management triangle that was quantitative and would compare the real amount worked with the planned amount, calculated the deviation of project's performance and took necessary precautions to prevent disasters. In this research, the entire project was broken down into smaller parts time-wise by focusing on work progress state (WPS).

As mentioned previously, there are many studies in the area of risk management and risk assessment; but, there are fewer studies in the area of risk assessment of software development projects, especially in banking industry. Nourian et al. (2019) proposed a hybrid fuzzy decision support system to decrease the risk in gas industries connected to gas transmission services. The advanced fuzzy expert system was programmed by C and CLIPS, and was joined with MATLAB for calling fuzzy membership functions. There are few number of studies as a case study in risk assessment techniques combined with fuzzy logic and proposed a model which was built using Mamdani algorithm and MATLAB's fuzzy logic toolbox. This could be implemented in many engineering problems as a smart risk assessment tool. Hadjimichael (2009) proposed a fuzzy expert system to assess flight procedures risk which depicted

the area of software development risk assessment using a fuzzy approach. Mousavi et al. (2011) extended fuzzy decision making methodology for risk response planning in large-scaled projects. In the current study, a case study was provided, for the first time, in banking software development company. An expert system was proposed that took into account the cost, time, quality and scope of a project for risk assessment in project management in addition to studying risks and criteria for the software development process. The most critical contributions of the present study are summarized as follows: First, this research is the first study that develops a fuzzy rule-based ES for software development projects. Second, ES is implemented in a fuzzy environment as a decision support system for risk assessment and could assist experts and managers in the banking industry to mitigate the potential risks of failure. The proposed model takes into account a wide range of project dimensions (cost, time, quality and scope) and the amount of risk that a project may have in each of these dimensions is shown. Most of the research considers only cost and time criteria. Given the input from the user and using the knowledge base, more than 600 rules are extracted to effectively control the whole system's performance based on fuzzy input data.

3. Modelling

Figure 2 shows the framework of the expert system.

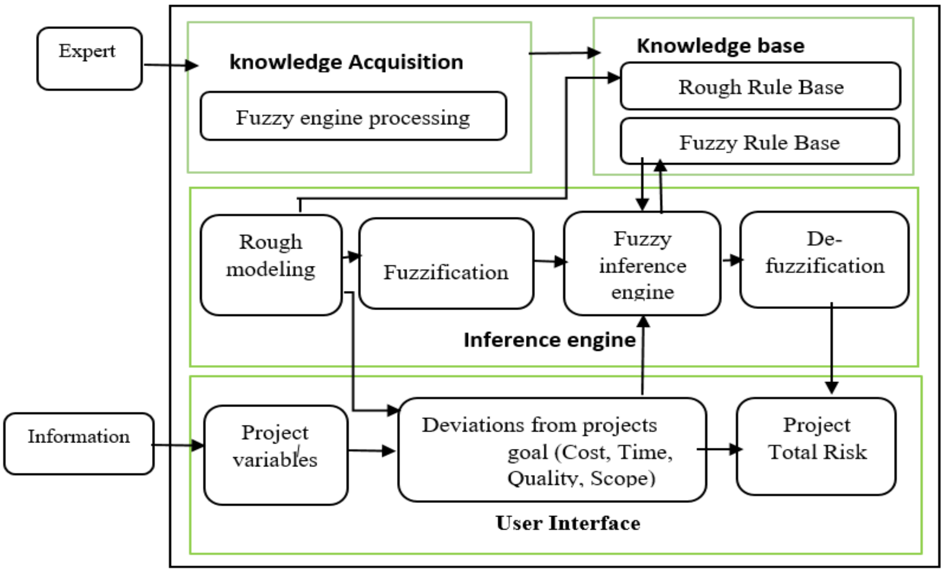

Fig 2. Framework of fuzzy expert system

4. Methodology

This research's results could apply to real-world problems by software development and production firms. This research is qualitative-quantitative and both qualitative and quantitative measures are employed to gather and analyze the data. The goals of this paper are to find and study relations. Data gathering methods used in this paper are library study methods, interviews, evaluating real-world documents and data. In this paper, combined research methodologies are employed.

Library and field studies were used to gather the data. Library methods were employed to study the theoretical and literature review and identify the general risk-prone areas of software development projects. To identify other criteria and probable areas and to localize them for banking software, questionnaire methods and deep interviews were used. Also, real data in banking software development firms' databases were utilized for modeling.

This paper's main study population was broken into two major groups: the first group consisted of highly exceptional university professors and the second group included IT managers, risk managers and other similar positions in one of the country's biggest information technology ecosystems. In total, 32 experts were available and willing to participate that had two major features (firstly, deep knowledge of the area and, secondly, more than 5 years of experience) and were chosen based on a combination of purposeful non-probabilistic sampling (judicial) and snow-ball sampling. Eventually, 30 experts participated completely. The sample of this study consisted of 629 finished projects within the last three years in the biggest IT ecosystem in Iran.

5. Research Results

5.1 Identified risks and variables in this research

A hundred and twenty-nine risks relating to software development projects were identified following a thorough study of the literature. Risks were then evaluated by experts in the field of banking software development and, using combinations and subtraction of redundant risks, 45 risks were chosen for further inspection in this paper. Twenty-two criteria were chosen for risk assessment with the help of experts and based on the literature review. These criteria could help regarding developing the banking software and, eventually, reaching the final goals of the projects. According to the experts, these criteria were all-inclusive for the task of assessing and studying risks of banking software development projects. Table 1 depicts all the risks and criteria as well as their calculation methods.

Table 1
Software development projects' risks and criteria

NO.	Risk factor	Project's Criteria	Explanation	Methodology
1	Unrealistic Budgeting risk	Budget	Budget is the financial plan and index of all planned costs and revenues for a define period; its wrong estimation or lack of sufficient budgets could result in project's risks.	Project budget is defined based on strategic annual planning
2	Financial Resources risk			
3	Unrealistic cost estimation	Cost	Total cost of the project. The most important factor in cost estimation for a software development project is the cost of man power involved.	Costs of project's variables are calculated based on person/hour
4	Market Risk	Revenue and ability to sell product in marketplace	Competitiveness and revenue generated or expected from sale of the project.	Each project is given a score based on factors such as customer's value, sales skills, profitability and priority of the project based on analysis, production, software support and customer service.
5	Risk associated with profit and returns			
6	Risk of pricing and wrong estimations	Cost deviation estimates	Wrong estimation for costs of a project from the beginning would increase cost-related risks	This is calculated by comparing planned costs with actual project costs
7	Risk of undetermined customer needs (undetermined obligations)	clearness of customer requirements in SRS	An obligation's feature is essentially an official document about the need of customers in software and its developers.	When defining a project, its SRS needs to be produced as well. A changing system, not knowing system's obligations and unclear obligations would result in unclear SRS.
8	Risk of misunderstanding obligations			
9	Risk of high complexity of project	Product's technical complexity	IEEE has defined software's complexity as the hardship when it comes to understanding a system and the item that is designed and implemented.	When announcing customers' requirements, these are evaluated in a technical committee, named CAB1, and their complexity is evaluated.

10	Risk of using new technologies	New technologies	Knowledge or skill needed to develop the software. Software engineers use many technologies and methods.	A customer's requirements are evaluated in a technical committee, called CAB, and the technology needed for its production is chosen.
11	Risk of software configuration	Infrastructure and project's process	A software's architecture is chosen as a structure in the implementation of a software project based on software system's obligations. In software architecture, system components and the connecting protocols between them are identified. Software design is, in fact, the process of solving a problem and planning in order to develop the software. In fact, each component defined in the software architecture will be designed. Software engineering is a systematic, orderly and accurate method for designing a high quality software product.	A score is calculated for infrastructure of a project based on its architecture, design, software engineering and tools needed for operations.
12	Consistency and availability of datasets risk			
13	Risk of infrastructures and processes			
14	Risk of tools and software engineering methods			
15	Risks of software design			
16	Inadequate architecture risk			
17	Risk of change in obligations	Change in Customer demand	Changes occurring in initial SRS2 of the project	Changes are calculated in comparison of initial project and final SRS.
18	Lack of documentation	Documenting projects	Documentation of a project is execution of a system based on all project documents that emphasizes the integration of the information and data gathered through a project's production life cycle, use and updating.	Documentation process of a project is identified via evaluating user documentation, technical product documentation, use case availability and update levels
19	Risk of development process	Development processes	Multiple techniques are invented in software development, which fall into either category of waterfall models or fast track development models.	Project manager will decide on the best development strategy.
20	Number of team members risk	Number of project's team members	The number of people who are directly involved in the design, production, testing and settlement of the software makes up the project team.	Total number of personnel involved in the project
21	Lack of cooperation from employees	Communication levels of team members	Weak communication could result in failures in projects. As such, we should develop strong communication throughout the project to succeed; risks such as users aversion to change, problems between users, users with negative opinions about the project, users with no commitment to the project and users who do not cooperate.	Commutation levels of a project are identified through number of team members, participation of employees, cultural differences of a development team, their attitude towards change and communication between team members.
22	Risk of difference in culture between development team members			
23	Users resistance to change			
24	Risk of team members' communication			
25	Risk of organizational structure	Environment and organizational changes	Organizational changes are strategic novelties that influence an organization's process and performance. Some of the factors impacting change in business are internal or external. Some of the environmental risks of a project are management changes during a project, policy changes with negative effects on a project, poor organizational environment and organization under revisions during projects.	All environmental changes are similar to the point that all projects are related to one case study.
26	Risk of an unsuitable team environment			
27	Risk of management changes			
28	Risk of changing systems			
29	Risk of human resources policies			
30	Risk of key employees leave the jobs	Manpower leaving	Organizations always face the risk of losing key members.	It is calculated based on people leaving previously
31	Lack of effective project management skill	Involvement, experience and commitment of a project manager	A project manager sets goals for the project and uses his or her skills and knowledge to build trust and unity in the project team. Some of the risks involved in planning and project control are: poor project management, no control over	Levels of involvement, expertise and commitment of a project manager are calculated based on factors such as adequacy, involvement and controlling the progression of the
32	Risk of poor project planning			

33	Project progress not monitored closely enough		the progression of the project, wrong estimates for the resources needed, poor planning, poor view of project's strength and ineffective communication.	project.
34	Risk of inexperience project managers			
35	Risk of unskilled team members	Quality of human resources of project	A project faces risks when the project team does not have the qualifications necessary, and the team is inexperienced and underdeveloped.	Each member is given a rank based on experience and skills, and this is calculated as the average of ranks based on number of team members.
36	Risk of developers not being familiar with the business			
37	Risk of a poor programmer			
38	Risk of uneducated team members			
39	Risk of support quality	Support of product after settlement	Software support is a collection of follow-up services such as installation, running and instruction of software produced. Acceptable support service in this stage could result in more success for software projects.	Based on surveys of customers and banks about the product support
40	Risk of security bugs	Security bugs of a project	A bug is an error in software's performance that results in wrong outcomes or software's crashing. A security bug is information security errors and relates to the processing of an organization's valuable data.	It is the average of the security bugs of a product in different time intervals.
41	Risk of functional bugs	Application bugs of project	The application bugs in a project are errors in software's performance.	It is calculated based on the average of application bugs in a product in different time intervals.
42	Risk of inadequate tests for the project	Project's testing levels	Software test is the process of evaluating the software to make sure it works properly during usage. With the use of inclusive tests, it is possible to notice failures in many software systems and to overcome these risks.	A project's test levels are identified based on the number of application tests, automatic tests and unit tests done.
43	Lack of automatic test			
44	Inadequate project time estimation	Estimate deviation	Wrong estimates of the time it takes for the project to finish will increase cost and dissatisfied customers.	It is calculated through comparison of actual and planned project lifetime.
45	Unrealistic work estimation			

5.2. Designing expert evaluation system based on fuzzy inference systems

To design this expert system, MATLAB software and Rosetta software were used. Specifically, in MATLAB, the graphical user interface development environment (GUIDE) and the fuzzy logic toolbox were employed. Via the integration of these tools, a system was intended to be designed which would be user-friendly due to its graphical user interface as well as flexible and effective in optimizing system performance due to the use of fuzzy logic toolbox. The deviation from goals of a project (cost, time, quality and scope) was modelled in Rosetta using Rough set theory. The data from the past three years from our case study were used for inference rules and the most important impactful variables for the project goals were identified and are presented in Table 2 as our expert system's input variables.

With respect to the cumulative rules based on Rough set theory, it is possible to calculate the deviation of each project from its goals. During this step, the input and output variables of the fuzzy expert system were defined. Input variables were calculated based on analysis of Rough set theory. Linguistic variables were then transformed into fuzzy variables with the use of triangular functions and the Mamdani fuzzy inference.

Table 2
Definitions of the criteria and important effective variables

Criteria measurement description	The most influential variables	Measurement criteria	Sub-index	Criteria of software development projects
Budget considered for projects		Budget	Budget	Budget
Project implementing duration (person/hour)	*	Duration	Cost	Cost
The priority and the importance level of a project from the beneficiary perspective	*	Change urgency	Change urgency	Income and profitability in market
Estimated deviation of the plan and real costs of project	*	Cost Estimation Deviation	Cost Estimation Deviation	Cost Estimation Deviation
Project transparency level	*	SRS Transparency	SRS Transparency	The transparency of customer demand
Complexity level	*	Complexity	Complexity	The product technical complexity
Are new technologies needed?	*	Technology	Technology	New technology
Project's infrastructure level		Infrastructure	Infrastructure	Infrastructures and processes of project
Project customer	*	Customer	change request	Customers' requirements changing
Estimated deviation of the plan and real time of project	*	Estimation deviation	Estimation deviation	Estimation deviation
Selecting the project development method		Method	Method	Development method
The level of project environment changes		Environment	Environment	Organization area and changes
Some criteria should be measured as a team. These criteria are analyzed based on the experts' opinion about in what team the project is developed.	*	Team	Resource	The team members of project
			Interaction	The interaction of team members
			Exit Rate	Team members Departure
			Recourse degree	Project human resource quality
			Security incident	Project's security incident
			Functional Incident	Project's functional incident
			Test Level	Project test level
Parent point to the technical group that the project developed in. Criteria measures according to the selected technical group.	*	Parent	Documentation	Project documentation
			PMscore	Project manager specialty level
			Service score	Support after deployment

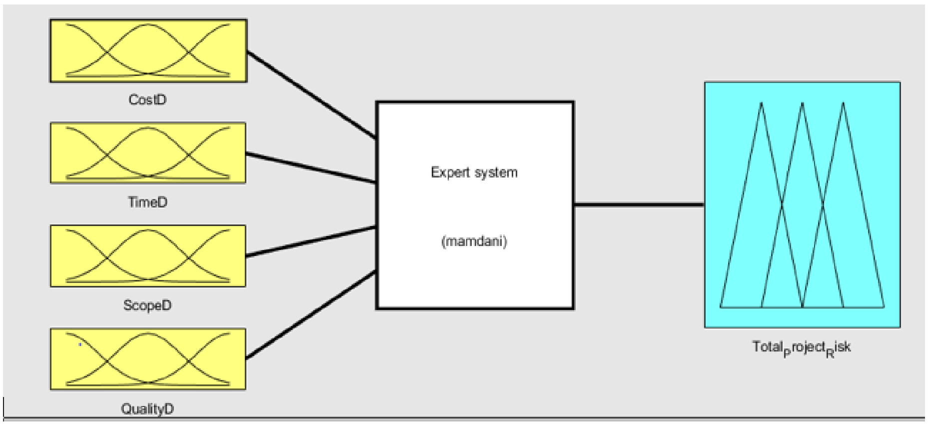

Fig 3. The input parameters and outputs of banking software development module

For de-fuzzification, a centroid method was used. Input variables of the system were as follows: cost deviation of banking software development projects (CostD), time deviation of banking software development projects (TimeD), scope deviation of banking software development projects (ScoreD) and quality deviation of banking software development projects (QualityD). The output variables were total project risk. These inputs and outputs of our expert system are presented in Figure 3.

Table 3
The total project risks' output variables

Scale	Fuzzy membership functions
Very Low	(0.15 0.03 0)
Low	(0.3 0.2 0.1)
Medium	(0.7 0.5 0.3)
High	(0.9 0.8 0.7)
Very High	(1 0.95 0.85)

Table 3 shows the partitioning of this expert system's output variables. In this paper, only one of these partitions is depicted in Figure 4.

The fuzzy expert system's rule base included extraction of expert rules and evaluation by experts. The first step to develop a fuzzy rule base in a fuzzy expert system is to build a set of "if-then" rules based on expert knowledge or the knowledge of the field.

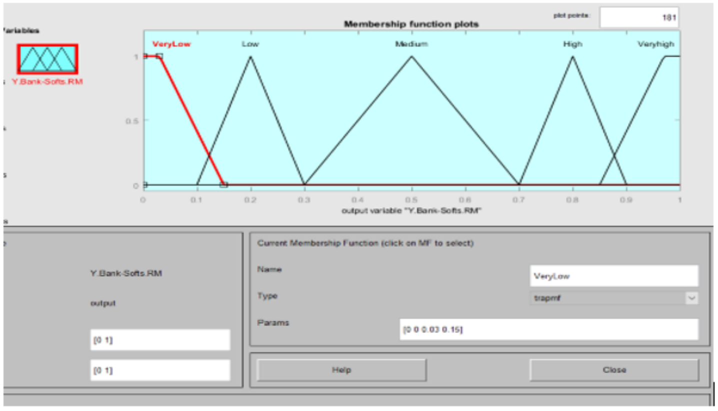

Fig. 4. The outputs of expert system- Value of fuzzy set variables (membership function) in MATLAB

The next step is to combine these rules in a single system. The following is the steps to build the rules for our fuzzy inference expert system base:

A. Using AHP method and expert opinion to calculate the weight of each main variable

Based on Ropponen and Lyytinen (2000), criteria such as system's performance, customer satisfaction, quality of product, cost and time of project and human resources management are taken into consideration for weighing goals using the AHP method. Finally, the weight for each variable is shown in Table 4.

A. Calculating output variables

Given the weight of each input variable, it is possible to calculate total risk of each project in different scenarios. Probability of different scenarios is considered to be the same in this paper. The number of rules produced is 625 provided we have 5 main variables, each having 5 states. Figure 5 shows the rule base of the fuzzy expert system.

All outputs are fuzzy in the previous step. To make the analysis easier, these fuzzy numbers need to be transformed into regular numbers. In other words, outputs need to be non-fuzzy and our de-fuzzification method of choice is the centroid method. When all elements of the fuzzy inference system including inputs, outputs, if-then rules and membership functions are identified, determining an interface for this system is the next step. The user interface relating to this expert system is depicted in Figure 6, which shows that in this sample, given the projects' variables, cost deviations (very low), time deviations (high), quality deviations (high) and scope deviation (medium), the total risk of this project is high.

Table 4
Input parameters of the system

Input parameters	***Final Weight***
Risk (deviation) of the software development projects-quality	0.372
Risk (deviation) of the banking software development projects-time	0.273
Risk (deviation) of the banking software development projects-cost	0.238
Risk (deviation) of the banking software development projects-scope	0.117

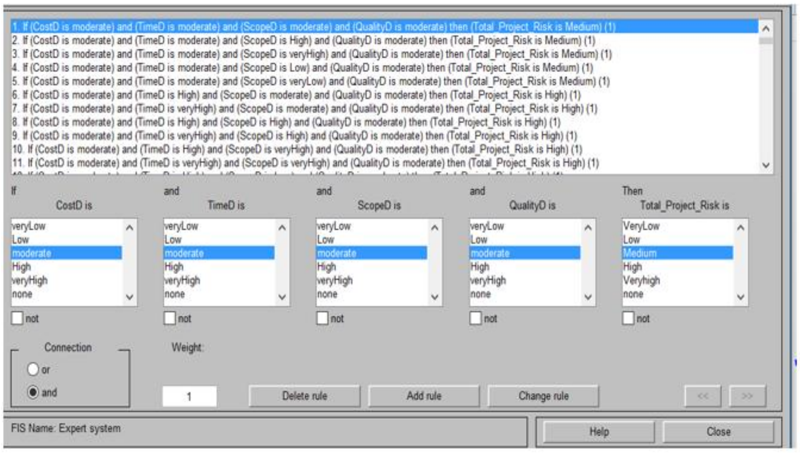

Fig. 5. The inside rules of the Rough module base (risk of the total software development projects)

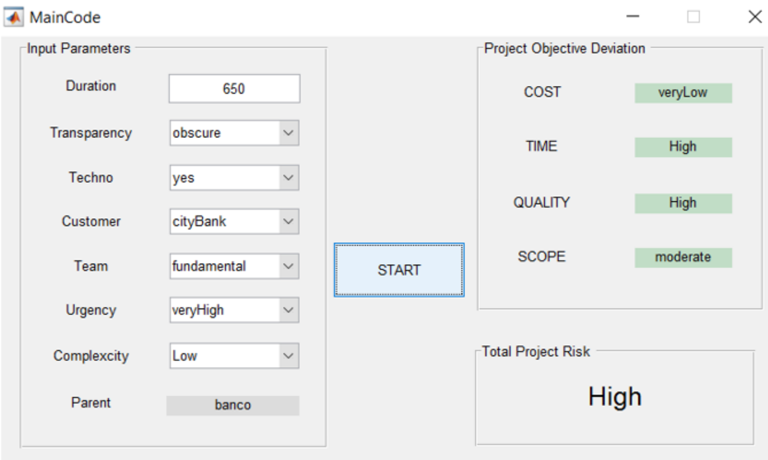

Fig. 6. System user interface

6. Model Testing

Given that implementing a conceptual model for a software program causes some errors, if the said errors are within an acceptable threshold, the model is also acceptable. Otherwise, it needs refurbishing. One of the main advantages of the proposed model is its high capacity for analyzing the results. In fact, a sensitivity analysis can be easily completed by changing input values of FIS model. In addition, the output surface of fuzzy inference systems provides the researchers and experts with an opportunity to examine the effect of criteria on performance. Also, the 3D plots are helpful in analyzing the consistency of the rules framed in the FIS and supporting the investigation of two-input and one-output systems (Pourjavad & Shahin, 2018). For instance, Figure 7 shows the output surface of the FIS.

Figure 7 shows the behavior of total project risk based on two-input variables, namely QualityD and CostD. As shown in Figure 7, input variable, namely QualityD, plays a more important role than input variable, namely CostD, for total project risk. In this section, the output's behaviors are analyzed to test the model. In this method, two input variables are fixed; then, the other two variables are increased (or decreased). Depending on the increase or decrease in the inputs, the outputs are recorded by the system. In the case that the output behavior is accepted based on expert opinions or literature relating to both fixed and changed variables, the expert system is accurate. Otherwise, changes have to be made. In addition to researchers, outputs are compared with the literature by experts and, then, analyzed and accepted.

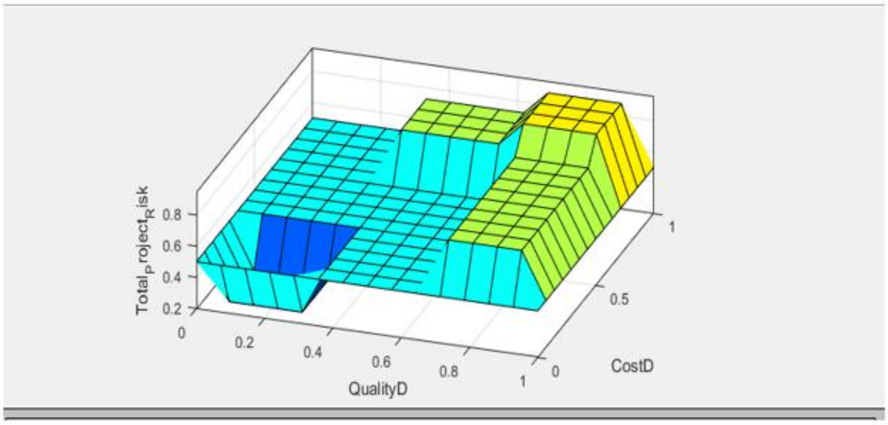

Fig 7. A 3D view of the output variable in terms of two-input variable quality and cost deviations

7. Results and Discussion

In the present study, first, the risks of a software development project were identified and discussed. Then, the literature and experts' opinions were studied to identify factors influencing the projects. Eventually, a fuzzy inference system was proposed to predict the total number of risks in software development projects. Four variables, namely time, cost, quality and scope, which explain the overall conditions of a project were chosen to evaluate the projects' risks. Based on the obtained results, the risks relating to quality and time had more importance and were more influential for the projects' overall deviation. Therefore, there is a need to have solutions and take preventive measures to deal with such risks in order to stop a project from failure. Several managerial implications for software development managers can be drawn from the suggested model in this study. The proposed model provided a general decision making framework for managers. In fact, managers could determine which dimensions of the project in their companies need more attention and which dimensions and criteria may be less significant. The system can predict the risks of each project before the project begins and help project managers be prepared to deal with these risks and consider ways to prevent the project from failing. The proposed model might be helpful for software engineers, project managers and researchers to make an idea about the reliability of software project in the early phase. Precisely, the proposed fuzzy rule formation expert system may be beneficial for the FIS users.

References

Addison, T. (2003). E-commerce project development risks: evidence from a Delphi survey. International Journal of Information Management, 23(1), 25-40.

Aqlan, F., & Lam, S. (2015). A fuzzy-based integrated framework for supply chain risk assessment. International Journal of Production Economics, 161, 54-63.

Arish, A. A. M., & Seyedesfahani, M. M. (2010). Model-based decision support in planning risk responses. International Journal of Industrial Engineering and Production Management, 20(3).

Arnuphaptrairong, T. (2011). Top ten lists of software project risks: Evidence from the literature survey. In Proceedings of the International Multi Conference of Engineers and Computer Scientists (Vol. 1, pp. 1-6).

Boehm, B. W. (1991). Software risk management: principles and practices. IEEE software, 8(1), 32-41.

Boehm, B. (1989). Software risk management. In European Software Engineering Conference (pp. 1-19). Springer, Berlin, Heidelberg.

Chapman, Ch., & Ward, S. (2011). How to manage project opportunity and risk: Why uncertainty management can be a much better approach than risk management. John Wiley & Sons.

Chen, W., Liu, K., Su, L., Liu, M., Hao, Zh., Hu, Y., & Zhang, X. (2014). Discovering many-to-one causality in software project risk analysis. In 2014 Ninth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (pp. 316-323). IEEE.

Chatterjee, S., Maji, B., & Pham, H. (2019). A fuzzy rule-based generation algorithm in interval type-2 fuzzy logic system for fault prediction in the early phase of software development. Journal of Experimental & Theoretical Artificial Intelligence, 31(3), 369-391.

Dokas, I. M., Karras, D., A., & Panagiotakopoulos, D. C. (2009). Fault tree analysis and fuzzy expert systems: Early warning and emergency response of landfill operations. Environmental Modelling & Software, 24(1), 8-25.

Faezy, F. (2015). A Grey-Based Fuzzy ELECTRE Model for Project Selection. Journal of Optimization in Industrial Engineering 17, 57-66.

Foong, K. Ch., Chee, Ch. T., & Wei, L. S. (2009). Adaptive network fuzzy inference system (ANFIS) handoff algorithm. In 2009 International Conference on Future Computer and Communication (pp. 195-198). IEEE.

Ford, F. N. 1985. Decision support systems and expert systems: a comparison. Information & Management, 8, 21-26.

Ghasemi, A., Golkar, M.J., & Eslami, M. (2015). A New Fuzzy Stabilizer Based on Online Learning Algorithm for Damping of Low-Frequency Oscillations. Journal of Optimization in Industrial Engineering 17, 1-10.

Han, W. M., & Huang, S. (2007). An empirical analysis of risk components and performance on software projects. Journal of Systems and Software, 80(1), 42-50.

Hadjimichael, M. (2009). A fuzzy expert system for aviation risk assessment. Expert Systems with Applications, 36(3), 6512-6519.

Jamshidi, A., Yazdani-Chamzini, A., Yakhchali, S. H., & Khaleghi, S. (2013). Developing a new fuzzy inference system for pipeline risk assessment. Journal of loss prevention in the process industries, 26(1), 197-208.

Lee, E., Park, Y., & Shin, J. G. (2009). Large engineering project risk management using a Bayesian belief network. Expert Systems with Applications, 36(3), 5880-5887.

Makui, A., Gholamian, M.R., Mohammadi, S.E. (2016). A Hybrid Intuitionistic Fuzzy Multi criteria Group Decision Making Approach for Supplier Selection. Journal of Optimization in Industrial Engineering 20, 61-73.

Mohagheghi, V., & Vahdani, B. (2017). An Assessment Method for Project Cash Flow under Interval-Valued Fuzzy Environment. Journal of Optimization in Industrial Engineering, 10(22), 73-80.

Mohammadipour, F., & Sadjadi, S. J. (2016). Project cost–quality–risk tradeoff analysis in a time-constrained problem. Comput. Ind. Eng. 95, 111–121.

Mousavi, S. M., Raissi, S., Vahdani, B., & MOJTAHEDI, SEYED MOHAMMAD HOSSEIN. (2011). A fuzzy decision-making methodology for risk response planning in large-scale projects. Journal of Optimization in Industrial Engineering, 4(7), 57-70.

Muriana, C., Vizzini, G. (2017). "Project risk management: A deterministic quantitative technique for assessment and mitigation" International Journal of Project Management 35 (2017) 320–340.

Nourian, R., Mousavi, S. M., and Raissi, S. (2019). A fuzzy expert system for mitigation of risks and effective control of gas pressure reduction stations with a real application. Journal of Loss Prevention in the Process Industries, 59, 77-90.

PMI. (2013). Project Management Body of Knowledge (PMBOK). 5thEd. Project Management Institute (PMI) Pub., USA.

Pimchangthong, D., & Boonjing, V. (2017). "Effects of Risk Management Practice on the Success of IT Project"Procedia Engineering 182 (2017) 579 – 586.

Paré, G., Sicotte, C. Jaana, M., & Girouard, D. (2008, January). Prioritizing clinical information system project risk factors: a delphi study. In Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008) (pp. 242-242). IEEE.

Pourjavad, E., & Shahin, A. (2018). The application of Mamdani fuzzy inference system in evaluating green supply chain management performance. *International Journal of Fuzzy Systems*, *20*(3), 901-912.

Ropponen, J., & Lyytinen, K. (2000). Components of software development risk: How to address them? A project manager survey. IEEE transactions on software engineering, 26(2), 98-112.

Hossain, Md. S. & Mahmud, Sh. (2016). Fuzzy Multi-Objective Linear Programming for Project Management Decision under Uncertain Environment with AHP Based Weighted Average Method. Journal of Optimization in Industrial Engineering, 9(20), 53-60.

Song, H., & Jiang, J. (2016). Risks Identification in Embedded Software Development: Evidence from MVBC Project Survey. Procedia Computer Science, 91, 798-806.

Sonchan, P., & Ramingwong, S. (2014, May). Top twenty risks in software projects: A content analysis and Delphi study. In 2014 11th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON) (pp. 1-6). IEEE.

Sumathi, S., & Paneerselvam, S. (2010). Computational intelligence paradigms: theory & applications using MATLAB. CRC Press.

Shrivastava, S. V., & Rathod, U. (2017). A risk management framework for distributed agile projects. Information and software technology, 85, 1-15.

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.

Wallace, L., & Keil, M. (2004). Software project risks and their effect on outcomes. Communications of the ACM, 47(4), 68-73.

Zhang, Y., & Fan, Z. P. (2014). An optimization method for selecting project risk response strategies. International Journal of Project Management, 32(3), 412-422.

This article can be cited: Karimi, T., Fathia, M.R. & Yahyazade, Y. (2021). Developing a Risk Management Model for Banking Software Development Projects Based on Fuzzy Inference System. *Journal of Optimization in Industrial Engineering*. 13 (2), 267-278.

http://www.qjie.ir/article_674017.html

DOI: 10.22094/JOIE.2020.1883892.1700